TECHNISCHE UNIVERSITÄT MÜNCHEN

Integrative Research Center Campus Straubing für Biotechnologie und Nachhaltigkeit

Heuristics in Service Operations Management

Fabian Schäfer

Vollständiger Abdruck der von der promotionsführenden Einrichtung Campus Straubing für Biotechnologie und Nachhaltigkeit der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktors der Wirtschaftswissenschaften (Dr. rer. pol.)

genehmigten Dissertation.

Voritzender:	Prof. Dr. Sebastian Goerg
Prüfer der Dissertation:	1. Prof. Dr. Alexander Hübner
	2. Prof. Dr. Clemens Thielen
	3. Prof. Dr. Jens Brunner

Die Dissertation wurde am 17.02.2020 bei der Technischen Universität München eingereicht und von der promotionsführenden Einrichtung Campus Straubing für Biotechnologie und Nachhaltigkeit am 10.07.2020 angenommen.

Contents

List of Figures

List of Tables

List of Algorithms

1 Introduction

Optimization problems are ubiquitous and part of everyone's life. On an ordinary working day we are faced with the questions of what to have to breakfast, what form of transportation to use to go to work, which route to take, in which order to do tasks, and so forth. All these kinds of optimization problems are well known in operations research (e.g., diet problem, shortest path problem, job-shop problem). Humans and also animals per se do hardly use mathematical formulations to solve their everyday problems. They rather rely on gut feeling, experience, knowledge or a combination of these three factors. This approach can also be referred to as *heuristic procedure.*

Emotion-based decision making is relatively satisfactory at a private level, whereas in business it rarely holds the promise of success. Therefore, data-driven decision making in organizations has become increasingly important. Thereby, customized solution approaches getting more and more into focus due to the high diversity of problem settings over the different economic sectors. Economic activities are traditionally divided into three sectors (Kenessey, 1987). First, the primary sector which involves extraction or production of raw materials (i.e., farming, forestry, mining, and fishing). Second, the secondary sector which is responsible for the transformation of raw materials into finished, usable goods (i.e., manufacturing and production). Lastly, the tertiary sector supplies services to other businesses or consumers (i.e., transportation, electric, gas as well as sanitary services, wholesale trade, and retail trade). The latter sector is also known as *service operations management.* While productivity has risen steadily in the primary sector and manufacturing sector in the past, this increase has been restrained in the service sector. With the emergence of new technologies, this is changing radically (Haller, 2015; Bordoloi et al., 2018).

This prevailing trend suggests a lot of operations management potential for the service sector. Therefore, this doctoral thesis examines possible real-world applications of operations research methods (i.e., heuristics) in service operations management.

The remainder of this doctoral thesis is structured as follows: The subsequent sections provide a deeper insight into the terms heuristic (1.1) and service operations management (1.2). Chapter 2 points out specific research areas, problems tackled and research contributions. Then chapters 3-5 consists of the articles published or submitted to the respective journals. Finally, chapter 6 concludes this dissertation.

1.1 Heuristics

Heuristic is a term derived from the Ancient Greek expression 'εὑρίσκω' which means to 'search', 'find' or 'discover'. It is also known as 'mental shortcut' or 'rule of thumb'. The main purpose is the abstraction of a problem by only taking into account a sub-quantity of all possible solutions or neglecting known information in the solution approach. Heuristic procedures are simplified decision-making processes, at the risk of often achieving a good but non-optimal solution. In real-world situations, due to the lack of information, limited time and, uncertainty, heuristic methods are necessary and efficient tools (Hertwig and Pachur, 2015).

In 1945 in the field of mathematics heuristic methods attracted attention through the work of the mathematician Pólya (2014). He proposes the following pattern consisting of four steps to solve a mathematical problem:

- Understanding the problem
- Create a plan
- Carry out the plan
- Look back, examine the solution obtained

If these steps do not result in a satisfying solution, adjustments have to be done. Hence, Pólya (2014) suggests simple generally applicable rules to develop a heuristic solution approach. For instance, trying to derive insights from similar problem settings that already have been solved, decomposing and recombining the problem, dividing the problem into smaller sub-problems, or defining a more specialized version of your problem. These rules laid the scientific foundations for the development of applied heuristics in the field of operations research and can still be found today in widespread use.

Then, also in the 1940s publications revealed construction methods that can achieve good results. Construction heuristics start from an empty solution and iteratively build their solution step by step according to specific scheme. For instance, the greedy algorithm selects the most promising element at each iteration (Sörensen et al., 2018). Correspondingly, when packing a knapsack with limited space, the item with the highest expected utility is packed first. The algorithms of Kruskal and Prim which faces the minimum spanning tree problem as well as the algorithm of Dijkstra which deals with the shortest path problem are examples demonstrating greedy algorithms (Cormen et al., 2009).

In the next generation of heuristics, researchers focused on the development of heuristic frameworks also known as *metaheuristics.* A metaheuristic can be defined as followed (Sörensen and Glover, 2013):

> *"A metaheuristic is a high-level problem-independent algorithmic framework that provides a set of guidelines or strategies to develop heuristic optimization algorithms. The term is also used to refer to a problem-specific implementation of a heuristic*

> *optimization algorithm according to the guidelines expressed in such a framework."*

In this phase, heuristics gained a lot of attention and a variety of new methods were proposed. Inspired by the concept of Darwin's theory of evolution which encompasses the existence of phenotypic variation among a generation, the hereditability over generations as well as the the survival of the fittest, evolutionary patterns were applied in heuristics (Fogel et al., 1966; Rechenberg et al., 1965; Schwefel, 1965). In 1975 the resulting framework of genetic algorithms were developed by Holland (1992) and redefined by Goldberg (1989). Further well known nascent frameworks at this time were for instance, the tabu-search, the simulated annealing algorithm and neural networks (Kirkpatrick et al., 1983; Glover, 1986; Hopfield, 1982).

To gain better results scientists started to merge metaheuristic approaches with one another around the turn of the millennium. This methodology established the age of *hybrid metaheuritics*. It compromises the combination and recombination of metaheuristcs with one another or any conductive method (e.g., mathematical programming, greedy heuristics) available at each level (Epitropakis and Burke, 2018; Sörensen et al., 2018). Hence, for instance, complete frameworks can be merged or only single operators of another framework can be applied. One popular widespread algorithm is the variable neighborhood search of Mladenović and Hansen (1997) which combine various local search operators and constructive procedures, respectively.

1.2 Service Operations Management

Operations management characterizes the designing and controlling of processes that are needed to produce goods or deliver services (Haller, 2015; Krajewski et al., 2016). Hereby, the premise is to satisfy customer needs as effectively as possible and in an efficient way that conserves resources (e.g., infrastructure, materials, labor). Managing production and services involves providing strategic (long-term), tactical (medium-term) and operational (short-term) decisions (Chase et al., 2006). This includes decisions on, for instance, facility, layout, capacity as well as production planning, inventory control, and shift scheduling.

Separate from the primary production and manufacturing sector, the service sector deals with the production of intangible products. This means, that the customer's added value is rendered by services provided through processes, people and information or a combination thereof (Krajewski et al., 2016). *Operations management* in the service sector is called *service operations management*. In the last centuries, the dominant form of employment sector swept first from agriculture to manufacturing and then to services due to the evolving industrialization with accompanying mechanization and amortization in production as well as manufacturing. Nowadays, all industrialized economies reveal the service sector as the leading form of employment and in generating revenue (Bell, 1976; Central Intelligence Agency, 2019). According to Bundesagentur für Arbeit (2019), Germany's employment-based largest branches in the service sector are 'health, veterinary and social work' (21%) and 'wholesale and retail trade; repair of motor vehicles and motorcycles' (19%). Revenue-based 'wholesale and retail trade; repair of motor vehicles and motorcycles' represents uncontested the lead with a market share of 59% in the service sector (Statistisches Bundesamt, 2019).

While the primary production and manufacturing sectors were already in the focus of operations management, service operations seemed to be neglected. The service sector could generally be described as labor-intensive and of low productivity, in which many activities could be carried out by low-skilled employees. This changed remarkably, the evolvement of technologies combined with the size of the service sector make *service operations management* in angle of operations research to one of the key topics in current research (Haller, 2015; Bordoloi et al., 2018).

2 Contributions

This chapter introduces three articles (chapters 3-5) written in conjunction with the doctoral thesis. As mentioned in the previous chapter, the topics dealt with are from the field of service operations management. The specified problem settings discussed were defined with collaboration partners, one of Germany's largest retailers and a large German hospital (on the basis of confidentiality agreements no clear names may be given). Therefore, the contributions are dived into two subjects, retailing and healthcare. The retailing project focuses on assortment and shelf-space optimization, whereas the healthcare project take look at the patient-bed assignment in hospitals. The sections 2.1 and 2.2 are intended as a guide to the specific topic. An overview of the contributions according title, journal submitted to and status of publication is given by Table 2.1. The following introduction of the contributions represents the initial situation at the beginning of the doctoral thesis after meeting the cooperation partners several times.

Methodology In the present application cases, real problems are identified and represented as mathematical models. These require the use of sophisticated solution approaches due to their inherent complexity (i.e., NP-hard). The peculiarity of each problem requires the development of a problem-specific solution method, which is based on heuristic approaches. Decisive for the suitability of a heuristic is runtime and solution quality. As a rule, these correlate negatively to one another. For example, in the case of time-critical problems, a loss of solution quality has to be accepted. The

Table 2.1: Status of publication

	Contribution	**Status**
Retailing		
1	Hübner et al. (2020): Maximizing Profit via Assortment and Shelf-Space Optimization for Two-Dimensional Shelves	Accepted in Production and Operations Management on 5th of September 2019
Healthcare		
2	Schäfer et al. (2019): Operational Patient-Bed Assignment Problem in Large Hospital Settings including Overflow and Uncertainty Management	Accepted in Flexible Services and Manufacturing Journal on 3rd of December 2018
3	Schäfer et al. (2020): Combining Machine Learning and Optimization for the Operational Patient-Bed Assignment Problem	Working paper, to be submitted to A-Journal

diversity of the problems leads in research to the development of innovative problem-specific algorithms, methods, and techniques.

Remark The versions of the contributions included in the following sections may slightly different from the versions published or submitted to the respective journals due to consistency of formatting, spelling, orthography, grammar, and nomenclature. This does not reduce the correctness or meaningfulness of the contributions.

2.1 Retailing: Assortment and Shelf-Space Optimization

Retailers use shelves to offer their products to customers. In doing so, they must decide how much shelf-space to allocate to which item. Shelf-space has been referred to as one of the retailer's scarcest resource. Hence, in order to maximize retail profit managing shelf-space is a critical point. Therefore, retailers have to consider multiple criteria. These can be split into customer and retailer based criteria. On the one hand, customers expect an assortment that attracts them and meets their demand. On the other hand, retailers are under pressure to reduce their inventory costs, provide products with the highest yields and avoid out of stock situations which reduce the revenue. This leads to decisions, which have to be made by retailers, relating to the products to be offered among a large number of competing products and the amount of shelf-space allocated to those products. The fact that shelf-space of supermarkets is permanently limited and the variety of products is steadily increasing, forces retailers to continuously adjust their assortment, as well as shelf-space allocation and make these decisions correctly to maximize their profits.

Contribution 1: Maximizing Profit for Two-Dimensional Shelves

Our cooperation company requires help with assortment and shelf-space optimization on tilted or two-dimensional shelves, respectively. In doing so, the partner company focuses on the optimization of profit. This applies to the following product groups, e.g., for meat, bread, fish, cheese or clothes. Therefore, a mathematical decision model needs to be formulated. Previous research provides insights into related problems. Geismar et al. (2015) modeled a revenue-optimizing two-dimensional shelf-space problem which takes into account a given assortment as well as given and known demand. They did not factor in substitutions for the assortment decision and space-elasticity for the space allocation. Hübner and Schaal (2017a) present a one-dimensional shelf-space problem which accounts for assortment decision and relevant demand effects, i.e., space-elasticity, out-of-assortment (OOA) and out-of-stock (OOS) substitutions. Because none of the one-dimensional approaches integrate the vertical and horizontal arrangement of items, they are not transferable to two-dimensional problem settings. Since the two basic problems (i.e., knapsack problem and two-dimensional knapsack problem) on its own are already NP-hard, a heuristic approach is needed to solve real-world data sets efficiently (Kellerer et al., 2004a; Pisinger and Sigurd, 2007). In summary, the following research questions are addressed:

i) How can the assortment and shelf-space optimization problem regarding two-dimensional shelves be represented as a mathematical model?
ii) Which insights can be derived from existing models in previous literature?
iii) Which method is suitable to efficiently optimize such an NP-hard problem?

2.2 Healthcare: Patients-Bed Assignment in Hospitals

Allocating patients to beds is an everyday task in hospitals, which is first of all driven by total bed capacity, patient compatibility, fluctuations in lengths of stay, and emergency arrival rates. Historically speaking, hospitals managed their patient-bed allocations on a first-come-first-serve basis. In addition, wards were typically dedicated to single departments with head nurses and doctors managing actual patient-bed assignment. To ensure profitability, occupancy levels have to be held high to ensure high utilization of hospital resources. High occupancy rates, however, greatly increase the probability of overflow situations which require disproportionate amounts of additional organizational work. This holds especially true for large maximum-care hospitals, which by definition are obligated to treat all incoming patients and have to deal with a lot of uncertainty due to a high share of emergency patients.

Contribution 2: Operational Patient-Bed Assignment Problem In a joint project with a large German hospital a decision model for the patient-bed assignment problem should be formulated. In contrast to previous literature (e.g., Demeester et al. (2010); Ceschia and Schaerf (2011); Range et al. (2014)), the main focus is on a real-world problem. Therefore, the three main stakeholders should be taken into account, namely patients, nursing staff, and doctors. The objectives and constraints of these stakeholders should be elaborated and integrated into the model. The patient-patient dependency should also be taken into account when occupying the rooms. For instance, a 90-year-old female patient who is dying has an influence on the satisfaction of an 18-year-old female patient who is accommodated in the same room. Furthermore, the model should consider planning of current emergency and elective patient arrivals, future elective patient arrivals, as well as anticipated future emergency patient arrivals. The patient-bed assignment problem is assumed to be NP-hard. Hence, a heuristic should be developed that is suitable and applicable for daily use as an efficient and quick support system for bed managers. In summary, the following research questions are dealt with:

i) What objectives do shareholders pursue in the patient-bed allocation process and what restrictions must be taken into account?
ii) Which solution method is suitable for practical use (i.e. proportion between computing time and solution quality)?

Contribution 3: Tackling Uncertainty in the Operational Patient-Bed Assignment Problem Based on the decision model developed by Schäfer et al. (2019), contribution (2.2) aims to exploit potential for improvement. This is expected to result in two areas. Firstly, combat uncertainties regarding emergency arrivals and secondly, an enhancement of the developed heuristic. To tackle uncertainties of emergency patient admissions, for instance, machine learning techniques can be applied to estimate these more precisely. To improve the solution quality of the heuristic solution a more sophisticated heuristic framework can be used (e.g., Mladenović and Hansen (1997); Duin and Voß (1999)). In summary, the following research questions are disscused:

i) Can the solution quality be improved with a more sophisticated approach?
ii) Which methods and metadata are appropriate for estimating emergency patient arrivals?
iii) Does the combination of an improved solution method and better estimation of input data lead to a significantly better solution approach?

3 Maximizing Profit via Assortment and Shelf-Space Optimization for Two-Dimensional Shelves

Co-autors: Alexander Hübner, Kai Schaal

Accepted in *Production and Operations Management* on 5th of September 2019
(in press, DOI: 10.1111/poms.13111)

Abstract Product proliferation and changes in demand require that retailers regularly determine how items should be allocated to retail shelves. The existing shelf-space literature mainly deals with regular retail shelves onto which customers only have a frontal perspective. This paper provides a modeling and solution approach for two-dimensional shelves, e.g., for meat, bread, fish, cheese or clothes. These are categories that are kept on tilted shelves. Customers have a total perspective on these shelves and can observe units of one particular item horizontally and vertically instead of just seeing the foremost unit of an item, as is the case of regular shelves. We develop a decision model that optimizes the two-dimensional shelf-space assignment of items to a restricted, tilted shelf. We contribute to current literature by integrating the assortment decision and accounting for stochastic demand, space elasticity and substitution effects in the setting of such self types. To solve the model, we implement a specialized heuristic that is based on a genetic algorithm. By comparing it to an exact approach and other benchmarks, we prove its efficiency and demonstrate that results are near-optimal with an average solution quality of above 99% in terms of profit. Based on a numerical study with data from one of Germany's largest retailers, we were able to show within the scope of a case study that our approach can lead to an increase in profits of up to 15%. We demonstrate with further simulated data that integration of stochastic demand, substitution and space elasticity results in up to 80% higher profits.

3.1 Introduction

This paper considers the problem of selecting, allocating and arranging products on retail shelves. Shelf space has been referred to as one of the retailer's scarcest resources (cf. e.g., Brown and Tucker (1961), Lim et al. (2004), Reiner et al. (2012), Hübner et al. (2013a), Kök et al. (2015), Geismar et al. (2015)). Up to 30% more products compete for the limited space than was the case ten years ago (EHI Retail Institute, 2014; Hübner et al., 2016). The increasing number of items to allocate, the shortage of shelf space, narrow margins in retail and the intensity of competition have greatly magnified the importance of retail assortment and shelf-space planning (cf. Hübner et al. (2013b)). Furthermore, customer satisfaction is mostly driven by availability of the right products. In order to achieve superior performance, retailers have to recognize customers' needs and identify these as key drivers (Nielsen, 2004; Eltze et al., 2013).

The selection of items and space allocation of the items to the shelf are interdependent planning problems when shelf space is restricted. The space available per product is less if broader assortments are offered and vice versa. Consequently, planning retail shelves involves the tasks of specifying the product assortments as well determining the space and quantities for selected items. These decisions are not only based on the margins of the products but also on associated demand and customer preferences. The more shelf space is allocated to an item, the more it attracts customers and the higher its demand. This is referred to as "space-elastic demand." This topic has gained a lot of research attention over recent years (see e.g., Hübner and Kuhn (2012), Kök et al. (2015), Bianchi-Aguiar et al. (2019)). Common characteristics of these models are that demand depends on the number of facings (= the foremost unit of an item in the front row of the shelf), and that retail shelves are observed by customers from a frontal direction. This firstly implies that a customer can only see the facing, and secondly, that two different products can only be positioned next to but not behind one another. We refer to this shelf type as a "regular shelf" herein.

For example, candies, coffee and tea, canned goods, cleaners and personal care products are presented on regular shelves.

Not all retail categories are kept on regular shelves. Some products are presented on “tilted shelves” (like counters, fridges or tables) onto which customers have a total perspective. Examples of these shelf types are to be found in Figure 3.1. These two-dimensional shelves types are for example used for the presentation of fresh food like bakery products and sausages, frozen products or in fashion retailing and consumer electronics. Many other retail formats fit into these settings, e.g., products and magazines in kiosks, snacks or electronics in vending machines and display ads (see also Geismar et al. (2015)). With these shelves items can be arranged more flexibly in the two-dimensional space, whereas with regular shelves the options are restricted by the shelf levels and their height. For example, two different products can be positioned next to and behind one another on two-dimensional shelves.

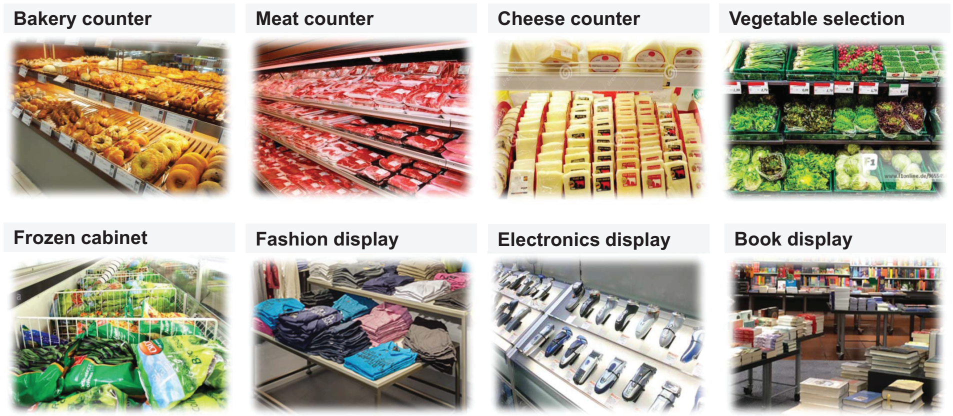


Figure 3.1: Examples of categories stored on two-dimensional shelves

There is already a rich literature on the planning of regular shelves. Typically, these models determine the shelf quantity and the number of facings for each shelf level (e.g., third level of second shelf). The most commonly used approach is to model the total shelf space via a one-dimensional shelf length (e.g., Lim et al. (2004); Martínez-de Albéniz and Roels (2011); Gilland and Heese (2013); Bianchi-Aguiar et al. (2016)). The models treat each shelf level with a one-dimensional front row space where only the front-row facings need to be determined as retailers usually fill up the entire shelf depth with more units of the respective product. Düsterhöft et al. (2019) propose a model for regular shelves that consider one-dimensional shelf levels of varying size in height, depth and width. As these models assume one-dimensional shelf space and defined shelf levels, they cannot be applied to two-dimensional applications where consumers have a different perspective. In one-dimensional approaches it is sufficient to determine the number of facings, whereas in two-dimensional problems the rectangular arrangement of the facings also needs to be determined. Two-dimensional problems require to compute horizontal and vertical number of facings (e.g., product A with 2 x 3 facings), the vertical and horizontal positioning of products within the two-dimensional area (e.g., product A positioned at certain x and y coordinates) and adjacent requirements of items (e.g., products A and B next to each other and C behind). Furthermore, vertical and horizontal sizes of products and shelves must be considered. Two-dimensional shelves face additional constraints, too, e.g., facings of a product need to be arranged in a contiguous rectangular shape, and not in other ways, such as L-forms.

To summarize, there are two different shelf types which each have their respective modeling requirements:

1. Regular shelves where items are allocated along an one-dimensional shelf length

2. Two-dimensional shelves where items are allocated to a two-dimensional shelf space and items need to follow particular arrangement constraints

One-dimensional solutions obtained for regular shelves cannot easily be transferred to two-dimensional selves as the arrangement of facings also needs to be integrated into decision-making. Only Geismar et al. (2015) have modeled a related two-dimensional shelf-space problem. Their model can also be applied to develop two-dimensional shelf plans. However, they assumed a given assortment, given and known demand and did not factor in substitutions for the assortment decision as well as space-elasticity for the space allocation. We extend this approach by accounting for assortment decisions, stochastic and space-elastic demand as well as out-of-assortment and out-of-stock substitution. We ultimately extend the two-dimensional problem that was introduced by Geismar et al. (2015) by using a more comprehensive demand function, a tailored solution procedure to the problem and numerical analysis to derive managerial insights. As such, the model of Geismar et al. (2015) represents a special variant of our demand model.

The remainder of this paper is organized as follows: Section 3.2 provides a detailed description of the setting and planning problem and related literature. Section 3.3 formulates the optimization model as a constrained multi-item newsvendor problem with substitutions. We develop a specialized heuristic to solve the related problem. This is represented in section 3.4. Numerical results and a case study are presented in section 3.5, while section 3.6 concludes.

3.2 Setting, planning problem and related literature

This section analyzes the scope (3.2.1), particularities of planning with two-dimensional shelves (3.2.2) and identifies the impact of these decisions on customer demand (3.2.3). Together, these build the foundation for the literature review and open research questions (see section 3.2.4).

3.2.1 Scope and planning approach

Shelf management comprises two hierarchical levels. One is a store (macro) level, deciding the space for product types (e.g., beverages, chocolate) and shelf types on a strategic level. The other a product category (micro) level which allocates individual products within each category on a tactical level. Our problem is concerned with the micro level, and considers the tactical allocation of a category of products onto a set of defined shelves. The shelf space available for a category is limited and determined by preceding decisions regarding store layout planning (cf. Hübner et al. (2013a)). The ultimate objective is to maximize retailers' profit which depends on the customer demand realized. This in turn depends on the positioning and space allocated to the products on the shelf, the product margins and operational costs. The traditional micro space-planning instrument of retailers is a planogram, representing an illustration of a shelf plan for a specific category. A planogram gives detailed information about the product's vertical and horizontal shelf position as well as the product's shelf quantity.

3.2.2 Particularities of two-dimensional shelves

Distinctive requirements of two-dimensional shelves need particular approaches. These are the (1) total customer perspective and two-dimensional item arrangement and (2) rectangular facing arrangements.

(1) Total customer perspective and two-dimensional item arrangement With the regular shelf on the left of Figure 3.2 customers only have a frontal perspective on the items offered. The retailer only needs to determine the number of facings, e.g., items A and B get one and item C gets three facings. The right of Figure 3.2 illustrates a two-dimensional shelf where the customer has a total perspective. The retailer must determine the total shelf quantity by choosing the shelf representation of an item, i.e. the number of vertical facings (width dimension) and horizontal facings (depth dimension). For instance, item F gets a shelf representation of (1 × 2), item G (1 × 4) and item I (2 × 2). Two products with different sizes can be positioned next to (e.g., F and G) and above one another (e.g., F and I). This means that item arrangements also need to reflect a two-dimensional neighborhood. With regular shelves there is a horizontal division represented by the shelf levels. The allocation of items to shelf levels is therefore restricted by shelf height. For example, a large family pack with a high box cannot be put at low-rise shelf level where small single-unit items are put. This is not the case for two-dimensional shelves where items do not necessarily need to be positioned along a dividing line or within a certain fixed compartment.

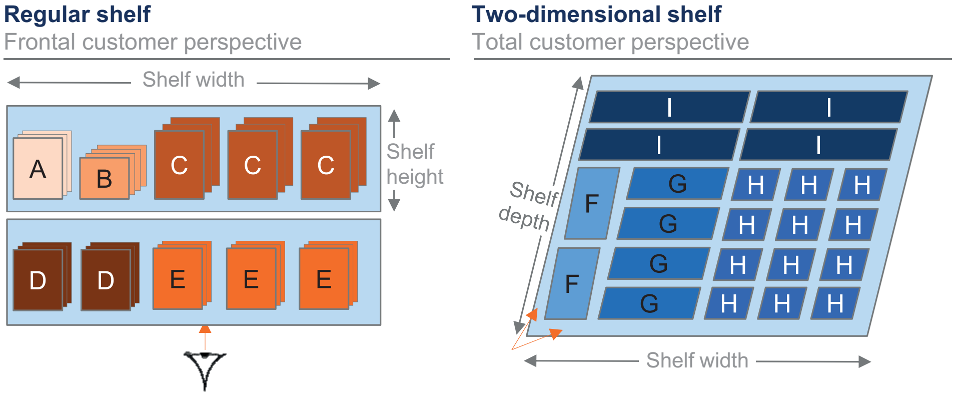


Figure 3.2: Illustration of a regular and a two-dimensional shelf

(2) Rectangular facing arrangements On two-dimensional shelves retailers usually arrange products in a rectangular shape, see e.g., empirical research in Marketing (Pieters et al., 2010) and Psychology (Berlyne, 1958). Figure 3.3 shows two related arrangement examples for two-dimensional shelves. This arrangement restriction implies that several facings of one item must be positioned adjacently and in a rectangular manner. For instance, if the retailer wants to place four facings of one specific product, these can only be positioned in three ways: (1 × 4), (2 × 2) and (4 × 1).

The rectangular requirement may result in "arrangement" and "prime number" defects if one-dimensional solutions (e.g., 5 facings) are transferred to a two-dimensional shelf setting (e.g., 2 × 2 facings). *Arrangement defects* occur if multiple rectangles (i.e., arrangements of different products) do not fit into one large rectangular arrangement (i.e., the shelf). Example 1

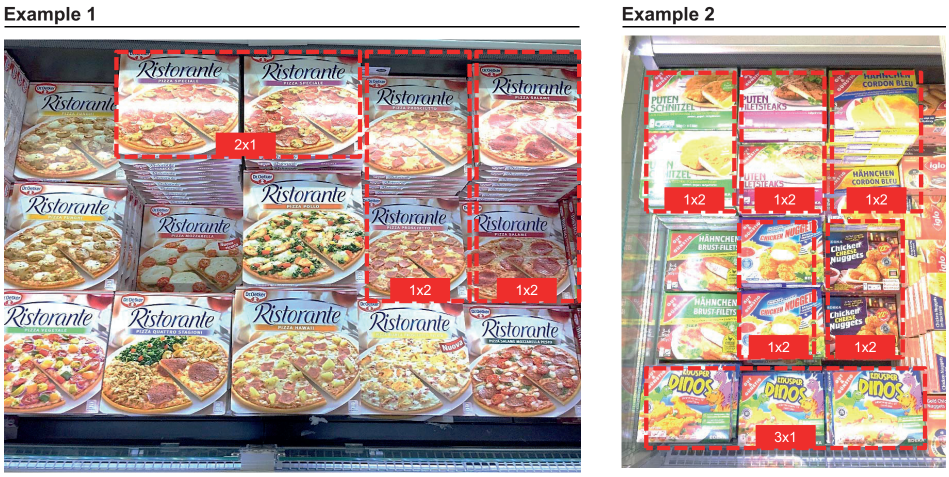


Figure 3.3: In-store arrangement examples for two-dimensional shelves

in Figure 3.4 shows this issue where not all facings of the optimal one-dimensional solution can be placed on the shelf such as to maintain a rectangular shape. We use identically sized items to simplify the illustration. The total shelf space is 9 for the one- and two-dimensional shelf. The optimal number of facings for the regular one-dimensional shelf is $A = 4$ facing, $B = 1$ facing and $C = 4$ facing. On one-dimensional shelves an item with 4 facings is placed in one row (1×4), whereas on two-dimensional shelves it can be placed in the form of 1×4, 4×1 or 2×2. Figure 3.4 shows that arranging both items A and C with 4 facings in a rectangular arrangement is not feasible as the total rectangular space is limited. The number of facings of item A or C therefore need to be reduced as only one item can have 4 facings. If, for example, item C now only has 3 facings, this may result in demand compensations by other items, and it may be preferable to list another item that compensates better the demand transfer between items. Example 2 in Figure 3.4 presents the *prime number defect*. Due to the rectangular requirement, quantities with prime numbers (like $3, 5, 7, 11, \ldots$) can only be arranged in a row (e.g., 1×3, 3×1, 1×5, 5×1, $1 \times 7, \ldots$). However, if this row is larger than the total horizontal or vertical space, this is a non-viable solution. The optimal number of facings of product A for a one-dimensional shelf is 5 in Example 2. Since 5 is a prime number and is greater than the length or depth of the shelf, the item cannot be displayed in a rectangular manner. The defects can be expressed formally as follows. Consider S as the total $(X \times Y)$-dimensional space and $\bar{S}$ its subset which represents the space currently unoccupied. Further, define the following set $R_{q_i} = (x \times y)$ as the set of all possible rectangular arrangements of the one-dimensional shelf quantity q_i of item i that needs to be assigned. The arrangement defect for an item i occurs if $\bar{S} \cap R_{q_i} = \emptyset$ and the prime number defect for an item i occurs if $|R_{q_i}| = 0$.

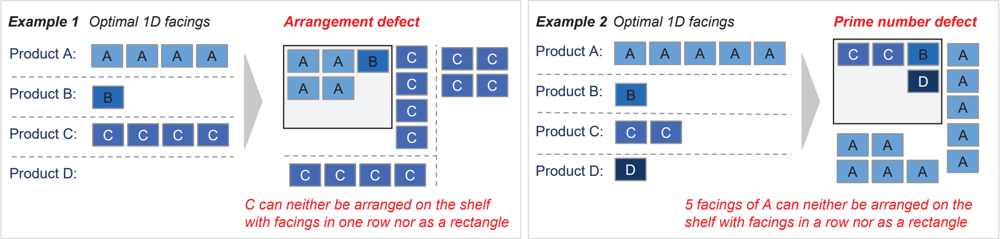

Example 1 Optimal 1D facings
Product A:
A A A A
Product B:
B
Product C:
C C C C
Product D:
Arrangement defect
C can neither be arranged on the shelf with facings in one row nor as a rectangle
Example 2 Optimal 1D facings
Product A:
A A A A A
Product B:
B
Product C:
C C
Product D:
D
Prime number defect
5 facings of A can neither be arranged on the shelf with facings in a row nor as a rectangle

Figure 3.4: Characteristics of an arrangement defect (example 1) and a prime number defect (example 2)

Summary To create a planogram for two-dimensional shelves, a shelf planner needs to make three simultaneous decisions for each category:

- *Item selection*: This decision involves *determining the assortment* of a category.
- *Space assignment*: This decision includes determining the *number of horizontal facings*, *number of vertical facings*, *quantity per facing*, and ultimately also the *total shelf quantity* for each product. The facings of one product can be arranged horizontally next to each other or vertically above one another. The total number of facings results from the multiplication of all vertical and horizontal facings.
- *Item arrangement*: This determines which vertical and horizontal coordinates are assigned an item, i.e., its exact location on the shelf. Furthermore, this also includes how different items are positioned next to each other (e.g., different types of bread next to each other). Finally, these

all need to follow arrangement guidelines so that a rectangular shape is obtained and adjacent requirements are adhered to.

Two-dimensional shelves are differentiated from regular shelves in terms of the options for space assignment and item arrangement. For regular shelves it is sufficient to use one-dimensional models to determine the horizontal number of facings. Two-dimensional shelves require a definition of horizontal and vertical facings in a rectangular shape. These rectangular shapes however depend on the arrangement of other items. An integrated approach is therefore required that simultaneously solves the four subproblems item selection, shelf quantity, space assignment and item arrangement. Solutions obtained from familiar one-dimensional models cannot be transferred directly to two-dimensional settings for this purpose as one-dimensional models lack the number of vertical facings and the item arrangement.

3.2.3 Related demand effects

All aforementioned decisions, namely item selection, space assignment and arrangement impact customer demand in three ways (see also Hübner and Schaal (2017a):

(1) Space-elastic demand The more facings an item is assigned, the higher its visibility on the shelf and the greater its demand. This demand effect is called space-elastic demand. Various empirical studies include tests that quantify space-elasticity effects (cf. Brown and Tucker (1961), Frank and Massy (1970), Curhan (1972), Drèze et al. (1994), Eisend (2014)). Chandon et al. (2009) show that the variation of facings is the most significant in-store factor, even stronger than pricing. Desmet and Renaudin (1998) reveal that space elasticities increase with the impulse buying rate. The magnitude of this demand increase depends on the item's space-elasticity factor, which indicates the percentage increase in demand of an item every time the number of facings goes up by a given amount. Using a meta-analysis, Eisend (2014) identifies an average demand increase by a factor of 17%. Cross-space elasticity measures responsiveness in the quantity demand of one item when the space allocated for another item changes. Eisend (2014) calculates an average cross-space elasticity of -1.6%. Schaal and Hübner (2018) used numerical studies to show that the low empirical cross-space elasticity values either do not have or have only very limited impact on optimal shelf arrangements. We therefore disregard cross-space elasticities in the following. The demand impact of an item's position can be neglected for two-dimensional shelves. These positioning effects are relevant for regular shelves where e.g., eye- vs. knee-level positions have a different demand impact. The same holds true for large categories where the shopper's walking path and positions at the beginning, middle or end of an aisle matter. With two-dimensional shelves, however, the basic idea is to allow the customer to oversee the total assortment of one (sub-)category at one glance.

(2) Out-of-assortment and (3) Out-of-stock substitution demand
Customers can substitute for their choice if items are unavailable. For example, Gruen et al. (2002), Kök and Fisher (2007), Aastrup and Kotzab (2009) and Tan and Karabati (2013) show that between 45% and 84% of the demand can be substituted. Unavailability of items can result from two scenarios: either an item is delisted as a consequence of the assortment decision (out-of-assortment, OOA), or it is temporarily unavailable and currently not available on the shelf (out-of-stock, OOS). In both situations, customers may replace the unavailable items with other items which results in demand increases for the respective substitutes.

Substitution rates can be obtained by direct consumer surveys or by transactional data (e.g., Kök and Fisher (2007), Tan and Karabati (2013)). A straightforward approach often applied to obtain substitution rates is to base them on market shares (Hübner and Kuhn, 2012). This means that if an item has an overall demand share of 50%, the substitution rate from all products to this particular product is 50%. Finally, expert workshops can also be used to define substitution rates by selecting related items and rates.

3.2.4 Related literature and contribution

Current shelf planning literature focuses on regular shelf types (see also the reviews of Hübner and Kuhn (2012), Kök et al. (2015)). We will first analyze this stream of literature and divide it into contributions that assume a given assortment and into contributions that integrate the assortment decision into shelf planning. This review is mainly used to gain insight into the different approaches for modeling demand and solution approaches. Secondly, we focus on particular applications to two-dimensional shelf space problems. This review is used to define open research gaps and specify our contribution.

(1) Applications for regular shelves Most shelf-space optimization models assume deterministic demand and optimize the number of facings for items with space-elastic demand to be assigned to limited shelf space. Respective approaches help retailers solve the trade off between more shelf space (and thus demand increases due to a higher number of facings) for certain items and less available space (and thus demand decreases due to a lower number of facings) for other items. One of the first models goes back to Hansen and Heinsbroek (1979) who formulate a shelf-space model that accounts for space elasticity and solve it using a Lagrangian heuristic. Corstjens and Doyle (1981) develop a shelf-space model that accounts for space and cross-space elasticities which is solved via geometrical programming. Zufryden (1986) presents a dynamic programming approach with space-elasticity effects. Lim et al. (2004) present a model that considers space and cross-space elasticities for which they develop various extensions, e.g., for product groupings. A specialized heuristic and the combination of a local search and a metaheuristic approach are used to solve it. Hansen et al. (2010) develop a formulation with space and cross-space elasticities for which they compare the performance of various heuristic and meta-heuristic algorithms. The model also differentiates between horizontal and vertical shelf positions. Bianchi-Aguiar et al. (2015) use a mixed-integer programming approach to formulate a deterministic model that considers product-grouping and display-direction constraints and incorporates merchandising rules. Hübner and Schaal (2017b) formulate the first stochastic shelf-space model that is solved with specialized heuristics. They account for space and cross-space elasticity as well as vertical positioning effects. The model assumes a given assortment and does not incorporate substitution effects. In summary, the shelf-space models mentioned assume a given assortment and optimize the number of facings. They do not take into account substitutions for unavailable items.

We further investigate contributions that integrate assortment decisions into their models in the following. Hübner (2011) develops a mixed-integer model for shelf-space planning that also takes assortment decisions into account. OOA situations are considered, but because the model assumes deterministic demand, OOS is ignored. Irion et al. (2012) use a piecewise linearization technique to solve a deterministic shelf-space model that accounts for space and cross-space elasticities. Although the model accounts for the assortment decision by setting facings to zero additional demand for OOA substitution is neglected. Hübner and Schaal (2017a) proposed the first integrated assortment and shelf-space optimization model that accounts for stochastic demand, substitution and space elasticity. To the best of our knowledge, they present the most comprehensive demand model. They showed that assortment and shelf planning are interdependent when shelf-space is limited. A heuristic was developed to address large-scale problems. The heuristic approach was modeled as an iterative MIP algorithm that uses recalculated precalculations for each step to circumvent the non-linear problem. The integrated approach outperforms alternative approaches, e.g., a sequential planning approach that first picks assortments and then assigns shelf space.

(2) Applications for two-dimensional shelves Solutions obtained from one-dimensional regular shelf settings, such as the above, cannot be transferred to two-dimensional shelves due to arrangement and prime number defects. Only Geismar et al. (2015) have developed a model and solution approach for two-dimensional shelves. They assume multiple shelves that are called cabinets. Each cabinet can have a distinct number of columns and rows. The capacity (or number of slots) of a shelf can be calculated by multiplying the columns and rows. Each product must have all of its units displayed within a single cabinet, and those units have to be displayed in a contiguous rectangle. All units need to have standardized unit sizes. To formulate the model in a more realistic and flexible manner, Geismar et al. (2015) did not divide cabinets into subsections to reduce the solution space or rather the complexity. Their formulation makes it possible to apply all the different dimensions of the product presentation within one cabinet according to the restrictions mentioned. In contrast to the majority of existing shelf-space models, the objective is to maximize revenues rather than profit. Moreover, demand effects such as substitution or space elasticity were neglected. Apart from that, the demand is assumed to be deterministically known. However, the demand is affected by the effectiveness of a row. Each row can have a distinct effectiveness value. Due to the fact that the MIP approach did not find a solution within a two-week time limit, they broke the problem into two subproblems. First, the products are assigned to the cabinets. Secondly, the units are arranged within the cabinets. The evaluation of observed data revealed an average revenue improvement of 3.7%.

Summary Table 3.1 gives an overview of the main contributions. The demand models and solution approaches for regular one-dimensional shelves have gradually been refined. Hübner and Schaal (2017a) present the most comprehensive model by integrating assortment and space allocation and taking relevant demand effects into account, i.e., space-elasticity, OOA and OOS substitutions. Previous literature suffers from one or more of

Table 3.1: Related literature on assortment and shelf-space optimization

		Demand effects						
Contribution	Decisions[1]	Space-elastic demand	Stochastic demand	OOA demand	OOS demand	Perspective	Items[2]	Solution approach
One-dimensional shelves								
Hansen and Heinsbroek (1979)	S	✓				Frontal	6,443	Specialized heuristic
Corstjens and Doyle (1981)	S	✓				Frontal	5	Geometrical programming
Zufryden (1986)	S	✓				Frontal	40	Dynamic programming
Lim et al. (2004)	S	✓				Frontal	100	Specialized heuristic
Hansen et al. (2010)	S	✓				Frontal	100	Meta-heuristic, simulation
Hübner (2011)	A/S	✓		✓		Frontal	250	MIP
Irion et al. (2012)	A/S	✓				Frontal	50	Piecewise approximation
Bianchi-Aguiar et al. (2015)	S	✓				Frontal	240	Specialized heuristic
Hübner and Schaal (2017b)	S	✓	✓			Frontal	200	MIP
Hübner and Schaal (2017a)	A/S	✓	✓	✓	✓	Frontal	2,000	Specialized heuristic
Two-dimensional shelves								
Geismar et al. (2015)	S/I					Frontal	579	Subproblem decomposition
This paper	A/S/I	✓	✓	✓	✓	Total	2,000	Specialized heuristic

[1] A = Assortment; S = Space assignment; I = Item arrangement (i.e., vertical and horizontal coordinates on the shelf)

[2] Maximum number of items considered in numerical tests

the following drawbacks. First of all, only isolated optimization of either assortments or shelf-space, ignoring the interdependence of both decisions. Secondly, limited consideration of relevant demand effects. Thirdly, applicability in practice is constrained by the limited assortment sizes that can be solved. None of the one-dimensional shelf models integrate the vertical and horizontal arrangement of items. Geismar et al. (2015) presented the first extension for two-dimensional problems and define the position of products. However, they apply a very restricted demand model and do not optimize assortments. We will base our extensions on the contributions of Geismar et al. (2015) and Hübner and Schaal (2017a). We contribute a new and more general approach by integrating assortment, space allocation and item arrangement decisions in a two-dimensional shelf-space setting. We further extend the demand model via space-elastic demand and substitutions. This also includes the modeling of stochastic demand. Integrating demand volatility is relevant for retail settings (see e.g., Agrawal and Smith (1996) or Hübner et al. (2016)), particularly for categories with perishable products (see e.g., Kök and Fisher (2007)). This becomes even more important for two-dimensional shelves as the majority of products kept on these shelves are perishable, e.g., fresh products like produce, products with limited sales periods like fashion and electronics. Finally, we relax the assumption of identical unit sizes as this does not hold true in most practical applications.

3.3 Development of the decision model

This section develops the Two-Dimensional Stochastic Capacitated Assortment and Shelf-space Problem (2DSCASP) in three steps: First, the decision model is formulated in section 3.3.1 which is then supplemented with the demand model in section 3.3.2. Finally, section 3.3.3 determines the arrangement and shelf space constraints. Table 3.2 shows the notation used.

Table 3.2: Notation

Indices and sets	
i, j	Item indices
$\mathbb{N}$	Total set of items
$\mathbb{N}^+$ ($\mathbb{N}^-$)	Set of listed (delisted) items
R	Total set of rectangles
Parameters	
β_i	Space elasticity of item i
γ_{ji}^{OOA} (γ_{ji}^{OOS})	Share of demand of item j that gets substituted by item i in the event that item j is out-of-assortment (out-of-stock)
$\hat{\delta}_i$	Total expected demand of item i
δ_i^{min} ($f_{\delta_i^{min}}$)	Minimum demand of item i (corresponding density function)
δ_i^{sp} ($f_{\delta_i^{sp}}$)	Space-elastic demand of item i (corresponding density function)
δ_i^{OOA} (δ_i^{OOS})	Out-of-assortment (out-of-stock) demand of item i
c_i	Unit cost of item i
d_i (w_i)	Item depth (width) per unit of item i
f_i^*	Demand density function for i, generic form
K_i	Maximum number of facings of item i
n_{ij}	Binary parameter indicating whether item i has to be a neighbor of item j (=1) or not (=0)
N	Total number of items
p_i	Sales price for one unit of item i
s_i	Penalty cost for one unit of item i
$S^{width}(S^{depth})$	Total shelf width (depth) available
v_i	Salvage value for one unit of item i
Decision variables	
q_i^x (q_i^y)	Integer number of facings of item i assigned in x-dimension (y-dimension)
q_i^t	Integer number of units of item i that are stacked behind one facing
$coor_i^x$ ($coor_i^y$)	Integer location coordinate of item i in the x-dimension (y-dimension)
l_{ij} (b_{ij})	Binary variable denoting whether item i is arranged on the left of (below) item j (=1) or not (=0)
Auxiliary variables	
k_i	Number of facings assigned to item i, with $k_i = q_i^x \cdot q_i^y$
q_i	Shelf quantity assigned to item i, with $q_i = q_i^x \cdot q_i^y \cdot q_i^t$
z_i	Binary variable indicating whether item i is selected in the assortment (=1) or not (=0)
D_i (W_i)	Space of item i occupied in a depth (width) dimension

3.3.1 Modeling the decision problem

The retailer must assign products of a particular category to a two-dimensional shelf limited in size. That means considering a set of items $\mathbb{N}$ with $N = |\mathbb{N}|$ and optimizing the profit by simultaneously deciding

- which products to list at all (item selection),
- how much shelf space to allocate to the items listed (space assignment),
- how the total item quantity is presented through horizontal and vertical facings in a rectangular shape, e.g., 4×1 or 2×2, *and* where the product is positioned, i.e., x- and y-coordinates of the shelf space (item arrangement).

We introduce various decision and auxiliary variables to express these decisions. We allow the shelf quantity q_i to be zero ($q_i = 0$) to account for delisting of items. The retailer must arrange the number of facings k_i, $i = 1, 2, \ldots, N$ for each item N in a contiguous rectangular shape on the two-dimensional shelf. The number of facings for the x-dimension is expressed by the integer decision variable q_i^x and by q_i^y for the y-dimension. The total number of facings k_i is therefore computed by $k_i = q_i^x \cdot q_i^y$. Since it is possible to stack each item, the entire shelf quantity q_i is computed by $q_i = k_i \cdot q_i^t$ where q_i^t denotes the number of units of item i that are stacked behind each facing k_i, including the facing itself. We assume that there is no backroom storage which implies that all products listed have to fit onto the available shelf space. The retailer objective is to maximize total profit Π which is the sum of the item profits π_i of all items $i \in \mathbb{N}$:

$$maximize\ \Pi(\bar{q}) = \sum_{i \in \mathbb{N}} \pi_i(q_i) \tag{3.1}$$

The item profit π_i depends on the shelf quantity q_i for each item $i \in \mathbb{N}$ that is available for demand fulfillment. Items can be sold at the sales price p_i and are purchased for the unit costs c_i which incorporate all purchasing and processing costs (e.g., for replenishment). If the expected demand D_i for item i is greater than the shelf quantity q_i, the excess demand is lost and the retailer suffers the shortage costs s_i. Conversely, if items remain in stock at the end of the period, they need to be disposed of at a salvage value v_i and the retailer incurs a loss, because $v_i < c_i$.

The profit for each item is calculated as shown in Equation (3.2) and consists of the following elements: The first term represents the overall purchasing costs, the second and fourth term calculate the expected revenues, the third term represents the expected revenues from leftover items sold for the salvage value, and the fifth term calculates the penalty costs in the event of stockouts. This generic form of the item profit π_i corresponds to the profit calculation in newsvendor problems and can therefore also be found in many other assortment related decision models (e.g., Smith and Agrawal (2000), Kök and Fisher (2007), Hübner et al. (2016)). The difference always stems from the demand that is taken into account which is represented by the density function f_i^*. This probability density function f_i^* in Equation (3.2) accounts for the relevant total demand distribution which must be quantified in accordance with the assumed customer behavior. In our case the density function must take into account OOA and OOS substitution as well as the space-elastic demand. We investigate the related demand function in more detail below.

$$\begin{aligned}\pi_i(q_i|_{q_i=q_i^x \cdot q_i^y \cdot q_i^t}) = -c_i \cdot q_i + p_i \cdot \int_0^{q_i} y f_i^* dy + v_i \cdot \int_0^{q_i} (q_i - y) f_i^* dy + p_i \\ \cdot \int_{qi}^{\infty} q_i f_i^* dy - s_i \cdot \int_{q_i}^{\infty} (y - q_i) f_i^* dy\end{aligned} \tag{3.2}$$

The model does not force the user to completely fill the available shelf space. It is permitted to leave free spaces due to penalty costs for oversupply. In constellations with large shelves, low demand and high oversupply costs, for example, there could be situations where the full space is not used. However, this is assumed to be rather a hypothetical situation due to general space constraints in retail stores.

3.3.2 Modeling the demand function

The probability density function f_i^* of the standard newsvendor formulation needs to be enriched in order to consider different demand effects. Because items can be delisted, we divide the set of all items $\mathbb{N}$ into listed items ($\mathbb{N}^+$) and delisted items ($\mathbb{N}^-$) in the following, such that $\mathbb{N}^+, \mathbb{N}^- \subseteq \mathbb{N}$, $\mathbb{N}^+ \cup \mathbb{N}^- = \mathbb{N}$ and $\mathbb{N}^+ \cap \mathbb{N}^- = \oslash$. The total expected demand $\hat{\delta}_i$ of an item i consists of three elements (see Equation (3.3)). The first is space-elastic demand δ_i^{sp} which is driven by the number of facings. Next is the OOA demand δ_i^{OOA} which depends on whether the items j, for which i is a substitute ($j \neq i$) are listed ($q_j > 0$) or not ($q_j = 0$). The third is OOS demand δ_i^{OOS} which depends on the available shelf quantity q of the other items j ($j \neq i$). We elaborate on the three demand components below.

$$\hat{\delta}_i = \delta_i^{sp} + \delta_i^{OOA} + \delta_i^{OOS} \qquad i \in \mathbb{N} \tag{3.3}$$

Space-elastic demand Customer demand for an item grows with the number of facings assigned for this item. The magnitude of the demand increase depends on the space elasticity β_i, the number of facings k_i and the minimum demand δ_i^{min}. The space-elastic demand is denoted by $\delta_i^{sp}(k_i)$ and calculated corresponding to Equation (3.4). The corresponding density is denoted by $f_{\delta_i^{sp}}$.

$$\delta_i^{sp}(k_i|_{k_i=q_i^x \cdot q_i^y}) = \delta_i^{min} \cdot k_i^{\beta_i} \qquad i \in \mathbb{N} \tag{3.4}$$

The space-elastic demand grows with a diminishing rate with $k_i^{\beta_i}$ for $k > 1$. The minimum demand δ_i^{min} is equal to the demand of an item if it were represented with one facing ($k_i = 1$), i.e., $\delta_i^{min} = \delta_i^{sp}(k_i = 1)$ (cf. Hansen and Heinsbroek (1979), Corstjens and Doyle (1981), Urban (1998), Hansen et al. (2010), Irion et al. (2012), Bianchi-Aguiar et al. (2015), or Hübner and Schaal (2017a)).

The space-elastic demand for an item i with $k_i = 0$ mathematically results in no demand as $\delta_i^{sp}(k_i = 0) = \delta_i^{min} \cdot 0^{\beta_i} = 0$. This does not hold true since some customers would still want to buy the item even if it was not shown on the shelf anymore. To factor in this effect, we assume the identical minimum demand for a product with no facings as if it had exactly one facing. In other words, the demand with one facing is described as the minimum demand even if a product is delisted. In cases of $k_i = 0$, this means we assign space-elastic demand by applying $\delta_i^{sp}(k_i = 0) = \delta_i^{min}$. The corresponding density function for the minimum demand is denoted by $f_{\delta_i^{min}}$.

Out-of-assortment demand OOA demand for a listed item i ($i \in \mathbb{N}^+$) occurs if another item j is delisted ($j \in \mathbb{N}^-$) and customers substitute this item j with item i. We assume that if item j is delisted, customers substitute a certain share γ_{ji}^{OOA} of the minimum demand δ_j^{min} of item j with item i, because some customers will still want to buy item j, even if it is not listed. The maximum quantity that can be substituted of item j cannot be higher than the minimum demand of the item j. This is first due to the aforementioned assumption that the space-elastic demand in the event of $k = 0$ corresponds to the minimum demand, and secondly because we follow the usual assumption that substitution takes place over one round only (cf. e.g., Ryzin and Mahajan (1999), Smith and Agrawal (2000), Gaur and Honhon (2006), Kök and Fisher (2007), or Hübner et al. (2016)). This simplification is common across most assortment literature (cf. Kök et al. (2015)). If consumers want to substitute their first choice by a product that is not available, the demand is lost as a result. There is no attempt to model individual consumer decisions. Instead, an exogenous model is applied that is capable of capturing aggregated consumer demand. The resulting model is cruder than some other substitution models but has the advantage of being much easier to analyze and requiring less data. That also means that demand is uniform across time. To summarize, this implies that demand is lost if a substitute is not available either. Therefore, if an OOA item is a substitute for another non-available item, the additional substitution demand for the OOA item would only occur if it was available. The OOA demand of an item i is calculated as follows:

$$\delta_i^{OOA} = \sum_{j \in \mathbb{N}^-, j \neq i} \delta_j^{min} \cdot \gamma_{ji}^{\mathrm{OOA}} \qquad i \in \mathbb{N} \tag{3.5}$$

The density function for OOA demand for item i is calculated by Equation (3.6). Since we assume that the distributions of the minimum demand of two items i and j, $i \neq j$, are independent, the convolution – represented by the operator $\circledast$ – can be used to calculate the distribution of the sum of the demand of the two items (cf. Hübner et al. (2016)). Equation (3.6) convolutes the (minimum) demand distribution functions of all delisted items and therefore accounts for the fact that the OOA substitution demand for item i depends on all delisted items $j \in \mathbb{N}^-$. To simplify, we have omitted the γ_{ji}^{OOA} parameters in the equation.

$$\circledast_{j \,\in\, \mathbb{N}^-} f_{\delta_j} = \int \cdots \int_{\mathbb{R}_0^{+,n}, j \in \mathbb{N}^-} f_{\delta_j^{min}} d\tau \dots dv \tag{3.6}$$

Out-of-stock demand OOS demand for a listed item i ($i \in \mathbb{N}^+$) occurs if another listed item j ($j \in \mathbb{N}^+$) is temporarily out-of-stock, i.e., if demand for item j exceeds the available shelf quantity of item j. In this case, we assume that customers substitute a certain share of the shortage quantity of item j with item i. The shortage quantity of item j is calculated via $(\delta_j - q_j | \delta_j > q_j)$ and the substitution share denoted by γ_{ji}^{OOS}. Equation

(3.7) shows the OOS demand calculation (also see e.g., Rajaram and Tang (2001), Kök and Fisher (2007), Hübner et al. (2016)):

$$\delta_i^{OOS} = \sum_{j \in \mathbb{N}^+, j \neq i} [(\delta_j - q_j) | \delta_j > q_j] \cdot \gamma_{ji}^{\text{OOS}} \qquad i \in \mathbb{N} \tag{3.7}$$

Equation (3.8) depicts the density function for OOS demand for item i. As above, we use the convolution to account for the fact that OOS demand for an item i depends on the expected shortage of all temporarily unavailable items other than item i.

$$\circledast_{j \,\in\, \mathbb{N}^+} f_{\delta_j} = \int \cdots \int_{q_j, j \in \mathbb{N}^+}^{\infty} f_{\delta_j} d\tau \ldots d\upsilon \tag{3.8}$$

3.3.3 Modeling the arrangement and space constraints

Before we specify the constraints of our problem, we give a broader context on the modeling of the arrangement constraints which also impacts the solution approach later on.

General modeling approach Our problem belongs to the class of Two-Dimensional Knapsack Problems. These problems deal with the selection and arrangement of a set of rectangles $r \in R$ to a capacitated two-dimensional rectangular container S with a certain width (S^{width}) and depth (S^{depth}). In our case, the rectangle r represents not the item i itself but its facings k_i and the corresponding width dimension W_i, depth dimension d_i, as well as its profit value π_i. Selected rectangles need to be orthogonally placed in the container and are not allowed to overlap the container limits (Bortfeldt and Winter, 2009). Different constraints are applicable to this problem. First, with regard to the number of reproductions of each rectangle, our problem belongs to the Single-Constrained Knapsack

Problems (c.f. Beasley (2004); Bortfeldt and Winter (2009)). In our case, each rectangle represents a certain facing number and its arrangement of a certain item that needs to be allocated to a single container. We need to apply an upper limit that restricts the maximum size of a rectangle but the item selection included ensures that no lower bound is set (as for doubly-constrained Knapsack problems). The second constraint type is the orientation constraint that determines whether a rectangle can be rotated by 90 degrees to fit onto the container or not (c.f. Lodi et al. (1999)). In our case, the dimension of a rectangle is dictated exogenously because the rotation of the rectangles is not allowed (e.g., because product labels need to be legible and the display is defined). A final differentiation is the guillotine cutting constraint that can be applied to divide the total solution space into parts. A container is divided into sections by using guillotine cuts. Guillotine cuts can be made horizontally or vertically and from one side to the opposite ("edge-to-edge") of the container, whereas one item can only belong to one container (=subsection). Each resulting subsection is considered separately and may be cut again. This procedure reduces the solution space as less combinatorial options are possible. Figure 3.5 depicts a guillotine and non-guillotine approach applied to a two-dimensional shelf. Our application does not allow guillotine cuts as this would reduce the degrees of freedom for how facing and arrangement options can be chosen within the container. The variable dimensions of the rectangle would also not allow meaningful cuts. According to the typology of Wäscher et al. (2007) the 2DSCASP is a Single Large Object Placement Problem that is transformed by the item consolidation to a Single Knapsack Problem.

Specification of arrangement and space constraints for 2DSCASP

We use the relative arrangement formulation of Pisinger and Sigurd (2007) as it meets the requirements of our application summarized above. This ensures proper arrangement of the selected items with their corresponding dimensions. We introduce the auxiliary variable z_i for the assortment decision to simplify the notation with $z_i = \begin{cases} 1 \text{ for } q_i \geq 1 \\ 0 \text{ else} \end{cases} \quad \forall i \in N.$

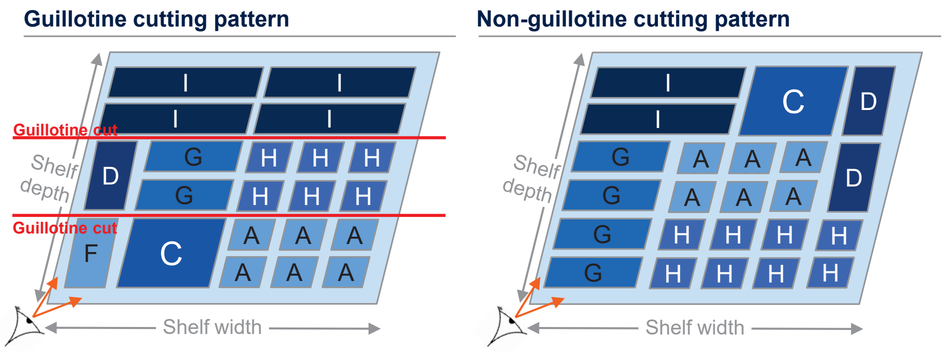


Figure 3.5: Guillotine cutting patterns

The two boolean decision variables l_{ij} and $b_{ij} (i \neq j) \in \{0, 1\}$ $\forall i, j \in N$ determine whether or not item i is arranged to the left of item j (l_{ij}) and/or below (b_{ij}) within the shelf space. Equation (3.9) ensures that all selected items have a position relative (left or/and below) to one another. The binary parameter n_{ij} indicates whether or not item i has to be a neighbor of item j. This allows the definition of joint positioning for related products within a category (e.g. rye bread belongs to the category bread). Equation (3.10) to (3.12) define required neighborhood constraints accordingly. Restriction (3.10) prevents diagonal neighborhoods, whereas restrictions (3.11) and (3.12) ensure that the borders of the item quantities q_i and q_j have adjacent edges for a certain stretch.

The two-dimensional shelf-space limits are represented by S^{width} for the width (x-dimension) and S^{depth} for the depth (y-dimension). Due to the fact that the dimensions of one item only represent the space occupied by the rectangle (rectangularly shaped quantity of one item) in the special case $q_i = 1$, we introduce the auxiliary variables W_i and D_i in restriction (3.13) and (3.14) that represent the space occupied. The parameters for width w_i and depth d_i represent the space occupied by all units of the item i. The decision variables for the coordinates that indicate the lower left position of the item's display are denoted by $coor_i^x$ for the x- and $coor_i^y$

for the y-coordinate. Equations (3.15) and (3.16) ensure that the items N do not overlap each other within the shelf space. Restrictions (3.17) and (3.18) guarantee that no item i crosses the border of the shelf space. In equation (3.19) a maximum facing limit of item i is set. This gives retailers the opportunity to ensure a variety of different products on their shelves by setting the parameter K_i. Equations (3.20) define the domain of the variables.

$$l_{ij} + l_{ji} + b_{ij} + b_{ji} \geq z_i + z_j - 1 \qquad \forall i, j (i \neq j) \in N \tag{3.9}$$

$$l_{ij} + l_{ji} + b_{ij} + b_{ji} \leq 3 - n_{ij} \cdot z_i - n_{ij} \cdot z_j \qquad \forall i, j (i \neq j) \in N \tag{3.10}$$

$$coor_i^x + W_i - coor_j^x \leq S^{width}(1 - l_{ij} \cdot n_{ij}) \qquad \forall i, j (i \neq j) \in N \tag{3.11}$$

$$coor_i^y + D_i - coor_j^y \leq S^{depth}(1 - b_{ij} \cdot n_{ij}) \qquad \forall i, j (i \neq j) \in N \tag{3.12}$$

$$W_i = w_i \cdot q_i^x \qquad \forall i \in N \tag{3.13}$$

$$D_i = d_i \cdot q_i^y \qquad \forall i \in N \tag{3.14}$$

$$coor_i^x + W_i \leq coor_j^x + S^{width}(1 - l_{ij}) \qquad \forall i, j (i \neq j) \in N \tag{3.15}$$

$$coor_i^y + D_i \leq coor_j^y + S^{depth}(1 - b_{ij}) \qquad \forall i, j (i \neq j) \in N \tag{3.16}$$

$$0 \leq coor_i^x \leq S^{width} - W_i \qquad \forall i \in N \tag{3.17}$$

$$0 \leq coor_i^y \leq S^{depth} - D_i \qquad \forall i \in N \tag{3.18}$$

$$K_i \geq k_i \qquad \forall i \in N \tag{3.19}$$

$$l_{ij}, b_{ij}, z_i \in \{0,1\}; q_i^x, q_i^y, q_i^t, q_i, k_i, coor_i^x, coor_i^y \in \mathbb{Z}_0^+ \qquad \forall i, j (i \neq j) \in N \tag{3.20}$$

3.4 Heuristic Approach

The 2DSCTSP is compounded by the NP-hard two-dimensional Knapsack Problem (see Beasley (2004); Kellerer et al. (2004b); Pisinger (2005); Pisinger and Sigurd (2007)) and the NP-hard assignment problem (see Kök and Fisher (2007); Hübner et al. (2016)). The combinatorial complexity of the latter increases very rapidly with the number of items being considered, N, and the shelf-space size S. The total number of possible allocations (Y) to a one-dimensional container can be calculated as expressed by $Y(N,S) = \binom{N+S-1}{S} = \frac{(N+S-1)!}{S!(N-1)!}$. For example, an instance of $N = 5$ and $S = 10$ results in 1,001 and an instance of $N = 50$ and $S = 100$ in $6.7 \cdot 10^{39}$

possible solutions. The two-dimensional problem is even more complex. Here, one obtains up to N^S combinations without any arrangement rules (rectangle, coherent). In the first example the number of combinations increases to 9,765,625 and in the second example to $7.9 \cdot 10^{169}$ combinations. Furthermore, the demand characteristics result in a profit function π_i for each item i (Equation (3.2)), which is non-linear with respect to the decision variables. A metaheuristic approach is therefore developed – a genetic algorithm (GA) suitable for solving real world problems sufficiently and efficiently. We propose a GA paired with a one-dimensional start solution and a bottom-left fill (BLF) heuristic.

Structure and notation We will use the general algorithmic-related terms "container" and "rectangle". A set of small rectangular pieces has to be allocated to a larger rectangle, known as a container. In our application, the container is equal to a shelf and a rectangle represents certain facing and arrangement options of a certain item. The algorithm is developed with object-oriented programming standards to avoid a complex de-/encoding of the solution of each individual object. Instead of complex encryption to represent the different rectangles with their corresponding attributes, the references of the objects are taken into account to execute genetic operations. This ensures that no information is lost while performing the operations and all attributes are accessible at any time. The decoding is implemented as an object function that invokes the operation that arranges the rectangles onto the container and calculates the fitness of the individual. Extensive en-/decoding to or of a binary, a permutation or a value notation are not necessary. We refer to Keijzer et al. (2002), Krishnamoorthy et al. (2002) and Zhang and Wong (2015) for similar implementations of object-oriented evolutionary algorithms. The necessary components for the implementation are detailed in the Appendix with the help of Unified Modeling Language (UML).

Pseudo Code Algorithm 3.1 summarizes the sequential, procedural program flow. This is a specialized heuristic tailored to our problem and based on a genetic algorithm. We apply different settings for the various steps of the algorithm as summarized in Table 3.3.

Algorithm 3.1 Genetic algorithm for 2DSCTSP

```
Require: $N$, $S^{width}$, $S^{depth}$, termination criterion
Ensure: fittest individual over all generations
 1: possibleArrangements ← generateArrangements($N$, $S^{width}$, $S^{depth}$)
 2: population ← generateStartPopulation(possibleArrangements)
 3: allocateProducts(population)
 4: formerFittestIndividual ← calculateFitness(population)
 5: while (termination criterion False) do
 6:     population ← selectAndDuplicateFittest(population)
 7:     population ← crossoverOperation(population)
 8:     population ← mutationOperation(population)
 9:     allocateProducts(population)
10:     currentFittestIndividual ← calculateFitness(population)
11:     population ← elitismOperation(currentFittestIndividual,
            formerFittestIndividual)
12:     formerFittestIndividual ←saveFittestIndividual(formerFittestIndividual,
            currentFittestIndividual)
13: end while
14: printFittest(formerFittestIndividual)
```

Table 3.3: Configuration settings of the genetic algorithm

Steps and methods	Possible settings		
Step 2: Start solution	Random	Adapted one-dimensional solution	
Step 3: Rectangle allocations	Bottom-left fill heuristic		
Step 5: Termination criterion	Runtime	Number of populations	Solution quality
Step 6: Selection	Wheel	Tournament	Rank
Step 7: Crossover	Fixed	Random	
Step 8: Mutation	Probability rate and variance configurable		
Step 11: Elitism	Injection rate of the previous fittest solution		

The algorithm starts with input of the set of items N, shelf dimensions S^{width} and S^{depth}, and the termination criterion. The objective is to find the fittest individual across all generations, which contains the container with the most profitable rectangles.

Step 1 generates the set of possible rectangles for each item, $i \in N$. It takes into account the shelf dimensions S^{width} and S^{depth}. The number of maximum facings K_i $(i \in N)$ is denoted by the shelf dimensions. The possible arrangement options for each item are generated as a result of the maximum quantity in each visible shelf dimension ($S^{width}/w_i \geq q_i^x$, $S^{depth}/d_i \geq q_i^y$) and exclusion of arrangement options that result in prime number defects.

Following *Step 2* generates a start population. We implemented two different options. In the simple case, a random start solution (RSS) is applied. In the advanced version, an adapted one-dimensional start solution (ASS) is generated using the model and solution approach of Hübner et al. (2015). They develop an iterative heuristic that solves a MIP for the assortment and space allocation problem in the first step and, in the subsequent step, updates the demand calculation according to the shelf configuration of the first step. This procedure is repeated until a solution-quality-related termination criterion is met. We extend this approach by using a constraint in the MIP to directly eliminate the prime number facings that exceed one S^{width} or S^{depth}. The arrangement issue that items cannot be allocated to the shelf because of their particular dimensions cannot be included in Hübner et al. (2015). The computed one-dimensional quantities are subsequently transformed to two-dimensional feasible arrangements.

Step 3 allocates the rectangles to containers. So far, the algorithm is composed of a population of individuals where each individual consists of a single container that contains one or multiple rectangles and where each rectangle has an item reference. We use the bottom-left fill (BL-F) pack heuristic to fill up the containers. Hopper and Turton (2001) identified the BL-F as an efficient approach for the two-dimensional packing problem. The BL-F is a modified version of the bottom-left (BL) pack heuristics. The BL algorithm starts with placing each rectangle in the top right corner of the container. From there the rectangle slides as far as possible (without crossing another item) to the bottom and then as far as possible to the left of the container. This movement process is repeated until the rectangle can no longer be moved, i.e., the rectangle collides with another rectangle or the frame of the container. This makes full use of the rectangle. The disadvantage of the BL algorithm is the empty space within the container. In contrast to this, the BL-F algorithm seeks the lowest left position in the container that the rectangle can fit into. This approach makes it possible to occupy what were previously empty spaces but also leads to a higher runtime. Furthermore, Hopper and Turton (2001) show computational benefits if the rectangles are sorted and filled by size $(W_i \cdot D_i)$ in descending order. Note that the algorithm is not forced to completely fill up the shelf space. It may be better to leave free spaces within the shelf due to penalty costs for oversupply in the objective function.

After this, rectangles are allocated and the fitness of each individual is evaluated in *Step 4*. The algorithm is terminated based on maximum runtime, number of populations or solution quality improvements. If the termination criterion in Step 5 is met, the fittest individual is displayed. Otherwise, the loop of Steps 6 to 11 is executed until the termination criterion is met.

Steps 6 to 8 describe the GA operator's selection, crossover, and mutation. The selection operation in *Step 6* intensifies the average fitness of a population through duplication of the fittest and disposal of the weakest individuals. We use different approaches. In Wheel Selection (WS) the selection probability of an individual is calculated by dividing the fitness of a selected individual by the total cumulative fitness of all individuals. This approach ensures that stronger individuals are more likely to be included in the adapted population than weaker ones. Tournament Selection (TS) is based on the comparison of two randomly picked individuals of a population. The individual with the higher fitness score is selected for the adjusted population. All chosen or not chosen individuals remain in the basic population and can be selected again. Rank Selection (RS) reevaluates the fitness of each individual depending on the fitness ranking. The technique takes the rank of the fitness value and not the nominal value into account. A common approach is to rate the worst as fitness 1, the second worst as fitness 2 and so on. The best is rated as N, where N equals the number of individuals considered.

The crossover operation in *Step 7* is a method for interbreeding the individuals of the selected population to form a new offspring population. Crossover is performed with a specified probability rate. The crossover operation can be executed with a fixed number or randomly generated amount of crossover points. The points are most evenly divided depending on the quantity of items, i.e., to build equal sized crossover parts the length of the individual is divided by the amount of crossover points, whereby the last part contains the size of the modulo value. All items between the crossover points alternately remain part of the individuals or swap between the individuals. In the mutation operation in *Step 8*, small segments of the individuals of the new offspring are randomly modified. The purpose of this is to preserve diversity across generations. The mutation probability rate and the variance of the modification can be chosen. During the execution, the new quantity of the item is also randomly transferred to feasible two-dimensional spaces. In *Steps 9 to 12*, all rectangles within each individual are allocated and evaluated. Crossover and mutation operations modify individuals so much that there is a high probability of losing the fittest individual across the following generations. Hence, an elitism method is applied to preserve the fittest individual across the next generations. The overall fittest individual is saved and injected into a population if the fittest individual of this population is not at least as fit as the fittest overall. The fittest individual of this generation is compared with the individual that is fittest overall to determine the new individual that is fittest overall. Then the algorithm returns to *Step 5*.

3.5 Numerical results

In this section we first describe the test setting before then conducting various numerical analyses with simulated data and data from a case study and different variants of the model. We gradually increase the complexity to demonstrate the efficiency of the models and solution approaches step by step. Section 3.5.2 investigates the error range if the solutions of an one-dimensional model (1DSCASP) are transferred to a two-dimensional problem (2DSCASP). The heuristic approaches are analyzed and compared in terms of runtime and solution quality in section 3.5.3. Section 3.5.4 assesses the impact of demand effects and correctly accounts for stochastic demand, space elasticity, and substitution on profit as well as facing changes. Finally, we apply our model to a case study in section 3.5.5. Table 3.4 gives an overview.

Table 3.4: Overview of numerical tests

Section	Purpose	Models	Demand included		Problem sizes	Solution approaches
			Space elast.	Subst.		
3.5.2	Transfer of 1D solutions to 2D problems	1DSCASP, 2DSCASP	✓		small	*exact*: full enumeration
3.5.3	Efficiency analysis of heuristics					
	Comparison of GAs vs. exact approaches	2DSCASP	✓		small	*exact*: full enumeration; *heuristic*: GA variants
	Comparison of GAs	2DSCASP	✓		large	*heuristic*: GA variants
	Comparison of GAs	1DSCASP, 2DSCASP	✓	✓	large	*heuristic*: GA variants, Hübner and Schaal (2017a)
3.5.4	Effect of combining space elasticity and substitutions	2DSCASP	✓	✓	large	*heuristic*: GA variants; ex-post evaluation
3.5.5	Case study	2DSCASP	✓	✓	large	*heuristic*: GA variants; ex-post evaluation

3.5.1 Data generation and test setting

To generalize our analysis, all input parameters are randomly generated within sections 3.5.2 to 3.5.4. We generated parameter values within reasonable ranges derived from literature or from the cooperation with a retailer. There are either sources from empirical studies (e.g., Gruen et al. (2002), Campo et al. (2004) or Aastrup and Kotzab (2009) for the range of substitution rates; Drèze et al. (1994), Desmet and Renaudin (1998) or Eisend (2014) for space-elasticity effects) or from other comparable modeling approaches (e.g., Kök and Fisher (2007), Hübner et al. (2016) for the ranges of profits and over-/undersupply costs). In generating our data sets we thus used conventional practice and followed the suggestions of previous literature. We made the data available at GitHub. These are equally distributed and satisfy the following rules. Each item $i \in N$ has a positive profit $r_i > c_i$, a positive salvage value v_i and positive shortage costs s_i. The ratio pattern between the parameters is defined as $r_i \geq c_i \geq v_i \geq s_i$ with $r \in [20; 25]$, $c \in [4; 9]$, $v \in [4; 9]$ and $s \in [1; 3]$. Hübner et al. (2016) reveal that continuous demand distributions serve as good approximations of discrete demand distributions. It is assumed that demand is normally distributed with an average minimum demand of $\mu_i \in [7; 25]$ and a corresponding coefficient of variation $CV_i \in [1\%; 40\%]$. Modeling demand volatility with CV_i ensures that negative demand cannot occur. The space elasticity β is assumed to vary between $0 \leq \beta \leq 0.40$ (cf. Eisend (2014)). According to Campo et al. (2004) the OOA substitution rates are suitable for providing approximations for OOS substitution rates. Without compromising the general applicability of our model, we assume that the substitution rates for OOA and OOS are the same, i.e. $\gamma_{ji}^{OOA} = \gamma_{ji}^{OOS}$, $\forall i, j$; $j \neq i$. To simplify, we denote the probability that an unavailable item i gets substituted by the aggregated substitution rate λ_i and assume that this rate is split equally among all other items such that $\gamma_{ji}^{OOA} = \gamma_{ji}^{OOS} = \frac{\lambda_i}{N-1}$, $\forall i, j$; $j \neq i$. To focus on the core demand effects, we assume that all items have a uniform size with an identical depth and width of $d_i = w_i = 1$ and a shelf stock per facing of $q_i^t = 1$. If not stated otherwise, we considered 100 randomly generated instances for each problem setting. For all instances of a problem setting the assortment size N and shelf size $S^{width} \times S^{depth}$ are assumed to be identical. All numerical tests were conducted on a Windows Server 2012 R2 64-bit with two Intel Core E5-2620 processors and 64-GB memory. The tests are implemented in VB.net (Visual Studio 2015) and GAMS 24.1.

3.5.2 Transfer of one-dimensional solutions to two-dimensional problems

The one-dimensional solution is easier to obtain, but it may not be a feasible solution due to arrangement and prime number defects (also see section 3.2). This analysis serves to assess the error impact of transferring solutions obtained by models that are based on one-dimensional shelf space to settings with two-dimensional shelf space. The best case would show that one-dimensional solutions are a good approximation for the two-dimensional problems. We solve the following three models exactly: 1DSCSP includes prime numbers, 1DSCSP$_{\text{ex-prim}}$ excludes prime numbers and 2DSCSP. Six test problem settings are defined with a varying total number of items (N), quadratic shelf sizes with $S^{width} = S^{depth}$ and an upper limit on the facings (K_i=$S^{width} \times S^{depth}$). The randomly generated demand of each item is set to $[1; 6]$ for sets 1 to 4 and $[1; 9]$ for sets 5 to 6. These problem sizes ensure computationally tractable runtimes. For each problem 100 instances are randomly generated by using the data ranges provided above.

Frequency of defects Table 3.5 reveals the occurrence of defects. The arrangement and prime number defect of the one-dimensional solution appear in all settings. Arrangement defects can be found in 14% and prime number defects in 32% of the cases. In some cases both defects exist. Consequently, one or both defects occur in 41% of cases.

Table 3.5: Analysis of arrangement defects of 1D solutions, average of 100 instances

Number of items N	4	5	6	7	6	6	Total
Total shelf space ($S^{width} \times S^{depth}$)	3 x 3	3 x 3	3 x 3	3 x 3	4 x 4	5 x 5	
Arrangement defect [%]	15	9	14	2	18	24	14
Prime number defect [%]	27	29	23	26	56	33	32
Total cases with defect/s [%]	37	32	31	28	66	51	41

Profit impact of defects Table 3.6 summarizes the profit impact due to the required arrangements on a two-dimensional shelf. It compares the exact solutions of the 2DSCSP with the 1DSCSP. The latter do not consider the rectangular arrangement and prime number requirements, whereby 41% of 1DSCSP solutions are non-viable solutions for the 2DSCSP. These additional requirements in the two-dimensional problem lower the profit by 0.8% on average. Hence, this expresses the total profit impact caused by the rectangular arrangement and prime number constraint. In other words, theoretically the feasible solution yields 0.8% lower profit compared to the non-viable solution without prime number and arrangement constraints.

Table 3.6: Profit comparison of 2DSCSP vs. 1DSCSP, 100 instances

Number of items N	4	5	6	7	6	6	Total
Total shelf space ($S^{width} \times S^{depth}$)	3 x 3	3 x 3	3 x 3	3 x 3	4 x 4	5 x 5	
Average profit[1]	0.991	0.989	0.992	0.994	0.992	0.997	0.992

[1] Calculation: 2DSCASP profit / 1DSCASP profit

Arrangement and prime number defect To quantify the individual profit impact for each type of defect, we compare the 1DSCSP and the 1DSCSP$_{\text{ex-prime}}$ where prime numbers are excluded. The results in Table 3.7 depict that the prime number defect leads on average to a 0.5% lower profit. Hence, imposing the arrangement constraints results in 0.3% lower profits.

Table 3.7: Profit comparison of exact solutions: 1DSCSP$_{\text{ex-prime}}$ vs. 1DSCSP, 100 instances

Number of items N	4	5	6	7	6	6	Total
Total shelf space ($S^{width} \times S^{depth}$)	3 x 3	3 x 3	3 x 3	3 x 3	4 x 4	5 x 5	
Average profit [1]	0.995	0.991	0.996	0.995	0.995	0.999	0.995

[1] Calculation: 1DSCSP$_{\text{ex-prime}}$ profit / 1DSCASP profit

Summary The one-dimensional solution is easier to obtain, however, it is not a feasible solution due to arrangement and prime number defects. These requirements impact optimal allocation. The optimal item quantities of the two-dimensional problem differ from those of the one-dimensional problem. Due to the additional constraints in the 2DSCSP, the total profit will always be equal or below the 1DSCSP. Corresponding one-dimensional solution approaches are not readily appropriate methods for solving two-dimensional problems. It has to be considered that in cases where the one-dimensional solution does not fit onto the two-dimensional shelf space, quantities of items need to be adjusted. It is not obvious which item quantities have to be increased or decreased to achieve the best feasible solution (e.g., via simple rounding or greedy heuristics). The decision process becomes even harder when substitution effects in the model are considered due to the demand interdependencies between the items. The consequence of this is that the loss in solution quality would be significantly higher if using the one-dimensional model.

3.5.3 Efficiency analysis of heuristics

Comparison of heuristics with space elasticity vs. exact approaches

This section examines the efficiency of the heuristic developed. To validate the GA it is compared to a full enumeration (FE) applied to smaller problem sizes. The GA is executed as described in section 3.4 with a random start solution and the selection methods WS, TS and RS. The random crossover operation and the elitism operation are applied. Mutation operations are not reasonable and can be neglected due to the small size of problem instances. Pretests have shown that a termination criterion of 100 seconds is more than sufficient to return the best solution.

Runtime Table 3.8 summarizes the computation time. For the FE it shows that the median runtime increases between four to ten times if the set is extended by only one additional item. A similar magnitude is recognizable when the space is extended gradually. For the different implementations of the GAs the runtime is significantly lower and the increases for extended problem sizes are much lower. Furthermore, the runtimes of the WS and TS are below 4 seconds on average in all instances. The smallest median execution time across all 600 problem instances was achieved via the TS, and is over 120 times faster than the FE.

Table 3.8: Median runtime of different approaches, in seconds, 100 instances

Number of items N		4	5	6	7	6	6	Total
Total shelf space ($S^{width} \times S^{depth}$)		3 x 3	3 x 3	3 x 3	3 x 3	4 x 4	5 x 5	
FE	*Median*	0.728	2.530	22.593	135.332	72.136	134.920	19.988
GA WS	*Median*	0.171	0.327	0.547	1.028	2.106	3.117	0.788
GA TS	*Median*	0.036	0.087	0.130	0.220	0.376	0.710	0.164
GA RS	*Median*	1.362	2.661	4.049	6.506	13.919	36.471	5.425

Solution quality The solution quality of the GA methods compared to the optimal solutions is shown in Figure 3.6. The boxplots show that the median is 100% in all three variants. Additionally, the data evaluation reveals that the average solution quality exceeds 99% in all cases. The first quantile is equal to 100% for the WS, and is greater than 97% for the TS as well as the RS.

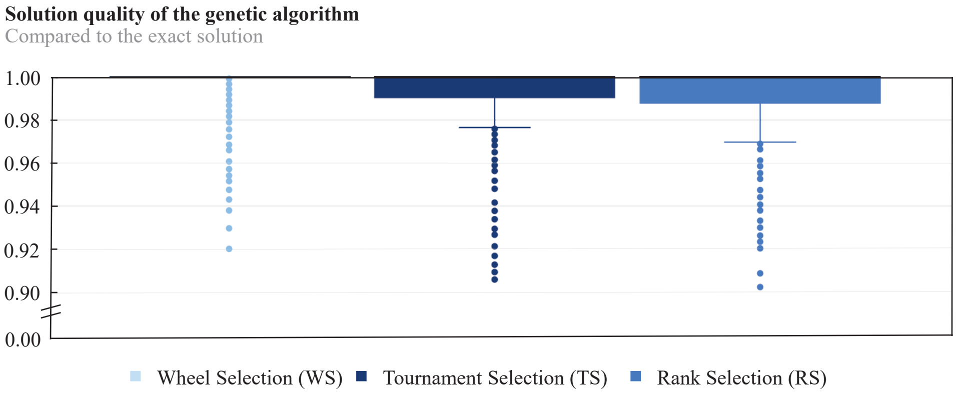


Figure 3.6: Solution quality of different selection operations in comparison to the exact solution

In reference to the solution quality the WS is slightly better than the TS and the TS is slightly better than the RS. To figure out what selection method is better suited for more extensive problem settings the execution time as a ratio of the solution quality achieved is examined more precisely. Problems five and six of Table 3.8 are considered which together consist of 200 problem instances. Figure 3.7 shows the median solution quality of the best individual solutions achieved up until the time shown on the x-axis. The curve of the RS obviously increases more slowly than the curves of the other selection methods and the solution quality of the TS increases slightly faster compared to the WS. In conjunction with the results of Table 3.8 that have been discussed RS does not appear to be a suitable selection approach.

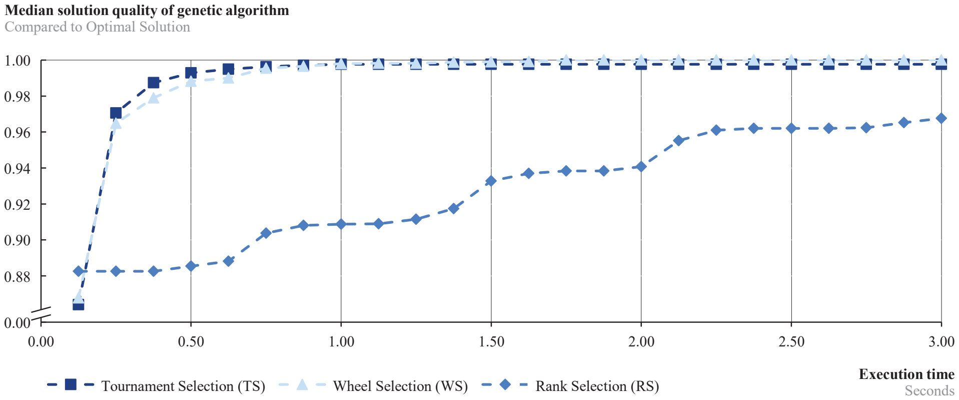


Figure 3.7: Median Solution Quality depending on Execution Time

Summary The results show that the median runtime for the full enumeration increases exponentially as the number of items N and shelf space $S^{width} \times S^{depth}$ increase. In comparison, the runtime of the GAs is lower and increases only very moderately. Furthermore, they achieved a close to optimal average solution quality of at least 99.1% in all three cases. In terms of runtime and solution quality the TS is the most promising approach for larger problem settings. This is due to three facts. First, the runtime increase of the TS is lower compared to the other selection operations. Second, Table 3.8 shows that the TS has the shortest average computation times over all problem settings. Third, the median of the solution quality of TS is equal to WS, and the solution quality of TS compared to WS increases slightly faster.

Efficiency analysis of heuristics with space elasticity for extensive problem settings

Three more extensive problem settings of practice-relevant size are tested. The number of products and shelf space are increased in steps and the number of facings is increased to $K_i = 30$. The positive demand of an item has a uniform distribution within $[1;30]$. All other parameters are applied as above. The maximal runtime is bound to 500 seconds. Here we use the random start solution (RSS) and adapted start solution (ASS) which are described in section 3.4. The ASS uses the one-dimensional solution of Hübner et al. (2015).

Runtime Table 3.9 once again shows that the TS is faster than the WS (GA WS vs. GA TS) for the smaller instances with 20 items. If the median runtime is close to the limit applied of 500 seconds, it means that in many cases the best solution has not yet been found due to the termination criteria. This means that the GA would still improve the solution with longer runtimes. This is the case for all GA WS applications and for the larger GA TS applications with 50 and 100 items. However, a significant runtime improvement can be obtained by applying the ASS. This makes it possible to obtain solutions within a few seconds, even for larger problems. Where the ASS TS has significantly shorter runtimes than the ASS RS.

Solution quality There is no exact solution available that can be generated in reasonable computation time. We therefore use a benchmark. We use the solutions of the 1DSCASP$_{\text{ex-prime}}$ problem which exclude non-viable prime numbers but might be still an infeasible approach in terms of the arrangement options. Our calculations in Tables 3.6 and 3.7 allow the

Table 3.9: Median runtime for larger problems, in seconds, 100 instances, rum time limit 500 seconds

Number of items N		20	50	100	Total
Total shelf space ($S^{width} \times S^{depth}$)		15 x 15	20 x 20	25 x 25	
GA WS	*Median*	430	412	466	441
GA TS	*Median*	117	480	482	466
GA ASS WS	*Median*	<1	2	5	2
GA ASS TS	*Median*	<1	1	3	1

conclusion that the 1DSCASP$_{\text{ex-prime}}$ is a suitable upper bound. For small instances the gap compared to the 2DSCSP is 0.3% on average. Scatterplot 3.8 shows the efficiency of the ASS methods. The ASS methods met the benchmark in almost all of the 300 test instances. The 300 test instances belong to the three test settings shown in Table 3.9 in ascending order of the problem size and are equally split (1-100, 101-200 and 201-300). The ASS TS method only missed the optimal solution in 3 cases. WS and TS with a random start solution demonstrate much lower performance for the larger test instances (201 to 300), as here the solution quality suffers from a deficit in runtime.

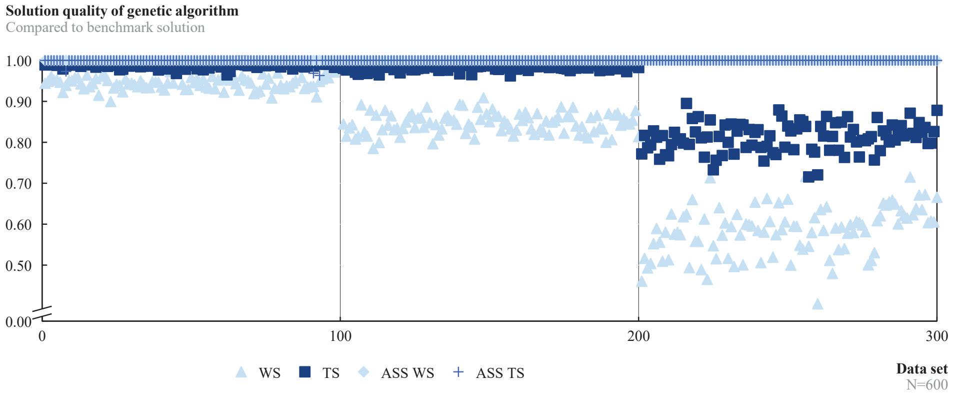


Figure 3.8: Profit levels of GA variants, *in % of benchmark approach, across all 300 extensive problem settings*

Summary The FE only has acceptable runtimes for very small problem sizes. This means it is not an appropriate procedure for real-world problems. The GA configured with the selection operation WS and TS performs well for small and medium problem sizes. For more extensive problem settings the GA with a random start solution also leads to unsuitable runtimes. The increasing number of products and larger shelf space generate higher degrees of freedom. This results in greater opportunities for allocating the optimal item quantities onto the shelf space. As a result the GA mostly only faces the prime-number defects in more extensive problem settings which makes the ASS an appropriate approach for solving them.

Efficiency of heuristics with space elasticity and substitutions

In this section the model is extended by the substitution effects. To obtain a first indication that the GA is suitable to account for substitution effects GA TS and GA ASS TS are compared to the heuristic approach AMIOAS (Algorithm for Mixed-Integer Optimization of Assortment- and Shelf-space problems) of Hübner and Schaal (2017a). Since this approach is only appropriate for the 1DSCASP with substitution, the GA is also applied to this setting with large problem settings. A second comparison with the GA TS and the GA TS ASS is applied to the two-dimensional problem.

Algorithm suitability test for substitution effects Tables 3.10 and 3.11 summarize runtime and the solution quality of the GA TS for the 1DSCASP. The model of Hübner and Schaal (2017a) is therefore a special case as it only yields feasible one-dimensional solutions as it does not take into account two-dimensional shelf space. The median solution quality of GA TS compared to AMIOAS is 99.2% and ranges between 97% to 99.9%. Despite the higher runtime and slightly lower solution quality for most problem settings, the GA TS has demonstrated appropriate performance for addressing substitution effects.

Table 3.10: Runtime of GA TS for 1DSCASP, in seconds, 100 instances

Number of items N	20	50	100	Total
Total shelf space ($S^{width} \times S^{depth}$)	225 x 1	400 x 1	625 x 1	
Average	535	1,208	2,153	1,301
Median	471	1,138	1,998	1,173
Min	120	441	1,773	119
Max	1,683	1,973	3,589	3,589

Table 3.11: Median solution quality GA TS vs. AMIOAS for 1DSCASP, 100 instances

Number of items N	20	50	100	Total
Total shelf space ($S^{width} \times S^{depth}$)	225 x 1	400 x 1	625 x 1	
Average[1]	0.997	0.992	0.969	0.986
Median[1]	0.999	0.993	0.970	0.992
Min[1]	0.984	0.978	0.942	0.942
Max[1]	1.000	0.999	0.986	1.000

[1] Calculation: GA TS profit / AMIOAS profit

Algorithm with a refined start solution to meet substitution effects

Due to the fast convergence times of Hübner and Schaal (2017a)'s algorithm, we will use an adjusted version of ASS in which the AMIOAS results are used as a start solution. Table 3.12 presents the percentage variance of the solution quality between the GA TS ASS and the GA TS after the limited runtime of 1,000 seconds. It shows that the GA TS ASS has achieved a 15.7% higher median on average for the most extensive problem setting. The difference between the two approaches is in evidence with a closer look at the time at which the best solution was found. The average median time of the smallest problem setting in Table 3.10 is 471 seconds for the GA TS, compared to 11 seconds for the GA TS ASS.

Table 3.12: Profit difference between GA TS ASS and GA TS for 2DSCASP, in %, 100 instances

Number of items	20	50	100	Total
Total shelf space ($S^{width} \times S^{depth}$)	15 x 15	20 x 20	25 x 25	
Average	0.3	1.9	15.6	5.9
Median	0.3	1.8	15.7	1.8
Min	-1.2	-0.6	6.6	-1.2
Max	1.6	3.8	24.7	24.7

[1] Calculation: (AMIOAS / GA TS profit profit -1) × 100

Summary The numerical results with the integration of substitution effects has shown that the heuristic developed is suitable for addressing these effects. The second analysis has shown that an intelligent start solution is advisable with substitution effects, too.

3.5.4 Effect of combining stochastic demand, space elasticity and substitution

Because this is the first integrated stochastic model for two-dimensional shelf spaces that accounts for space elasticity and substitution, this section illustrates the difference vis-à-vis the existing two-dimensional model of Geismar et al. (2015) who do not account for demand effects. Total profits and shelf quantity assignments are compared. The parameters CV and β cover the values 0 and 0.35 with an interval of 0.05. The substitution rates considered range between 0 and 0.7 in 0.1 increments. All resulting combinations of the three parameters are evaluated. To investigate the impact of ignoring stochastic demand, space elasticity and/or substitution, a retailer is considered who makes assortment and facing decisions by assuming $CV = \beta = \lambda = 0$, while in reality there are $CV > 0$, $\beta > 0$ and

$\lambda > 0$. To do so, we first run the model with $CV = \beta = \lambda = 0$ and evaluate ex-post the results with the actual demand effects with $CV > 0$, $\beta > 0$ and $\lambda > 0$. This result is compared with an optimization run where the actual values of CV, β and λ are directly applied. This allows to compute the impact of incorrect demand assumptions on assortment and facing decisions as well as the profit.

Figure 3.9 shows that the retailer gains up to 78% more profit on average (i.e., when $\beta = 0.35$, $\lambda = 0.80$ and $CV = 0.35$). Additionally, Figure 3.10 shows that up to 100% of all items get different facing quantities if stochastic demand, substitution and space-elasticity effects are correctly taken into account. It becomes clear that all three demand effects need to be considered jointly.

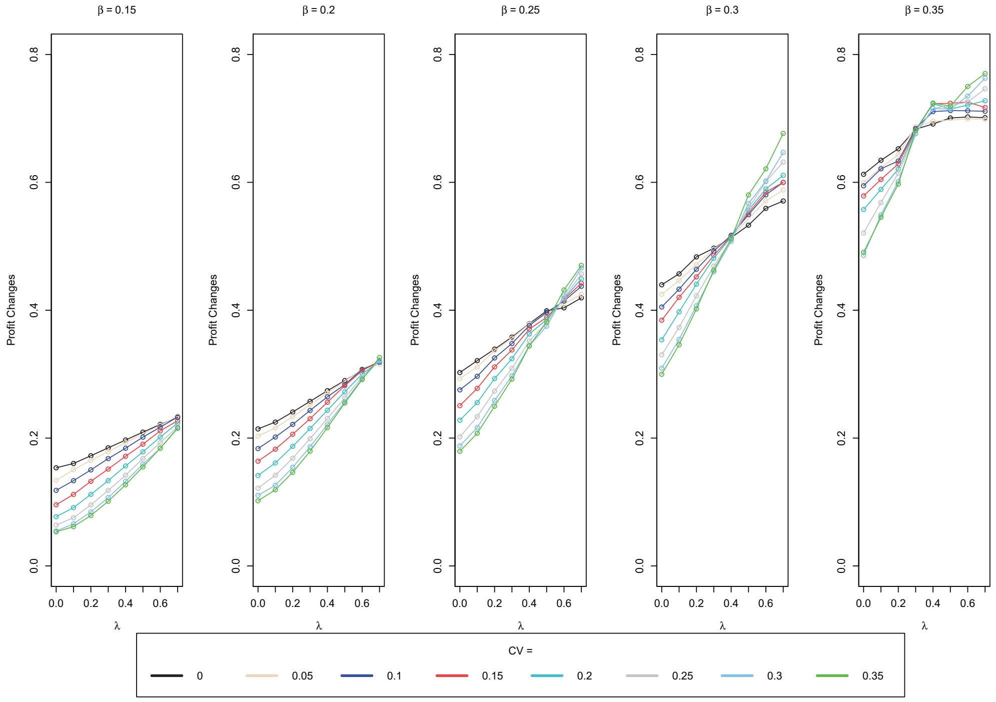


Figure 3.9: Profit changes

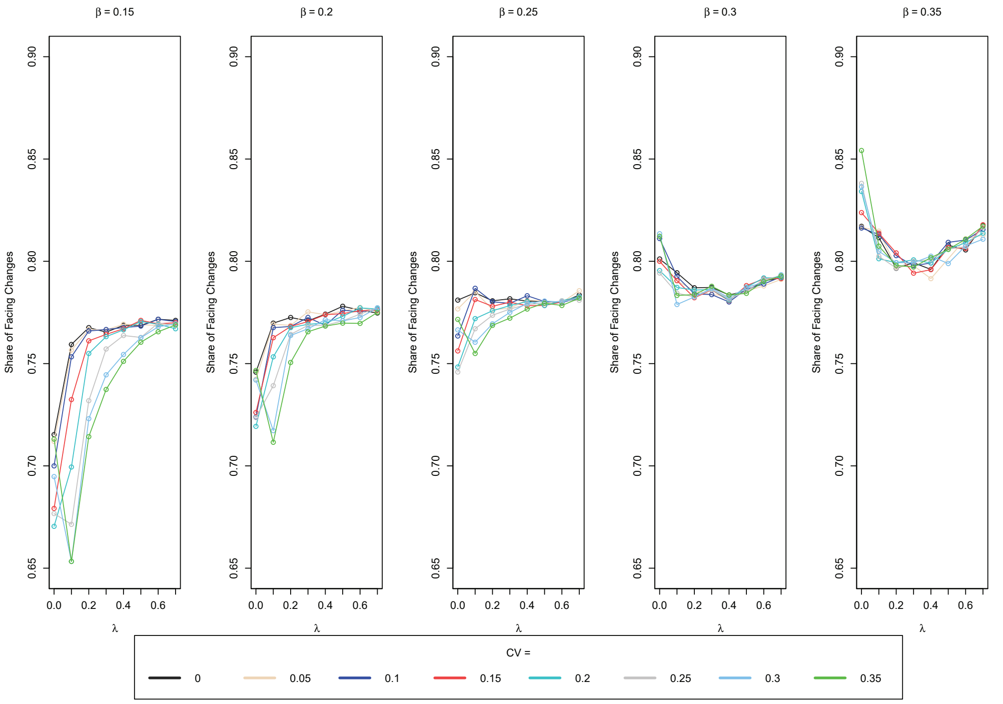


Figure 3.10: Share of facing changes

3.5.5 Case study

After having shown that 2DSCASP can be efficiently solved to near-optimal results within very short runtimes, it will be applied on a real data set in this section. The daily sales data of an assortment of 21 varieties of bread roll were collected at one of Germany's largest retailers. Substitution rates between the items were identified using customer surveys. We interviewed $n = 2,412$ customers and asked them which substitute they would purchase if their first choice were unavailable. Asking customers whether the product they bought was really their first choice also captured substitute purchases for items that are actually unavailable. The substitution rates between two items i and j were then obtained by $\frac{\text{No. of customers purchasing } j \text{ as substitute for } i}{\text{No. of customers choosing } i \text{ as first choice}}$. We had at least 30 interviewees for each item. Substitution rates per substitute amounted to up to 40%. The exact parameters are subject to confidentiality obligations. The minimum daily demand δ_i^{min} varied between 1 and 25 units with a variation coefficient of $CV_i = [40\%; 152\%]$. Sales prices r_i ranged between €1.5 and €3.95 with unit costs of c_i=[€0.32; €1.02]. The penalty costs s_i are set at zero. Because the items are perishable and our case study retailer has no further use for the items after the stated expiry date, the salvage value v_i is assumed to be zero. The retailer does not offer special discounts for items close to the expiry date. This is due to the short shelf life of bakery products (see also Kök and Fisher (2007) and Hübner et al. (2016) who analyze settings with no salvage values). Beyond the specific setting in our case study we use non-zero salvage values in the numerical analysis above to generalize our findings. To maintain a certain diversification on the shelf, the number of facings ranges between 1 and 30, whereby the shelf depth S^{depth} is 0.50m and the shelf width S^{width} is 1.20m. Currently, the retailer assigns shelf space to the 21 products based on sales proportions, i.e., without explicit margins taken into account, demand volatility, space elasticity or substitution. The space elasticity β ranges in the sensitivity analysis between 0% and 30% in 5% increments. Additionally, 17% is added which is the average demand increase driven by space elasticity (Eisend, 2014).

Table 3.13 shows the profit potential from applying our model. The retailer can increase profits by up to 15% depending on the assumed space elasticity. Furthermore, it can be seen that optimized assortments contain up to 38% fewer items than the current assortment. The increase in space elasticity leads to more shelf space for the most profitable items. This results in smaller assortments and an increasing number of items with facing changes.

As a result of the remaining uncertainties of determining the parameters, we analyzed the profit potential together with the retailer depending on parameter robustness based on the average space elasticity of Eisend (2014). Moreover, we investigated the options for defining the appropriate shelf space for the bread roll category. We applied a sensitivity analysis for that purpose. To do this, the estimated substitution effects λ, variation

Table 3.13: Results of case study

	Space elasticity β							
	0%	5%	10%	15%	17%	20%	25%	30%
Profit potential[1]	5.3%	5.6%	7.3%	8.1%	12.5%	11.2%	12.9%	14.8%
Assortment size[2]	86%	86%	81%	76%	71%	62%	62%	62%
Facing changes[3]	62%	67%	76%	81%	81%	86%	95%	95%
SD facing changes[4]	0.70	1.35	1.72	1.96	2.03	2.18	2.05	2.28

[1] Calculation: (2DSCASP profit / 2DSCASP* profit)-1

[2] Optimized assortment size as a share of current assortment size

[3] Share of items with facings different to current facings

[4] Standard deviation of absolute facing quantity changes

coefficients CV and shelf space S are individually adjusted pro rata between 60% and 140% in 10% increments, whereas the other parameters remained unaffected. To ensure in-store practicability, 20% increments are used for the shelf space. Table 3.14 shows that in a higher existing parameter ratio, substitution effects λ and shelf space S create more profit, whereby variation coefficients CV lead to decreasing profit. The following profit-oriented managerial insights can be concluded for the retailer:

i) Inaccuracies in estimating the substitution effects have a slight impact on profit.

ii) Slight deviations in determining the variation coefficients significantly affect profit.

iii) If there is additional shelf space available, the retailer should enlarge the shelf space size for the bread roll category to increase profit.

Table 3.14: Profit potential depending on parameter robustness

	Existing parameter ratio								
	60%	70%	80%	90%	100%	110%	120%	130%	140%
λ	-4.2%	-3.0%	-2.0%	-0.6%	0.0%	0.7%	1.6%	2.5%	3.4%
CV	11.3%	8.9%	6.3%	3.3%	0.0%	-3.5%	-7.3%	-11.1%	-14.8%
S	-36.1%	-	-17.3%	-	0.0%	-	15.6%	-	27.7%

3.6 Conclusion and outlook

Conclusion Our model integrates assortment and shelf-space optimization and takes into account stochastic demand, substitution and space elasticity. It supports retailers in creating a planogram for two-dimensional shelves by determining optimal assortments and shelf quantities as well as the adjacently rectangular arrangement of each item's facings. It is an integrated approach that simultaneously solves the four subproblems item selection, shelf quantity, facing arrangement, and item arrangement. Previous shelf planning literature focuses on regular shelf types where customers just see the foremost unit of an item. Solutions obtained for regular shelves cannot easily be transferred to two-dimensional and tilted shelves. The combinatorial complexity of the model leads to a rapid increase in runtime with the number of items and the shelf-space size. We developed a problem-specific specialized heuristic that is based on a genetic algorithm. In the numerical results we have shown that

i) one-dimensional solution approaches of current literature are not readily appropriate methods for solving the two-dimensional problems,
ii) our algorithm efficiently yields near-optimal results as our specialized heuristic achieves >99% of the exact approach on average for small instances,
iii) neglecting stochastic demand, substitution and space elasticity leads to 78% lower profits and changes in facings of up to 85%, and

iv) in a numerical analysis with the scope of one of Germany's largest retailers, it may be possible to increase profits by up to 15%.

Future areas of research Various opportunities exist for further research. Our model is based on several assumptions that could be relaxed in the future, e.g., we assumed that substitution takes place across one round only. Future models could account for several rounds of substitution, if substitutes are not available. The extension of our model is linked to the further development of solution approaches. Further heuristics can be developed to approach the stochastic non-linear problem. Another topic of research interest is combination of the tactical problem described in this paper with *operational topics*, such as shelf refilling, order management and inventory accuracy (cf. e.g., DeHoratius and Raman (2008); Curseu et al. (2009); Donselaar et al. (2010); DeHoratius and Ton (2015); Xue et al. (2017); Sharma et al. (2019)). Further extensions in this area would address additional operational restrictions in backroom inventory and delivery frequency (cf. e.g., Eroglu et al. (2013); Holzapfel et al. (2016)). Finally, the question of how a *multi-store environment* can be taken into consideration requires investigation. For example, Bianchi-Aguiar et al. (2015) developed an approach to replicate a standard planogram for several stores of a retail chain. A holistic multi-store approach would also consider the potential impact of store segmentation on the efficiency of supply chain processes. The model and solution approach presented in this paper has laid the foundation for these research questions.

4 Operational Patient-Bed Assignment Problem in Large Hospital Settings including Overflow and Uncertainty Management

Co-autors: Manuel Walther, Alexander Hübner, Heinrich Kuhn

Accepted in *Flexible Services and Manufacturing Journal* on 3rd of December 2018
(in press, DOI: 10.1007/s10696-018-9331-0)

Abstract Managing patient to bed allocations is an everyday task in hospitals which in recent years has moved into focus due to a general rise in occupancy levels and the resulting need to efficiently manage tight hospital bed-capacities. This holds true especially when being faced with high volatility and uncertainty regarding patient arrivals and lengths of stay. In our work with a large German hospital we identified three main stakeholders, namely patients, nurses, and doctors, whose individual objectives and constraints regarding patient-bed allocation (PBA) lead to a potential trade-off situation. We developed a decision support model that tackles the PBA problem considering this trade-off, while also being capable of handling overflow situations. In addition, we anticipate emergency patient arrivals based on historical probability distributions and account for uncertainty regarding patient arrival and discharge dates. We develop a greedy look-ahead heuristic which allows for generating solutions for large real-life operational planning situations involving high ratios of emergency patients. We demonstrate the performance of our heuristic approach by comparison with the results of a near-optimal solution achieved by Gurobi's MIP solver. Finally, we tested our approach using data sets from the literature as well as actual clinic data from our case study hospital, for which we were able to reduce overflow by over 96% while increasing overall utilization by 5%.

4.1 Introduction

This paper deals with the operational planning question of assigning incoming patients to specific rooms and beds upon their arrival at the hospital. This so-called patient-bed allocation (PBA) problem has been gaining more and more attention in recent years after a basic version of the problem was formulated by Demeester et al. (2010). Based on this seminal work, related research was mostly directed at either improving the computational efficiency (see for example Bilgin et al. (2012) or Range et al. (2014)) or proposed ways to incorporate upstream planning problems such as surgery or elective patient scheduling (see for example Ceschia and Schaerf (2016)).

In our joint project with a large German hospital we identified several challenges with respect to the PBA problem that have to be dealt with in real-life situations including emergency and elective patients of all major disciplines.

First and foremost, large hospitals with 500 or more beds covering all major disciplines exhibit high ratios of emergency patients, e.g., up to 90% in internal disciplines such as cardiology and gastroenterology. Due to the nature of the patient clientele in these hospitals (inherent multimorbidity, unknown medical history, etc.) it is oftentimes not possible to accurately determine the actual length of stay (LOS) of a patient once they arrive as well as throughout their stay.

Second, a shift in demographics as well as advances in medical technologies are forcing hospitals to operate as cost-efficiently as possible. This leads to high overall bed occupancy levels which in turn may more often lead to situations in which bed capacities are insufficient. To minimize such overflow situations while keeping bed occupancy levels high, a common approach is to pool bed capacities across similar medical disciplines to create a balancing effect across the associated wards (see for example Hübner et al. (2015) and [2018]). However, as opposed to single wards with ten to twenty beds, managing operational patient bed assignments within a set of designated wards comprising more than a hundred beds leads to a highly complex planning problem which typically cannot be dealt with efficiently by conventional planning approaches, e.g., a dedicated bed planner who manually assigns patients to beds.

Third, there is a need to adapt patient-bed allocations ad-hoc to changes, as any plan made at a certain point in time is likely to be obsolete only a few hours later due to new emergency arrivals, sudden complications after surgery, or new diagnostic findings (see for example Hulshof et al. (2016)). In practice, this means that the decision problem has to be solved whenever there is a change in the system which merits the physical allocation of a newly arrived patient or a patient waiting in an overflow area to a bed. The large hospitals considered in this paper are deciding on this issue several hundred times a day.

Fourth, three major stakeholders have to be kept in mind, namely patients, nurses, and doctors. Specifically, it is important to make the stay for patients as comfortable as possible while simultaneously respecting patient-specific constraints, balancing the workload for nurses, and making it as efficient as possible for doctors to do rounds.

In essence, this leads to an assignment problem that respects the diverse interests of patients, nurses, doctors and hospital management while simultaneously considering medical, gender, and capacity constraints. Hence, there is a need for a PBA system which is capable of anticipating future developments while at the same time being able to provide quick online recommendations for patient-bed allocations within seconds when prompted (see for example Hulshof et al. (2012)).

In this regard, the present paper proposes a new modeling and solution approach to the PBA problem that incorporates stakeholder-specific objectives for patients, nurses, and doctors. In addition, the paper provides a greedy look-ahead heuristic that allows for flexible bed allocations while managing overflow situations and anticipating future arrivals of elective and emergency patients. The findings and insights discussed herein are not limited to the German health-care system but may well be of importance to any large hospital setting faced with the above-described circumstances.

The remainder of this paper is structured as follows: Section 4.2 provides a detailed problem description discussing relevant literature and further elaborates on the specific contribution of this paper. Section 4.3 lays out the modeling and solution approach. Section 4.4 then provides numerical examples. In particular, we compare the results of our heuristic solution approach with the results of a near-optimal solution achieved by Gurobi's MIP solver for selected problem instances. Furthermore, we test our approach with real-life data from a large hospital in Germany and use several sensitivity analyses to investigate solution quality and run time required. In addition, we further test our approach with data from the literature (see Demeester et al. (2010)). Finally, Section 4.5 presents a summary of the main results and gives an outlook on possible future avenues of research.

4.2 Problem description, related literature and contribution

To understand the main objectives of the patient-bed allocation process in a hospital we interviewed nurses, doctors, and hospital management of our case-hospital. The following subsections describe the general planning problem and related literature as well as open research questions that we tackle.

4.2.1 General planning problem

Operational bed occupancy management in hospitals comprises two inherently different planning problems, namely patient admission scheduling (PAS) and patient-bed allocation (PBA). It should be noted, that in the literature these expressions have been used with varying definitions. We consider the PAS problem as merely comprising the problem of scheduling elective patient admission dates. The PBA problem, however, relates to the problem of allocating a physical room and bed to a patient. In large hospitals with more than 500 beds and a high rate of emergency arrivals the two decision problems are typically solved in a hierarchical manner for reasons set out below.

In a first step, the goal of a PAS system is to ensure a high and balanced utilization of the available bed capacity over time. In principle, four patient classes need to be considered. Namely, elective patients and emergency patients who are already physically available in the hospital, as well as planned elective patients and future emergency patients who are already scheduled to or anticipated to arrive in the future, respectively. Figure 4.1 shows a schematic example for a typical PAS situation and depicts the number of beds occupied by or reserved for the afore-mentioned patient classes for the first night of the planning horizon, i.e., a Monday night, and on each of the consecutive 13 nights.

On the first Monday a certain number of beds are already physically occupied by elective and emergency patients. These numbers decrease over time as most patients occupying a bed on the first day of the planning horizon will leave the hospital on the following days. Note, these numbers mostly stay stable on Saturdays and Sundays, since no discharges take place on those days.

In addition, a certain number of beds have to be reserved for incoming elective and emergency inpatients which are planned or anticipated to show up in the future. Whatever bed capacity is still available after incorporating

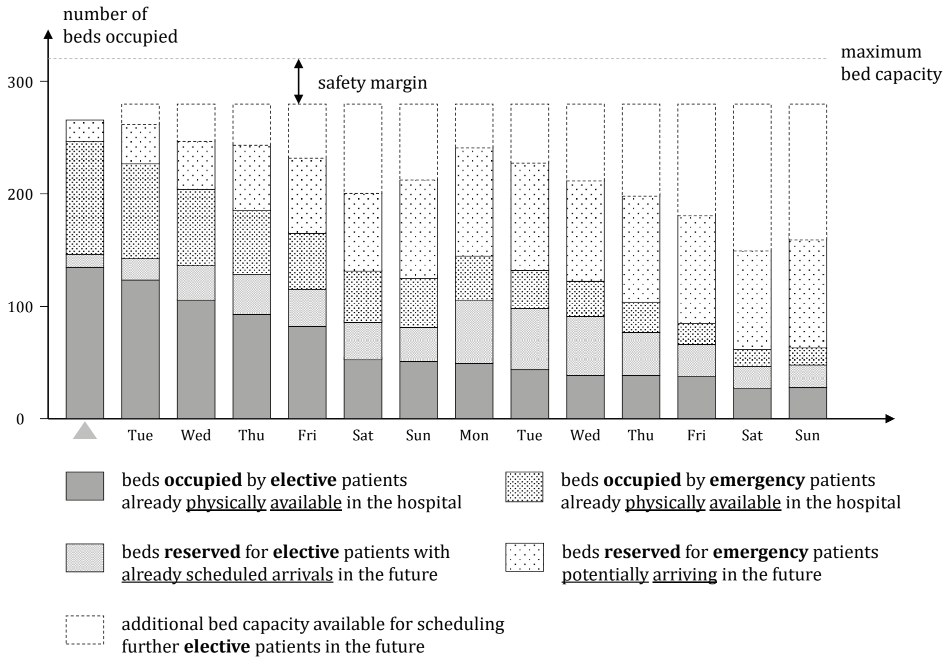


Figure 4.1: Schematic example of a typical planning situation in a large hospital serving elective and emergency inpatients

these four patient classes, respectively, may then be used to schedule additional elective patient arrivals as needed. Please note, patients leaving the hospital on a respective day are already excluded from the bars from a particular day. Newly incoming patients, however, are included in the bars representing the required bed capacity for elective and emergency patients of that day, respectively.

In addition, due to uncertainty regarding the anticipated number of emergency patients as well as LOS changes, a safety margin of beds is established. This is illustrated on the top of all bars in Figure 4.1. The safety margin lowers the available capacity for scheduling elective patients below the maximum possible bed capacity to avoid potential shortages of beds.

Scheduling patients for elective inpatient treatment is usually done a couple of days or even weeks in advance and typically cannot be adjusted at short notice. This is because elective patients have to prepare for their hospital stay well in advance, e.g., plan and schedule transportation, make necessary arrangements at work and/or at home, or simply have to adhere to certain dietary requirements from their physician in the days leading up to a surgery. In addition, scheduling elective patient arrivals is also dependent on master surgery schedules for patients who require surgery. Master surgery schedules as well as staff rosters and staff scheduling are typically fixed weeks in advance which in turn additionally limits the possibilities for rescheduling patients at short notice (see for example Beliën and Demeulemeester (2007), Bilgin et al. (2012) and Gross et al. (2017)). Finally, emergency patients can typically not be deferred to other hospitals once they have been admitted, i.e., once treatment has started.

It is therefore important to distinguish between PAS and PBA (see Figure 4.2). In PAS elective patients need to be scheduled such that the overall ward utilization is balanced and overflow situations are minimized. In the second step, i.e., the PBA, elective and emergency inpatients need to be assigned actual physical rooms and beds upon entering the hospital. In principle, the PBA problem can be viewed as a downstream decision problem with regard to the PAS problem. For the PAS problem it is not necessary to know which bed exactly will be held available for a certain patient as long as it is guaranteed to a certain extent that a bed will be available.

The crucial question in PBA is to determine when the actual allocation takes place and whether or not it should be possible to reserve a specific bed for a specific patient in advance prior to their stay. In hospitals equipped with a large number of beds, however, it oftentimes happens that allocation plans made at the beginning of a specific day are obsolete shorty after, due to changes in lengths of stay, no-shows, sudden complications during surgery or treatment, or simply due to emergency arrivals. Thus, any planning system which fixes patient bed allocations several days in advance will

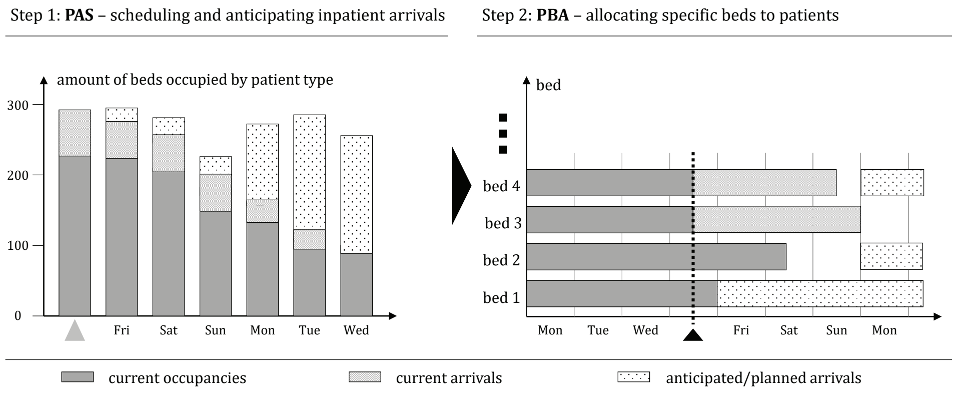


Figure 4.2: Schematic overview of the difference between patient scheduling and patient-bed allocation

inevitably produce allocations that will almost certainly become outdated or even infeasible. Instead, a PBA system should be able to produce a viable allocation for each patient directly when the patient physically needs to occupy his or her room and bed.

In addition, many large hospitals that need to cover all major disciplines exhibit high emergency arrival rates which lead to a higher volatility and uncertainty regarding future occupancy levels. Overflow situations are an inevitable consequence of tight capacities and uncertain demand. In such cases, inpatients need to be assigned to overflow areas such as hallways, emergency- or treatment-rooms, or to other wards outside their dedicated ward space. Staying in such intermediate areas is unpleasant for patients and will always entail additional work for nursing staff and doctors alike, as they typically will not be able to offer the same level of medical assistance. However, the overall LOS of a patient mostly stays the same as necessary surgical procedures and medication treatment will still take place even if a patient is not within his designated ward space. Nevertheless, a bed planner will always try to move patients out of overflow areas whenever the situation allows it to avoid the above-mentioned drawbacks.

As a result of the situation just described, the PBA problem has to be solved several hundred times a day. For each of these planning instances, anticipated future emergency arrivals as well as already scheduled elective inpatients have to be considered. To give an example, a hospital comprising 500 beds and an average LOS of 3 or 4 days requires at least 330 or 250 reruns of the PBA system per day, respectively, i.e., each time a new arrival or departure becomes known to the system.

Objectives In general, patients want their stay to be as pleasant as possible while receiving top-level medical care. This means that patients want to have a room within a designated ward space that caters to their medical needs while avoiding unnecessary room transfers and/or having to wait in an overflow area. In addition, patients want to have pleasant roommates they can get along with and relate to in case they have to share a room. Age difference is a very good indicator for how good patients get along with each other when sharing rooms, especially for longterm stays. This hypothesis was verified by numerous interviews with nursing staff and doctors conducted at our case hospital. Therefore, it is desirable to combine patients of a similar age who have similar illnesses in terms of their specific medical conditions and severity thereof.

As opposed to emergency patients, elective patients are less likely to accept that a room and bed within their respective department is not “reserved” for them upon arrival at the hospital. Emergency patients on the other hand are more willing to accept having to temporarily stay in dedicated overflow areas. In other words, elective patients should in general be preferred when allocating patients to beds during overflow situations. If staying in an overflow area does become necessary, patients wish to be transferred to a “regular room” as soon as possible. In general, it should be noted that elective and emergency patients get the same treatments and the same amount of medical care. The above-described focus on elective patients with regard to patient satisfaction is mainly due to the fact that elective patients will change hospitals for their surgery or treatment if their subjective opinion of a hospital suffers, which would be detrimental to any hospital's reputation.

Doctors are typically bound to a specific department, i.e., a specific medical specialty. In order to facilitate doing rounds and patient visits, it is essential to minimize walking distances for doctors.

One of the main issues when managing patient-bed allocations with regard to nursing staff is creating a balanced workload. This is especially important as nurses are typically dedicated to specific wards in well-coordinated teams, which are used to working with each other and therefore cannot easily be transferred to other wards.

Constraints When trying to optimize PBA, the following hard constraints are typically taken into account. First, non-ICU female and male inpatients are not allowed to be allocated to the same room. Second, certain medical conditions require patients to be in rooms which are equipped with the necessary infrastructure, e.g., telemetry for certain cardiology patients. Third, it may be the case that a patient or several patients need to be isolated from other patients during their stay due to medical reasons. Finally, non-medically induced room transfers are not allowed, meaning that allocations of patients who already physically occupy rooms in their designated department are treated as unchangeable. This is because every physical room transfer entails significant additional work for hospital personnel (e.g., cleaning and sanitizing rooms, moving beds, reorganizing tasks) as well as unnecessary discomfort for the patient. In this context, the only exceptions are transfers due to medical reasons (e.g., transfers to and from the ICU, which may be modeled as separate patient arrivals and discharges).

4.2.2 Related literature and open research questions

Related Literature Scheduling elective inpatients for surgery or treatment such that utilization of bed capacity is optimized has been thoroughly investigated in the literature. For example, Beliën and Demeulemeester (2007) optimize bed capacity utilization by incorporating the LOS of surgical patients into master surgery schedules in order to balance bed capacity utilization over time. A similar approach has been developed by Fügener et al. (2014) who investigate the effects of scheduling surgery patients on several downstream resources such as the ICU or general ward capacities. Gartner and Kolisch (2014) further investigate scheduling procedures for elective patients such that the contribution margin per patient as well as the utilization of hospital resources such as beds are optimized.

The PBA problem has been introduced by Demeester et al. (2010). Note, that Demeester et al. (2010) define the PBA problem as “patient admission scheduling problem”. However, they consider and solve the PBA problem as defined in Section 4.2.1. Demeester et al. (2010) suggest a decision support system that assigns incoming patients to beds. They consider a situation in which a hospital is initially empty and all future patient arrivals within a given time horizon are known as well as their respective parameters, i.e., actual LOS, gender, department adherence, individual infrastructural needs and so forth. In their model, every patient has to be assigned to a room such that an overall cost function based on violating patient-specific requirements and objectives is minimized. The formulated cost function acknowledges gender-specific room allocation, assignment of patients to departments suited for their age, availability of relevant infrastructure, adherence to medical isolation, patient-specific room type preferences (e.g., single or double room) and patient transfers. Based on this cost function patients are assigned to available rooms of a certain type while taking predefined admission and discharge dates of each patient into account. Demeester et al. (2010) neglect nurse- and doctor-specific objectives and do not distinguish between emergency and elective patients. In addition, they assume a static offline planning situation in which all given patients are assigned to the

available rooms. An overflow buffer is not considered. Therefore, it has to be ensured in advance that a given data set allows for a feasible assignment of all patients to the limited number of rooms. Demeester et al. (2010) solve the assignment problem using a tabu search algorithm.
Several authors have contributed to the problem of operational bed allocation either by providing alternative and/or improved heuristic solution approaches for the problem defined by Demeester et al. and/or by adding certain aspects to the problem.

Ceschia and Schaerf (2011) build on the model, solution approach, and data sets provided by Demeester et al. (2010) by introducing new neighborhood search strategies. They further propose a relaxation procedure to provide lower bounds and introduce a simple dynamic version of the planning problem. Subsequently, the authors expanded on their work and introduced a more sophisticated heuristic solution approach involving simulated annealing, incorporated emergency patient arrivals (see Ceschia and Schaerf (2012)), and most recently included operating room utilization (see Ceschia and Schaerf (2016)). Additionally, Ceschia and Schaerf (2016) allow admission delays while penalizing delays that happen close to the originally planned admission date but do not consider overflow per se.

Bilgin et al. (2012) build on the work of Demeester et al. (2010) by investigating a hyper-heuristic approach to the PBA problem which focuses on optimizing the trade-off between run-time and solution quality. A different solution approach similarly aimed at finding a faster solution approach was proposed by Range et al. (2014) who use a column generation approach for solving the PBA problem. Vancroonenburg et al. (2016) propose to divide the PBA problem into two IP models which assign current patients to beds and reserve beds for future patient arrivals, respectively. A further approach to solving the PBA problem worth noting was presented by Schmidt et al. (2013), which in contrast to the afore-mentioned approaches, focuses on assigning patients to bed contingents rather than individual beds, i.e., they neglect room- and bed-specific characteristics, while respecting a given set of patient preferences, respectively.

Open research and contribution to the literature In our joint project with a large German hospital we identified a variety of additional aspects which to the best of our knowledge have not been dealt with in the literature currently available regarding the PBA problem. We therefore suggest a more comprehensive decision support model and a specialized solution approach that overcomes actual planning shortages. The new modeling and solution approach respects diverse interests of patients, nurses, doctors and hospital management while simultaneously considering several hard constraints when assigning patients to rooms and beds, i.e., medical, gender as well as capacity constraints. In addition, we explicitly distinguish between emergency and elective patients and consider their specific needs and requirements. Furthermore, we deal with ad-hoc overflow situations in which it is not possible to simply reschedule or defer patients. We assume a dynamic online planning situation in which the PBA problem needs to be solved several hundred times a day, i.e., at each point in time an inpatient gets admitted or discharged or when any other change in the system merits moving patients from an overflow area to a regular bed. In addition, the developed approach is based on real time data that also anticipates future developments, such that the decision support system can provide reliable online recommendations for patient-bed allocations. Last but not least we prove the general applicability of the approach suggested in hospital practice using data sets from the literature as well as actual clinic data from our case study hospital.

4.3 Modeling and solution approach

In the present section we develop a decision support model and a greedy look-ahead heuristic (GLA heuristic) to assign elective and emergency inpatients to beds. The model and the solution approach is designed to be solved every time a change in the underlying parameters of the system may lead to the physical allocation of a newly arrived patient or a patient waiting in an overflow area to a regular bed. This may lead to several hundred reruns of the designed procedure per day.

4.3.1 Model development

The deterministic model maximizes a utility function which quantifies the trade-off between patient-specific, doctor-specific, as well as nurse-specific objectives, while simultaneously considering medical, gender as well as capacity constraints when assigning patients to rooms and beds. In addition, the model allows to assign patients to an overflow area if regular beds are not available during the first or — as the case may be — for up to all days of their designated stay. Table 4.1 summarizes the sets, parameters and variables used when formulating the model.

Table 4.1: Notation

Sets	
B	set of beds which are scheduled to be vacated within the planning horizon of $\|T\|$ days, $B = \{1, 2, ..., b, ..., \|B\|\}$
D	set of departments, $D = \{1, 2, ..., d, ..., \|D\|\}$
P	set of patients who require a bed at some point in time within the planning horizon of $\|T\|$ days including patients already waiting in the overflow area, $P = \{1, 2, ..., p, ..., \|P\|\}$
R	set of rooms which have at least one available bed, $R = \{1, 2, ..., r, ..., \|R\|\}$
T	set of days within the planning horizon, $T = \{1, 2, ..., t, ..., \|T\|\}$
W	set of wards which have at least one available bed, $W = \{1, 2, ..., w, ..., \|W\|\}$
Parameters	
$\alpha, \beta, \gamma, \delta$	weighting factors for patient- (α and β), doctor- and nurse-related utilities, respectively
Ξ_p	weighting factor that allows to distinguish between patient types, e.g., elective and emergency patients
A_p	age of patient p

Continued on next page

Table 4.1 – *Continued from previous page*

$\mathrm{A}_{rt}^{\max}$ ($\mathrm{A}_{rt}^{\min}$)	$\mathrm{A}_{rt}^{\max}$ ($\mathrm{A}_{rt}^{\min}$) is set to the maximum (minimum) age of all patients already physically occupying room r for the night on day t and to 0 (M) if the room is empty
OV_p	utility parameter depending on the time patient p has already spent in the overflow area due to a previous overflow situation
c_{wt}	additional care capacity for scheduling additional patients $p \in P$ on ward w on day t
C_p	care level required to accommodate patient p
d_{rt}	d_{rt} represents the department of the prior occupants of room r on day t only in case all of them are allocated to the same department and 0 otherwise
D_p	associated department of patient p with $\mathrm{D}_p \in D$
e_{bt}	$\mathrm{e}_{bt} = 1$ if bed b is located in a room that is initially empty on day t and 0 otherwise
f_{rt}	$\mathrm{f}_{rt} = 1$ if room r is initially empty on day t and 0 otherwise
G_p	$\mathrm{G}_p = -1$ if patient p is male and $\mathrm{G}_p = 1$ if patient p is female
I_p	$\mathrm{I}_p = -1$ if patient p requires medical isolation and 1 otherwise
K_{br}	$\mathrm{K}_{br} = 1$ if bed b is in room r and 0 otherwise
L_{bw}	$\mathrm{L}_{bw} = 1$ if bed b is in ward w and 0 otherwise
M	large integer value
π_{bp}	utility of assigning patient p to bed b based on overflow and patient type (basic model)
Q_t	relevance of a bed allocation for a patient on day t as anticipated/planned; Q_t approximated as $\mathrm{Q}_t = (1 - \mathrm{q})^t$ with discounting parameter $\mathrm{q} \in]0;1[$
s_{bpt}	$\mathrm{s}_{bpt} = 1$ in case bed b is available for patient p on day t of his stay in the hospital and 0 otherwise (“availability” further considers gender, infrastructural, and medical isolation constraints based on pre-occupancies in the room of bed b)
Decision variable	
x_{bp}	$x_{bp} = 1$ if patient p is assigned to bed b and 0 otherwise
Auxiliary variables	
$a_{rt}^{\max}$ ($a_{rt}^{\min}$)	$a_{rt}^{\max}$ ($a_{rt}^{\min}$) is the maximum (minimum) age of all patients p assigned to room r on day t

Continued on next page

Table 4.1 – *Continued from previous page*

o^+_{wt}	o^+_{wt} denotes the additional accumulated care level surpassing a predefined threshold for a given ward w on day t
y_{rt}	$y_{rt} = 1$ if all patients assigned to an empty room r on day t are from the same department and 0 otherwise
z_{rt}	$z_{rt} = 1$ if all patients assigned to a partially occupied room r are from the same department as the patients already occupying room r and 0 otherwise

We formulate the objective function as a multi-objective utility maximization function to accommodate the trade-offs between the diverse interests of patients, nurses and doctors that exist when allocating patients to beds. The objective function (4.1) is formulated as follows:

$$\max\ \Pi = \alpha\, f_{\text{basic}}(x_{bp}) - \beta\, f_{\text{patient}}(x_{bp}) + \gamma\, f_{\text{doctor}}(x_{bp}) - \delta\, f_{\text{nurse}}(x_{bp}) \tag{4.1}$$

Equation (4.1) consists of four terms that represent (I) basic patient-specific objectives, (II) extended patient-specific objectives, (III) doctor-specific objectives and finally (IV) nurse-specific objectives. In the following, we will gradually develop the four parts. The four partial objectives are weighted by the factors α, β, γ, and δ. These weighting factors are used to control the influence of the individual objectives on the overall solution. They are derived from managerial decisions. All four objective values depend on the assignment variable x_{bp} which equals to 1 if patient p is allocated to bed b and 0 otherwise.

(I) Basic patient-specific objectives and constraints The first term quantifies the patient-type-specific objectives:

$$f_{\text{basic}}(x_{bp}) = \sum_{b \in B} \sum_{p \in P} \pi_{bp} x_{bp} \tag{4.2}$$

Parameter π_{bp} denotes the patient-type-specific “utility” of assigning patient p to bed b. It depends solely on information known prior to updating the bed allocation planning. Thus, parameter π_{bp} is not influenced by other assignments of patients $p \in P$ to beds $b \in B$ during a specific planning instant. As room transfers are not allowed, every assignment of a patient p to a bed b, i.e., $x_{bp} = 1$ generates a utility of π_{bp}, which accounts for the days that patient p actually spends in bed b within the planning horizon T. For a given patient p and a given bed b the utility value is quantified as follows:

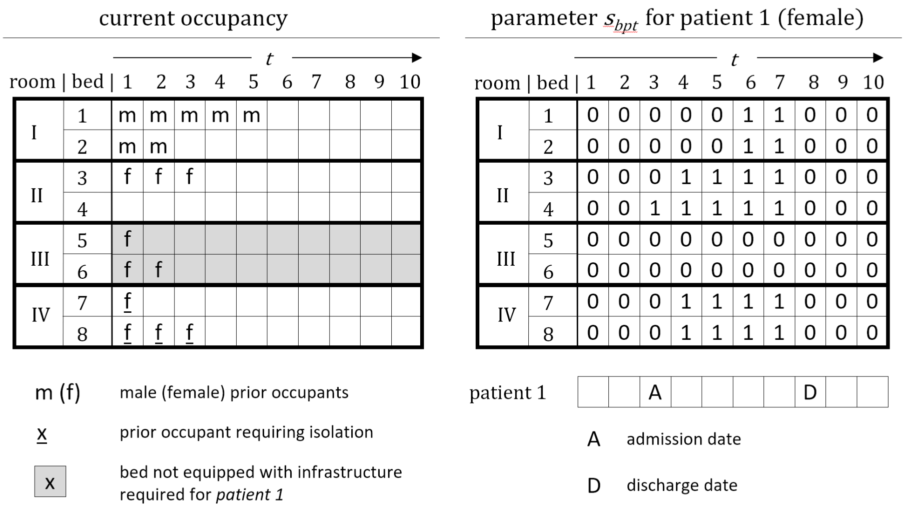
$$\pi_{bp} = \mathrm{OV}_p + \Xi_p \sum_{t \in T} \mathrm{s}_{bpt} \mathrm{Q}_t \tag{4.3}$$

Here, OV_p represents a predetermined utility value which depends on the time a patient p has already spent in the overflow area in the past. This “overflow bonus” is only awarded to patients who are already waiting in the overflow area at the time the decision model is solved. This is done to ensure that patients who are already in the overflow area do not risk staying there for the entirety of their stay. In other words, the set of patients P includes not only current and future planned and anticipated emergency patient arrivals but also patients that are currently waiting in the overflow area. Patients already waiting in the overflow area will be preferred to otherwise similar patients who have just arrived in the hospital as a result of the additional utility value OV_p.

The second part of Equation (4.3) rewards the actual time that a patient p spends in one of the beds $b \in B$. To this end, the predetermined parameter s_{bpt} is introduced, which is preset to 1 in case bed b is available for patient p on day t and 0 otherwise. As non-medical room transfers are not allowed, s_{bpt} can be determined entirely during preprocessing and is used to reflect not only bed availability but also bed compatibility by incorporating gender constraints, infrastructural constraints, as well as medical isolation constraints for each possible patient-bed combination. The advantage of summing up s_{bpt} over $t \in T$ can be seen in that s_{bpt} may be determined entirely during preprocessing. Figure (4.3) shows an example illustrating how parameter s_{bpt} is set to 0 or 1. The example given considers 4 rooms, i.e., room I with beds 1 and 2, room II with beds 3 and 4, and so forth. Here, there are multiple options for allocating female patient 1 to a bed. This patient arrives at the hospital on day 3 and is scheduled to be discharged on day 8, thus having an anticipated LOS of 5 days. An allocation to bed 1, for example, would imply an initial stay of three days within the overflow area before moving to bed 1 for the remaining two days. Accordingly, an allocation to bed 1 would have a lower utility than an allocation to bed 4, for example, as the first three days spent in the overflow area do not create any additional benefit. Beds 5 and 6, for example, are not allowed to be used as this room is not equipped with essential medical infrastructure specifically required for treating patient 1. Beds 7 and 8, however, are both currently occupied by female patients requiring medical isolation from non-quarantined patients for the duration of their stay, hence forcing patient 1 to spend one day within the overflow area before moving into either of these beds, should she be allocated to one of them.

It is important to note, that s_{bpt} does not define the LOS of patient p. This is because treatment of patients (e.g., surgery, medication) typically starts once the patient arrives at the hospital, regardless of where in the hospital their bed is physically located.

In addition, x_{bp} defines not only the bed b that patient p is allocated to, but also the time patient p has to spend in the overflow area depending



current occupancy

room	bed	1	2	3	4	5	6	7	8	9	10
I	1	m	m	m	m	m					
	2	m	m								
II	3	f	f	f							
	4										
III	5	f									
	6	f	f								
IV	7	$\underline{f}$									
	8	$\underline{f}$	$\underline{f}$	$\underline{f}$							

parameter s_{bpt} for patient 1 (female)

room	bed	1	2	3	4	5	6	7	8	9	10
I	1	0	0	0	0	0	1	1	0	0	0
	2	0	0	0	0	0	1	1	0	0	0
II	3	0	0	0	1	1	1	1	0	0	0
	4	0	0	1	1	1	1	1	0	0	0
III	5	0	0	0	0	0	0	0	0	0	0
	6	0	0	0	0	0	0	0	0	0	0
IV	7	0	0	0	1	1	1	1	0	0	0
	8	0	0	0	1	1	1	1	0	0	0

patient 1			A					D		

t
t
m (f) male (female) prior occupants
$\underline{x}$ prior occupant requiring isolation
x bed not equipped with infrastructure required for *patient 1*
A admission date
D discharge date

Figure 4.3: Example for quantifying s_{bpt}

on the current occupancy situation at the time the planning is updated. In addition, spending time in the overflow area does not affect the overall LOS. The parameter Ξ_p is a factor that allows to distinguish between patient types, i.e., elective patients, emergency patients, or patients with special infrastructural requirements. This factor may, for example, be used to ensure that elective patients are more likely to be assigned to a bed within their target ward upon arrival than emergency patients, or to ensure that patients with special infrastructural needs are preferred. For example, patients returning from the ICU could be attributed an even higher value such that it is highly unlikely for them to be moved to an overflow area.

Finally, $\sum_{t \in T} s_{bpt} Q_t$ incorporates the time a patient is assigned to a regular bed during his/her LOS. Q_t is a parameter that reflects the relevance of a bed allocation for a patient on day t as anticipated/planned where Q_t is decreasing with increasing t. Thus, otherwise similarly evaluated patients contribute to the overall objective function with a higher utility if they require a bed earlier in the planning horizon considered. This modeling approach anticipates the possibility of reassigning later arriving patients to other beds at planning instants in the future. Due to uncertainties it is quite reasonable that a patient, who is planned to arrive far in the future, will be reassigned to another bed at later planning periods, which may then even lead to a higher overall utility value for that patient. Possible uncertainties are related to LOS, emergency arrivals, treatment progression, no-shows and so forth. The decreasing parameter Q_t is approximated as follows, assuming $\mathrm{q} \in]0;1[$:

$$\mathrm{Q}_t = (1-\mathrm{q})^t \tag{4.4}$$

In the following, the basic set of hard constraints is listed which have to be adhered to regardless of how the individual parts of the objective function are actually weighted.

$$\sum_{b \in B} x_{bp} \leq 1 \qquad \forall p \in P \tag{4.5}$$

$$\sum_{p \in P} \mathrm{s}_{bpt} x_{bp} \leq 1 \qquad \forall b \in B; t \in T \tag{4.6}$$

$$\sum_{t \in T} \mathrm{s}_{bpt} \geq x_{bp} \qquad \forall b \in B; p \in P \tag{4.7}$$

$$\mathrm{K}_{br}\mathrm{e}_{bt}\mathrm{G}_p\mathrm{s}_{bpt}x_{bp} - \mathrm{K}_{lr}\mathrm{e}_{lt}\mathrm{G}_h\mathrm{s}_{lht}x_{lh} \geq -1 \quad \forall b,l \in B; p,h \in P; r \in R; t \in T \tag{4.8}$$

$$\mathrm{K}_{br}\mathrm{e}_{bt}\mathrm{I}_p\mathrm{s}_{bpt}x_{bp} - \mathrm{K}_{lr}\mathrm{e}_{lt}\mathrm{I}_h\mathrm{s}_{lht}x_{lh} \geq -1 \qquad \forall b,l \in B; p,h \in P; r \in R; t \in T \tag{4.9}$$

$$x_{bp} \in \{0,1\} \qquad \forall b \in B; p \in P \tag{4.10}$$

Equation (4.5) prevents double-booking by ensuring that each patient is assigned to no more than one bed. Please note, a patient receives no bed assignment if he/she entirely stays in the overflow area during his/her scheduled LOS. In addition, Equation (4.6) prevents overbooking, such that no two patients are allocated to the same bed on the same day. Equation (4.7) ensures that a patient p can only be assigned to a bed b, i.e., $x_{bp} = 1$ if bed b is at least available for this patient on one day of the planning horizon, i.e., $s_{bpt} = 1$ for at least one $t \in T$.

Furthermore, Equation (4.8) in combination with s_{bpt} ensures that there are no mixed male and female rooms on any given day t. Here, G_p is set to -1 if patient p is male and to 1 if patient p is female. In particular, Equation 4.8 compares all patients $p \in P$ which may be allocated to room r on day t. In case a male patient is mixed with a female patient, the equation would not be satisfied as it would then read $-1 - 1 \geq -1$, i.e., $-2 \geq -1$. In addition, most rooms are already preoccupied on specific days, such that only male or female patients are additionally allowed, respectively. As pointed out above, this prior occupancy is integrated into s_{btp}. Prior occupancies are reflected in s_{bpt} such that $\mathrm{s}_{bpt} = 0$ for a female patient p in case a bed b is located in a room which is still occupied by at least one male patient on day t and vice versa (see Figure (4.3) for an example). The parameter e_{bt} is set to 1 if bed b is located in a room that does not have any current occupants (i.e., which is empty at the time the planning is updated) on day

t and 0 otherwise. Finally, K_{br} connects beds to rooms and is set to 1 if bed b is located in room r and 0 otherwise.

Using a similar approach, equation (4.9) in combination with s_{bpt} ensures that medical isolation requirements are respected. Specifically, patients that need to be isolated due to infectious diseases, for example, may only be put into empty rooms or into rooms with patients that suffer from the same condition. Here, I_p is set to -1 if patient p requires medical isolation and 1 otherwise.

(II) Further patient-specific objectives and constraints The second term of the objective function (4.1) is used to model the preferences of the patients. This part of the objective function tries to minimize the age differences within rooms since it is desirable to combine patients of a similar age as it is more likely for them to share common interests. In addition, they potentially share similar illnesses when associated to the same medical department. Numerous interviews at our case hospital verify this approach. The second term of the objective function is then denoted as follows:

$$f_{\text{patient}}(x_{bp}) = \sum_{r \in R} \sum_{t \in T} (a_{rt}^{\max} - a_{rt}^{\min}) \tag{4.11}$$

Here, $a_{rt}^{\max}$ ($a_{rt}^{\min}$) denotes the maximum (minimum) age of all patients which during run-time of the model are going to be assigned to room r on day t. As such, both auxiliary variables $a_{rt}^{\max}$ and $a_{rt}^{\min}$ are dependent on the overall decision variable x_{bp}. The following constraints are used to link these auxiliary variables to x_{bp}:

$$a_{rt}^{\max} \geq \mathrm{A}_{rt}^{\max} \qquad \forall r \in R; t \in T \tag{4.12}$$

$$a_{rt}^{\max} \geq \mathrm{K}_{br}\mathrm{A}_{p}\mathrm{s}_{bpt}x_{bp} \qquad \forall b \in B; p \in P; r \in R; t \in T \quad (4.13)$$

$$a_{rt}^{\min} \leq \mathrm{A}_{rt}^{\min} \qquad \forall r \in R; t \in T \quad (4.14)$$

$$a_{rt}^{\min} \leq \sum_{b\in B}\sum_{p\in P} \mathrm{A}_{rt}^{\min}\mathrm{K}_{br}\mathrm{s}_{bpt}x_{bp} \qquad \forall r \in R; t \in T \quad (4.15)$$

$$a_{rt}^{\min} \leq \mathrm{K}_{br}\mathrm{A}_{p}\mathrm{s}_{bpt}x_{bp} + \mathrm{A}_{rt}^{\min}(1 - x_{bp}) \quad \forall b \in B; p \in P; r \in R; t \in T \quad (4.16)$$

Here, $\mathrm{A}_{rt}^{\max}$ is set to the current maximum age of all patients already occupying room r on day t and to 0 in case room r is empty on day t. As it is solely dependent on prior occupancy, $\mathrm{A}_{rt}^{\max}$ is determined entirely during preprocessing and is not affected by x_{bp}.

With the same logic, $\mathrm{A}_{rt}^{\min}$ is set to the minimum age of all patients already occupying room r on day t and to a large integer value, e.g., 150, the maximum age of any possible patient, in case there are no prior occupants in room r on day t. Thus, Equations (4.12) and (4.13) ensure that the auxiliary variable $a_{rt}^{\max}$ reflects the maximum age of prior occupants and newly allocated patients in a room r on day t. Likewise, Equations (4.14) to (4.16) ensure the same for $a_{rt}^{\min}$ while also making sure that $a_{rt}^{\min}$ equals $a_{rt}^{\max}$ in the case room r is only occupied by one person or completely empty on day t.

(III) Doctor-specific objectives and constraints The third term of the objective function (4.1) rewards assigning patients of the same department to identical rooms. Medical rounds for doctors are easier when having several patients they are responsible for in the same room. In addition, walking distances are reduced. The third term of the objective function is then formulated as follows:

$$f_{\text{doctor}}(x_{bp}) = \sum_{r\in R}\sum_{t\in T} \mathrm{f}_{rt}y_{rt} + \sum_{r\in R}\sum_{t\in T}(1 - \mathrm{f}_{rt})z_{rt} \quad (4.17)$$

As before, an additional set of constraints is required to establish the link between the decision variable x_{bp} and Equation (4.17):

$$\mathrm{K}_{br}\mathrm{D}_p\mathrm{s}_{bpt}x_{bp} - \mathrm{K}_{lr}\mathrm{D}_h\mathrm{s}_{lht}x_{lh} \geq -M(1 - y_{rt}) \qquad \forall b, l \in B; p, h \in P; r \in R; t \in T \tag{4.18}$$

$$\sum_{p\in P}\sum_{b\in B}\mathrm{K}_{br}\mathrm{s}_{bpt}x_{bp} \geq y_{rt} \qquad \forall r \in R; t \in T \tag{4.19}$$

$$\mathrm{K}_{br}\mathrm{D}_p\mathrm{s}_{bpt}x_{bp} - \mathrm{d}_{rt} \leq M(1 - z_{rt}) \qquad \forall b \in B; p \in P; r \in R; t \in T \tag{4.20}$$

$$\mathrm{d}_{rt} - \mathrm{K}_{br}\mathrm{D}_p\mathrm{s}_{bpt}x_{bp} \leq M(1 - z_{rt}) \qquad \forall b \in B; p \in P; r \in R; t \in T \tag{4.21}$$

$$\sum_{p\in P}\sum_{b\in B}\mathrm{K}_{br}\mathrm{s}_{bpt}x_{bp} \geq z_{rt} \qquad \forall r \in R; t \in T \tag{4.22}$$

$$y_{rt} \in \{0, 1\} \qquad \forall r \in R; t \in T \tag{4.23}$$

$$z_{rt} \in \{0, 1\} \qquad \forall r \in R; t \in T \tag{4.24}$$

Here, D_p is an integer parameter that depicts the department that corresponds to the medical condition of patient p. In addition, d_{rt} depicts the department of all prior occupants of room r on day t only if all prior occupants are from the same department and is set to 0 otherwise. As such, both D_p and d_{rt} are determined entirely during preprocessing. The parameter f_{rt} is 1 in case room r does not have any prior occupants on day t and 0 otherwise. Accordingly, the auxiliary variable y_{rt} is set to 1, if all patients assigned to an empty room r on day t are from the same department which is achieved by Equations (4.18) and (4.19). An additional auxiliary variable z_{rt} is used in case a room r is already preoccupied on day t and is set to 1 only if all patients assigned to room r as well as the patients in room r are already from the same department. This is achieved with Equations (4.20) to (4.22).

(IV) Nurse-specific objectives and constraints Finally, the fourth term of the objective function (4.1) is used to balance workload for nursing staff (see Section 4.2 for details) and is quantified as follows:

$$f_{\text{nurse}}(x_{bp}) = \sum_{w \in W} \sum_{t \in T} o^{+}_{wt} \tag{4.25}$$

In particular, exceeding a predefined care capacity for nursing staff assigned to ward w on day t is penalized. To this end, the following additional set of constraints is required:

$$\sum_{b \in B} \sum_{p \in P} \mathrm{L}_{bw} \mathrm{C}_p \mathrm{s}_{bpt} x_{bp} \leq \mathrm{c}_{wt} + o^{+}_{wt} \qquad \forall t \in T; w \in W \tag{4.26}$$

$$o^{+}_{wt} \geq 0 \qquad \forall t \in T; w \in W \tag{4.27}$$

Parameter C_p quantifies the level of care required for patient p. This represents the effort and resources that go into taking care of a particular patient. In addition, the available number of nursing staff and thus, workforce per ward w and day t is predetermined due to shift schedules, staff rosters, and so forth. Thus, the parameter c_{wt} represents the additional care capacity of a given ward w on day t, i.e., the capacity to take in additional patients $p \in P$ requiring C_p units of care, respectively. For instance, assume $c_{wt} = 6$ for a given ward w and a given day t. This would then mean that ward w could additionally take up 2 patients with a care level $C_p = 3$ before overloading the nursing staff of that ward on that day, for example. Nursing staff typically cannot be moved from ward to ward on an ad-hoc basis. This means that having patients in a first ward that are very easy to handle cannot balance out a second ward filled with a very labor-intensive patient clientele. Thus, the auxiliary variable o^+_{wt} is introduced which denotes the additional accumulated care level surpassing the predefined care capacity threshold for a given ward w on day t. Equations (4.26) and (4.27) are used to link x_{bp} to o^+_{wt}.

4.3.2 Greedy look-ahead heuristic

An efficient bed allocation support system needs to be able to give a bed planner online recommendations for patient bed allocations within seconds when prompted. This is due to real-life planning situations in large hospitals requiring highly flexible planning systems which are able to adapt to ad-hoc changes in real time. However, solving the model by Gurobi's MIP solver requires more than 12 hours for relevant problem instances (see Section 4.4 for details). Likewise, other approaches followed in the literature (see for example Demeester et al. (2010); Ceschia and Schaerf (2011)) also had to resort to using heuristic approaches for the same reasons. We therefore develop a novel greedy look-ahead heuristic (GLA heuristic) which bases on the general idea of Atkinson (1994) by sequentially assigning the most utility-attractive patient to his or her most beneficial bed while anticipating potential room allocations still to be made in futher steps of the algorithm. Table 4.2 summarizes the additional notation required to formulate the GLA heuristic.

Table 4.2: Expanded notation for the GLA heuristic

U_{bp}	partial utility that an allocation of patient p to bed b may add to the overall utility Π, $p \in P$, $b \in B$
U_p^{argmax}	index value of the bed that adds the maximum partial utility to the overall utility Π when patient p, $p \in P$ will be allocated to this bed
U_p^{max}	maximum partial utility that an allocation of patient p may add to the overall utility Π, $p \in P$

The basic premise of the GLA heuristic is based on a greedy algorithm when assigning patients to beds which approximates assignments of patients to beds that may be realized in later stages of the algorithm. To this end, a utility matrix U_{bp} is used which, upon initiation of the PBA-algorithm, is prefilled with the partial utilities that a respective allocation of patient p to bed b would add to the overall utility Π of the objective function (4.1). It should again be noted in this context, that the set of patients P as well as the set of beds B includes not just current but also future patient arrivals and bed availabilities, respectively, and as such every value of U_{bp} implicitly includes time already spent in and time to be spent in the overflow area as well as uncertainty regarding future arrival and discharge dates. Should a specific bed b not be available at all for patient p at any time of their planned stay, the value U_{bp} is set to zero.

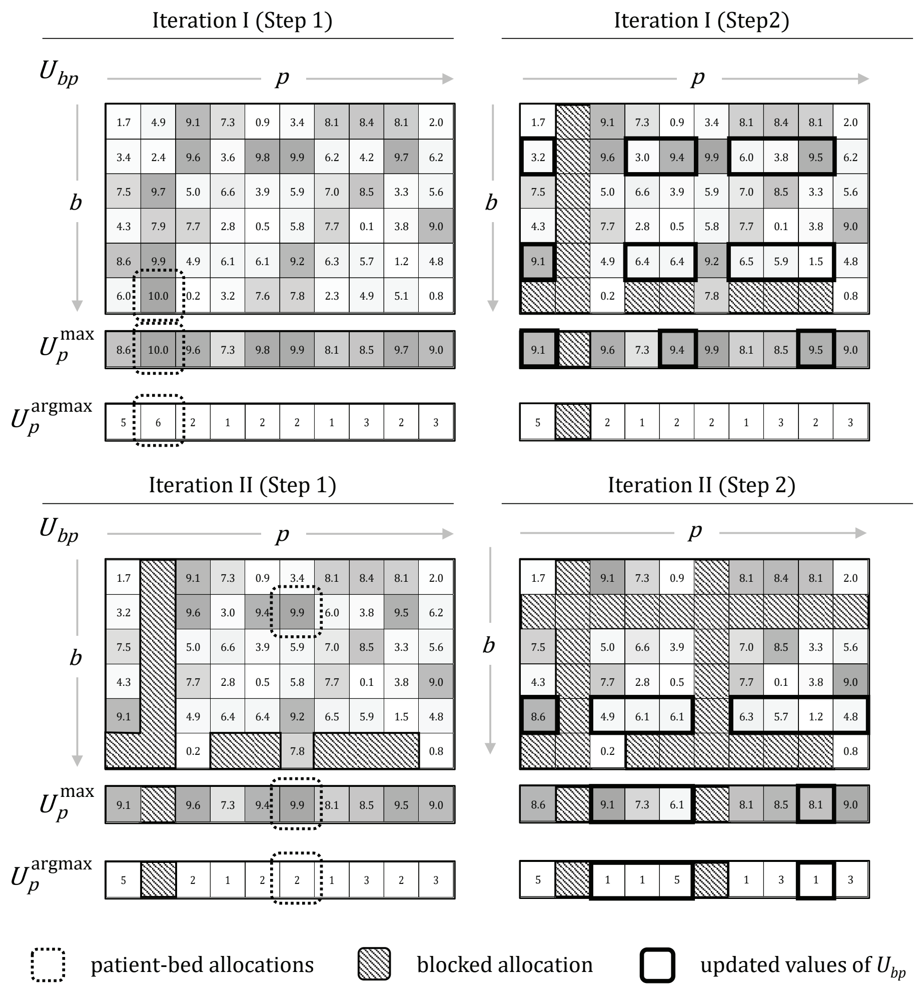
Upon initiation of the GLA heuristic, x_{bp} is set to 0 for all $b \in B$ and $p \in P$. As described above, the initial values for U_{bp} are calculated for every $b \in B$ and $p \in P$. Subsequently, the highest value in U_{bp} is identified and x_{bp} is set to 1 correspondingly, i.e., patient p is allocated to bed b. Finally, all elements in U_{bp} that are affected by any allocation are updated before the next patient is allocated. To streamline the computations, only the vectors U_p^{max} and U_p^{argmax} are calculated. U_p^{max} contains the maximum partial utility that an allocation of patient p may add to the overall utility Π and U_p^{argmax} reveals the index value of the corresponding bed b. If necessary the values U_p^{max} and U_p^{argmax} are also updated after every allocation. This way, the PBA-algorithm only has to compare $|P|$ values instead of $|P| \times |B|$ values.

Figure 4.4 illustrates the first steps of the GLA heuristic. Step 1 of Iteration I shows the initial utility matrix U_{bp} as well as the initial corresponding values for U_p^{max} and U_p^{argmax}. The highest value of U_{bp} then determines the first allocation, i.e., x_{62} is set to 1. This initial allocation of patient $p = 2$ to bed $b = 6$ then has an effect on a series of potential allocation combinations x_{bp} of the remaining patients P and beds B. Therefore, in Step 2 of Iteration I the utility matrix U_{bp} is updated and if necessary the variables U_p^{max} and U_p^{argmax} are redetermined. In the example shown in Figure 4.4, the values marked with black boxes were updated. Iteration II is then substantially equivalent and subsequent to Iteration I. Algorithm 4.1 summarizes the sequential, procedural program flow.

Algorithm 4.1 GLA heuristic

```
Require: $P$, $B$
Ensure: patient-bed allocations $x_{bp}$
$U_{bp} \leftarrow$ calculatePatientBedMatrix($P$, $B$)
$U_p^{\text{max}} \leftarrow \max(U_{bp})$
$U_p^{\text{argmax}} \leftarrow \text{argmax}(U_{bp})$
while ($\max(U_{bp}) \neq 0$) do
    $p \leftarrow \text{argmax}(U_p^{\text{max}})$
    $b \leftarrow U_p^{\text{max}}[p]$
    $x_{bp} \leftarrow 1$
    $U_{bp} \leftarrow$ updatePatientBedMatrix($p$, $b$, $U_{bp}$, $P$, $B$)
end while
printPatientBedAllocations($x_{bp}$)
```

Allocating patients to rooms that are “still empty” at the time of allocation but will be filled during later iterations, i.e., during runtime of the algorithm, is approximated as follows. The value B_{bp} for the case that patient p is allocated to bed b in a previously unoccupied room (at this exact point in the GLA heuristic run through) is calculated assuming that any potential



Iteration I (Step 1)

U_{bp}	1.7	4.9	9.1	7.3	0.9	3.4	8.1	8.4	8.1	2.0
	3.4	2.4	9.6	3.6	9.8	9.9	6.2	4.2	9.7	6.2
	7.5	9.7	5.0	6.6	3.9	5.9	7.0	8.5	3.3	5.6
	4.3	7.9	7.7	2.8	0.5	5.8	7.7	0.1	3.8	9.0
	8.6	9.9	4.9	6.1	6.1	9.2	6.3	5.7	1.2	4.8
	6.0	10.0	0.2	3.2	7.6	7.8	2.3	4.9	5.1	0.8
U_p^{max}	8.6	10.0	9.6	7.3	9.8	9.9	8.1	8.5	9.7	9.0
U_p^{argmax}	5	6	2	1	2	2	1	3	2	3

Iteration I (Step2)

U_{bp}	1.7	▨	9.1	7.3	0.9	3.4	8.1	8.4	8.1	2.0
	3.2	▨	9.6	3.0	9.4	9.9	6.0	3.8	9.5	6.2
	7.5	▨	5.0	6.6	3.9	5.9	7.0	8.5	3.3	5.6
	4.3	▨	7.7	2.8	0.5	5.8	7.7	0.1	3.8	9.0
	9.1	▨	4.9	6.4	6.4	9.2	6.5	5.9	1.5	4.8
	▨	▨	0.2	▨	▨	7.8	▨	▨	▨	0.8
U_p^{max}	9.1	▨	9.6	7.3	9.4	9.9	8.1	8.5	9.5	9.0
U_p^{argmax}	5	▨	2	1	2	2	1	3	2	3

Iteration II (Step 1)

U_{bp}	1.7	▨	9.1	7.3	0.9	3.4	8.1	8.4	8.1	2.0
	3.2	▨	9.6	3.0	9.4	9.9	6.0	3.8	9.5	6.2
	7.5	▨	5.0	6.6	3.9	5.9	7.0	8.5	3.3	5.6
	4.3	▨	7.7	2.8	0.5	5.8	7.7	0.1	3.8	9.0
	9.1	▨	4.9	6.4	6.4	9.2	6.5	5.9	1.5	4.8
	▨	▨	0.2	▨	▨	7.8	▨	▨	▨	0.8
U_p^{max}	9.1	▨	9.6	7.3	9.4	9.9	8.1	8.5	9.5	9.0
U_p^{argmax}	5	▨	2	1	2	2	1	3	2	3

Iteration II (Step 2)

U_{bp}	1.7	▨	9.1	7.3	0.9	▨	8.1	8.4	8.1	2.0
	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨
	7.5	▨	5.0	6.6	3.9	▨	7.0	8.5	3.3	5.6
	4.3	▨	7.7	2.8	0.5	▨	7.7	0.1	3.8	9.0
	8.6	▨	4.9	6.1	6.1	▨	6.3	5.7	1.2	4.8
	▨	▨	0.2	▨	▨	▨	▨	▨	▨	0.8
U_p^{max}	8.6	▨	9.1	7.3	6.1	▨	8.1	8.5	8.1	9.0
U_p^{argmax}	5	▨	1	1	5	▨	1	3	1	3

⬚ patient-bed allocations ▨ blocked allocation □ updated values of U_{bp}

Figure 4.4: Example of the GLA heuristic

future room-mates will not have the same department and will have the largest possible age-difference based on the pool of patients that are still to be allocated during the current run-through of the GLA heuristic. The approach, therefore, “looks ahead” or approximates potential assignments of rooms which will happen at a later stage of the run-through of the algorithm. The procedure avoids that patients are disproportionately assigned to so far empty rooms.

4.4 Numerical study

In this section we present detailed results for our proposed approach. First, the choice of parameters generally used for the numerical studies is stated in Section 4.4.1. In Section 4.4.2 we assess the performance of the GLA heuristic by comparing runtime and solution quality of the GLA heuristic with near-to-optimal solutions obtained by solving our model with Gurobi's MIP solver. In Section 4.4.3, we then solve a case study for a large German hospital. In Section 4.4.4 we analyze the general applicability of our approach by employing different sized problem instances from literature. Section 4.4.5 then further investigates our contribution to literature with regard to patient-specific, doctor-specific, and nurse-specific objectives. All computational steps were carried out in Python 3.6.3 and Gurobi 7.5. All computations were run on a work station equipped with 2 Intel Core E5-2620 processors and 64-GB of RAM.

4.4.1 Parameters

The parameters presented in this Section are used for the following numerical tests. In discussions with nurses, doctors, and hospital management we determined the basic parameters to be used for our case study (see Table 4.3).

Table 4.3: Overview of weighting factors used

Parameters	α	β	γ	δ	Ξ^{el}	Ξ^{em}	Ξ^{an}	Q_t
Values	1	0.01	0.2	0.2	20	19	3	$Q_t = (1-q)^t; q = 0.01$

The main goal was to ensure that elective patients are generally preferred over emergency arrivals to prevent allocating them to the overflow area (see Section 4.2 for details). To achieve this, the weighting factor Ξ_p was set to three distinct values depending on the patient type. Notably, these consist of Ξ^{el} for elective patients, Ξ^{em} for current emergency arrivals, and

Ξ^{an} for anticipated emergency arrivals. Here, current emergency arrivals are preferred over anticipated future emergency arrivals. This is due to the fact, that the parameters of recently arrived emergency patients requiring a bed are well known, whereas the relevant parameters of future emergency arrivals have to be anticipated based on historical probability distributions. Finally, the parameters α, β, γ, δ, and Q_t were set such that patient-specific, doctor-specific, and nurse-specific objectives reflect managerial decisions regarding PBA in our case hospital. In our case study, two specific effects regarding uncertainty of future events stood out. First, the no-show probability was higher, the farther in the future a patient arrival was scheduled. This is to be expected, as the time for potential problems or issues to arise is longer. Second, doctors responsible for giving LOS estimates based on their patients' medical conditions tended to be more conservative with these estimates the longer the remaining LOS was. This was mainly due to doctors wanting to "avoid false promises" to patients and the admission scheduling office alike. We approximate these issues by a geometric function $Q_t = (1 - q)^t$ wherein q represents the associated discounting factor. In essence, patients requiring a bed earlier within the planning horizon obtain a higher priority than those who require a bed at a later stage.

4.4.2 Performance of the GLA heuristic

To assess the overall solution quality of our GLA heuristic we created different sized problem instances for testing purposes ranging from 24 to 120 beds while using a planning horizon between 1 and 9 days. For each of these problem sizes, we created 20 unique data sets based on the original data obtained from our case hospital, i.e., over a year. In particular, 20 different "snap shots in time" were chosen at random from a 9 month period worth of raw data to provide 20 completely different starting situations or problem instances.

We then compared average run times for each problem size by comparing near to optimal solutions of a Gurobi implementation of our model with results obtained by our GLA heuristic (see Table 4.4). Near to optimal means that we allowed a MIP Gap of up to 1%.

Table 4.4: Computational time analyses

GLA heuristic - average solution time in seconds

|B|	|T| = 1	|T| = 3	|T| = 5	|T| = 7	|T| = 9
24	0.005	0.007	0.014	0.014	0.095
48	0.012	0.028	0.060	0.106	0.171
72	0.015	0.051	0.165	0.261	0.445
96	0.022	0.073	0.157	0.296	0.551
120	0.044	0.139	0.246	0.585	0.853

Gurobi solution - average solution time in seconds

|B|	|T| = 1	|T| = 3	|T| = 5	|T| = 7	|T| = 9
24	17	125	740	2124	9360
48	157	3492	19780		
72	1212	16476			
96	2822			stopped after 12 hours	
120	10479				

Table 4.4 gives an overview of the average run times obtained. For the smallest problem size, i.e., |B| = 24, the run time of the Gurobi implementation increases considerably from 17 seconds when only considering a planning horizon of |T| = 1 to over 2.5 hours when using a planning horizon of |T| = 9. A similar increase can be observed when augmenting the amount of beds included in the problem. Hence, run time heavily depends on both planning horizon and the amount of beds considered such that typical problem sizes, e.g., 100 beds and more with a planning horizon of 1 week, cannot be solved within a reasonable time frame. For the purpose of our analyses we stopped the Gurobi solver after 12 hours for each data set. However, when using our GLA heuristic, solution times stayed at under a second even for the largest problem size tested. In addition, the solution times obtained by the heuristic show a significantly lower rate of increase compared to the Gurobi implementation when moving from smaller to

larger problem sizes.

Table 4.5: Overview of average MIP Gap of the Gurobi implementation

Gurobi solution - average MIP Gap					
|B|	|T| = 1	|T| = 3	|T| = 5	|T| = 7	|T| = 9
24	0.46%	0.53%	0.57%	0.72%	0.79%
48	0.80%	0.87%	0.89%	3.59%	6.79%
72	0.84%	0.95%	3.78%	6.82%	7.30%
96	0.96%	6.45%	6.70%	7.54%	8.19%
120	0.98%	6.88%	7.87%	10.60%	15.39%

Table 4.5 shows an overview of the average MIP Gaps obtained with the Gurobi implementation. The near-to-optimal solutions shown above the dashed lines were all able to be solved with a MIP Gap of about 1% or less within less than twelve hours (see Table 4.4). For all other values, the Gurobi MIP solver was stopped after 12 hours and the respective solutions and their corresponding MIP Gap at that time were recorded. Here, it can be seen that solving the model in adequate time with a standard MIP program does not seem feasible for typical problem instances in large hospitals.

Table 4.6 shows the average as well as the minimum and maximum solution quality obtained for each problem size. Solution quality is defined as the comparison between the values of the objective function based on the patient bed allocations created by both approaches. In particular, the GLA heuristic was able to achieve a solution quality of more than 95% for all comparable problem sizes. In addition, two effects can be observed from the data in Table 4.6. First, the average solution quality of the heuristic decreases slightly when increasing the planning horizon. This is to be expected since a longer planning horizon creates more favorable combinatorial combinations of patient bed allocations which are not straightforward and as such will likely not be detected by the heuristic approach. Second, the minimum solution quality and with it the average

Table 4.6: Solution quality of GLA heuristic compared to Gurobi solution

GLA heuristic - average solution quality compared to Gurobi solution[1]

|B|	|T| = 1	|T| = 3	|T| = 5	|T| = 7	|T| = 9
24	98.68%	98.08%	98.95%	98.59%	98.45%
48	98.92%	97.40%	96.83%	96.30%	96.89%
72	98.80%	98.34%	97.53%	98.01%	99.58%
96	99.54%	99.27%	99.40%	99.78%	100.55%
120	99.55%	99.46%	99.13%	101.49%	101.00%

GLA heuristic - maximum solution quality compared to Gurobi solution[1]

|B|	|T| = 1	|T| = 3	|T| = 5	|T| = 7	|T| = 9
24	100.0%	99.98%	99.93%	99.87%	99.65%
48	100.0%	99.70%	99.32%	99.48%	99.67%
72	100.0%	99.80%	99.76%	99.82%	99.58%
96	100.0%	99.85%	99.60%	100.21%	100.55%
120	100.0%	99.67%	99.20%	102.45%	101.00%

GLA heuristic - minimum solution quality compared to Gurobi solution[1]

|B|	|T| = 1	|T| = 3	|T| = 5	|T| = 7	|T| = 9
24	90.71%	91.04%	95.66%	95.27%	96.17%
48	96.52%	91.33%	94.19%	92.40%	94.24%
72	91.74%	96.21%	96.24%	96.04%	99.58%
96	98.38%	98.69%	98.99%	99.36%	100.55%
120	97.71%	99.05%	99.00%	99.57%	101.00%

[1] values below dashed line reflect the solution obtained when stopping the Gurobi solver after 12 hours

solution quality generally increases the more beds are considered. This may be attributed to the higher probability of having comparable solutions in terms of the respective objective function value when increasing the number of beds. In other words, the heuristic will likely find a similarly adequate patient bed allocation even it moves away from the near to optimal solution. This can also be seen when comparing the solutions obtained by the Gurobi solver when stopped after 12 hours, i.e., the values below the dashed line. Here, the heuristic even outperforms the solution obtained by Gurobi's MIP solver for large problem instances while using only a fraction of the time. In summary, these analyses indicate that the use of the GLA heuristic developed may indeed deliver high solution quality results even for very large problem instances while at the same time providing ad-hoc online recommendations within seconds.

4.4.3 Case study

The modeling and solution approach suggested is applied at a large hospital in Germany. The first paragraph presents the data and parameters used followed by the second paragraph which presents the main results for our case study.

Environment, data used and methodology For our case study we investigated two departments covering 55 rooms with a total of 120 beds spread across 5 wards. This combination represents the pooled bed capacity for inpatients from the cardiology and gastroenterology departments.

We further obtained a detailed data set covering admission, discharge, as well as room transfer time stamps comprising the exact date and time on which each individual patient was actively booked in or out of a room and bed. In addition, the data set contains the department, age, gender, and care level of each individual patient and includes all data points recorded for the cardiology and gastroenterology departments between January 2013 and September 2016. In this context, it is important to note that the available data only represents ex-post data, representing “what actually happened”. However, in a real-life situation, it is often not clear before-hand how long a patient will stay as the anticipated discharge date is very likely to change throughout the stay of a patient. Thus, we tracked actual patient movements, as well as the actual predictions from physicians regarding the anticipated discharge date on site over the course of 4 weeks for all cardiology and gastroenterology patients on the associated wards. We then used these distributions and combined them with the ex-post data set at our disposal to prepare a series of event-based data points per patient which may be used to mimic all relevant information known to a potential bed planning system at a certain point in time. To this end, each data point comprises all patient-specific parameters and time stamps of anticipated arrivals or discharges as well as the exact time and date, on which these parameters and time stamps were last updated. Using the above-described approach, we apply nine data sets spanning from January 2016 to September 2016, with each set comprising all events, i.e., initial patients, admissions, discharges, and updates of LOS, occurring within a specified 28-day period. On average, every data set comprises 648 unique patients with around 2000 unique events taking place over the course of 28 days. This means that a deterministic problem instance was solved around 2000 times to simulate real-life application of our solution approach over time. The actual bed occupancy situation at the beginning of each time period is taken to initialize the calculation of each data set. All relevant patient parameters of current emergency and elective patient arrivals, future elective patient arrivals, as well as anticipated future emergency patient arrivals are taken into account to run the GLA heuristic. Due to the fact that almost all elective arrivals are known two weeks in advance, the time horizon taken into account for each run-through of the GLA heuristic was set to 14 days. In order to prevent overfitting, we used the available data from 2013 to 2015 to determine probability distributions for day-specific arrival rates, LOS, care level, department affiliation, and the age as well gender of emergency patients.

To test our modeling and solution approach, we compared the status quo, i.e., the actual PBA decisions taken in the case hospital, with the patient-bed allocations decisions that our GLA heuristic would have taken within the same time period. The GLA heuristic reruns every time a new event occurs, with the best known data currently available. After having undertaken all patient-bed allocations within the relevant data sets, the actual patient-bed allocations were analyzed ex-post facto based on the objective function of our decision model. We summarized the objective values as well as the relevant patient-specific, doctor-specific, and nurse-specific indicators and weighted them by the actual hours they were valid, respectively. In this context, it should be noted that this does not include the additional benefit OV_p attributed to patients coming from the overflow area, because it is merely an instrument to ensure that patients are not "left" in the overflow area. The "status quo", i.e., the ex-post evaluation of the patient-bed allocations that have actually taken place at the case hospital were used as a baseline. In addition, to assess the performance of our approach, we evaluated an "elective" scenario in which every future emergency patient and their characteristics as well as expected LOS are deterministically known beforehand, i.e., a scenario in which all patients are considered to be elective patients. However, the uncertainty of changes in LOS during the hospitalization are still existent in this scenario. Due to the remaining uncertainty, the result of the "elective" scenario is not necessarily better than the status quo. Nevertheless, it is expected that the "elective" scenario having significantly fewer uncertainty factors than the status quo achieves a higher solution quality.

Case study results Looking at the results of our case study in Table 4.7, the normalized values (normalized to the average of the status quo) of the objective function give a first indication of the performance of our approach. It can be noted that by using our GLA heuristic, i.e., the results in the columns termed "heuristic", it was possible to improve the patient-bed allocations for every available data set. In addition, the accumulated values of the objective function are fairly close for the "heuristic" and the "elective" scenario case. The application of our modeling approach significantly reduces the time patients have to spend in the overflow area and increases the average utilization. Utilization in this context is defined as the ratio of patients occupying a regular bed within their associated department-ward combination to the total number of beds within that department-ward combination. This is mainly due to three different effects with regard to the status quo. First, female and male patients are more efficiently combined to rooms such that situations in which several male-occupied rooms still have beds available while there is no room left for incoming female patients

Table 4.7: Case study analyses

	accumulated OF values (normalized[1])			**utilization** (in percent)		
DS	status quo	heuristic	elective scenario	status quo	heuristic	elective scenario
1	98.3%	103.0%	103.3%	76.5%	82.0%	81.9%
2	100.9%	105.3%	105.7%	77.8%	82.9%	82.9%
3	98.9%	103.4%	103.6%	76.3%	82.5%	82.5%
4	106.6%	111.5%	112.0%	81.9%	88.0%	88.0%
5	102.7%	106.1%	106.5%	78.4%	82.8%	83.2%
6	100.4%	103.3%	103.3%	77.6%	82.0%	81.9%
7	102.0%	106.1%	106.2%	78.7%	83.4%	82.8%
8	94.0%	97.1%	97.4%	72.1%	76.9%	76.9%
9	96.2%	99.6%	98.8%	73.5%	77.9%	77.5%
∅	100.0%	103.9%	104.1%	77.0%	82.0%	82.0%

	overflow (in hours)			**age difference** (average in years)		
DS	status quo	heuristic	elective scenario	status quo	heuristic	elective scenario
1	2951	14	140	11.6	4.9	4.4
2	2960	158	209	11.1	5.0	4.3
3	3174	89	95	11.1	5.4	5.2
4	3407	147	430	12.1	5.5	5.4
5	2692	64	132	10.5	5.1	5.3
6	2625	142	117	11.3	5.6	5.3
7	2354	111	426	10.5	5.1	4.7
8	2408	115	115	10.1	5.1	4.8
9	2223	63	29	9.9	5.1	4.0
∅	2754	100	188	10.9	5.2	4.8

	same department (percentage of rooms)			**care level surplus** (average excess of threshold)		
DS	status quo	heuristic	elective scenario	status quo	heuristic	elective scenario
1	73.9%	92.1%	92.7%	0.27	0.21	0.16
2	69.8%	94.3%	94.1%	0.21	0.24	0.10
3	86.1%	95.4%	94.5%	0.47	0.46	0.27
4	81.8%	93.6%	92.7%	0.59	0.16	0.09
5	87.4%	96.3%	95.5%	0.22	0.21	0.14
6	81.4%	93.3%	95.3%	0.39	0.46	0.29
7	77.8%	95.5%	95.0%	0.40	0.50	0.21
8	80.0%	95.2%	96.1%	0.17	0.23	0.07
9	83.1%	95.0%	95.3%	0.04	0.06	0.06
∅	80.1%	94.5%	94.6%	0.31	0.28	0.15

[1] normalized to the average of the status quo values

are prevented, and vice versa. Second, “standard patients” are less likely to block rooms and beds equipped with special infrastructure which they do not need. Third, medical isolation cases that may be combined, e.g., due to similar medical conditions, are more likely to be allocated to the same room instead of blocking multiple rooms. In comparison, the “elective” scenario actively generates slightly more overflow as all future emergency patients are already known which allows for trade-offs such that a slightly longer allocation of a patient to an overflow area may entail a better combination of patients in rooms and wards with regard to patient-specific, doctor-specific, and nurse-specific objectives. This is expected.

Furthermore, the results of our “heuristic” approach regarding average age difference, adherence to the same department, and care level all show significant improvements compared to the "status quo" scenario. In particular, it was possible to cut the average age difference in half while at the same time improving the percentage of rooms that only accommodate patients from a single department by 18%. Finally, it was also possible to decrease the total amount of additional workload for nurses exceeding the respective predefined thresholds per ward by 10%.

Table 4.8: Runtime analysis for one run-through

	runtime [sec]		
	average	**minimum**	**maximum**
data set 1	1.026	0.213	1.557
data set 2	0.949	0.220	1.487
data set 3	0.955	0.259	1.249
data set 4	1.034	0.329	1.394
data set 5	0.904	0.289	1.378
data set 6	1.115	0.353	1.495
data set 7	1.017	0.348	1.317
data set 8	0.995	0.272	1.341
data set 9	0.923	0.195	1.410
∅	0.991	0.275	1.403

Traced run-times of the case study are demonstrated in Table 4.8. The average for each data set is built over all run-throughs during the period of 28 days. Any event that may change the planned patient-bed assignments, i.e. an emergency arrival or an update of the LOS, triggers a rerun of the GLA heuristic. This ensures a planning decision based on all information known to the system at that particular point in time. Each of the nine datasets averaged around 950 total run-throughs of the algorithm, i.e., complete patient-bed allocation updates. For each complete update the average runtime was less than one second and the maximum runtime does not exceed 1.6 seconds. It should be noted, that this runtime comprises the complete replanning effort, i.e., the assignment of all patients $p \in P$ to all available beds $b \in B$. In summary, the case study proves that the developed GLA heuristic is suitable and applicable as decision support system for the daily use in a large hospital.

4.4.4 General applicability

In addition, we investigated the general applicability of our proposed approach. Therefore, we drew on the data sets made publicly available by Demeester et al. (2010) and applied our approach for each data set. Although these data sets do not include previous occupancies, uncertainty, or care levels of patients, they do provide several large-sized problem instances which may be used for assessing computation times.

Table 4.9: General applicability analyses

DS[1]	|P|	|B|	|T|	**utilization**	**age dif.**	**same**	**run time**
					[yr]	**dep.**	**[sec]**
1	693	286	14	60%	3	61%	2.5
2	778	465	14	60%	2.8	59%	4.7
3	757	395	14	57%	2.5	60%	3.8
4	782	471	14	54%	2.4	65%	4.7
5	631	325	14	49%	2.4	67%	2.5
6	726	313	14	64%	4.2	56%	3.5
7	770	472	14	34%	2.4	78%	1.5
8	895	441	21	44%	3.3	70%	4.3
9	1400	310	28	77%	11.2	40%	12.7
10	1575	308	56	48%	4.3	68%	17.6
11	2514	318	91	46%	3.6	71%	46.5
12	2750	310	84	55%	5.8	62%	55.8

[1] made publicly available by Demeester et al. (2010) for benchmarking purposes.

Our results which can be seen in Table 4.9 show that the overall utilization in Demeester's data sets is so low that short-term allocations of patients to an overflow area are basically not required. Nonetheless, even for the largest problem instance, i.e., data set 12 comprising 2750 patients and a planning horizon of 84 days, we found a solution with less than a minute computation time. The achieved average age difference of the data sets varies around 3 years with one outlier of 11.2 years for data set 9. This is caused by the fact that data set 9 involves a pediatric and geriatric department as well as high utilization. Thereby, to avoid overflow situations, the model is forced to combine patients with a large age gap. The percentage of patients adhering to the same department in one specific room per night ranges between 40% and 78%. This depends on the amount of specialities and the utilization.

In summary, the data sets provided by Demeester et al. (2010) significantly differ to what we encountered at our case hospital, in particular due to the unusually low utilization rates, as well as the lack of uncertainty in patient arrivals and updates of LOS. Thus, we additionally tested our model and solution approach with real-life data provided by the case hospital.

4.4.5 Sensitivity analyses

To better understand the trade-off effects that exist between the different objectives for patients, doctors, and nurses we created four additional scenarios in which we increased each of the four weighting factors α, β, γ, and δ by a factor of 10, respectively (see Table 4.10).

Table 4.10: Scenarios for sensitivity analyses

	base	**scenario**	**scenario**	**scenario**	**scenario**
	scenario	**1**	**2**	**3**	**4**
α	1	10	1	1	1
β	0.01	0.01	0.1	0.01	0.01
γ	0.2	0.2	0.2	2	0.2
δ	0.2	0.2	0.2	0.2	2

Each scenario is run with each of our nine real-life data sets. The results in Table 4.11 show the aggregated average values for each scenario and can be interpreted as follows. Throughout all four additionally created scenarios the utilization remains fairly constant around 82%. However, the individual results regarding overflow, age difference, department affiliation, and care level surplus show significant differences. This behavior is to be expected as the GLA heuristic will always try to fill up the available bed capacities. Nonetheless, significant trade-offs can be seen when focusing on optimizing age differences, department adherence per room, and workload for nurses. For instance, it can be seen that focusing optimization on parameters with a higher variance such as the age of patients significantly increases overflow as patients are “held back” in the overflow area to achieve even better pairings with other patients in the future. On the other hand, optimizing parameters with a low variance, such as department adherence, does not have a measurable effect on overflow. Focusing on age difference or department adherence significantly reduces the performance of the respective other parameter, as both parameters are room-specific, meaning that a trade-off has to be found. By contrast, the overall workload for nurses is ward-specific. Thus, a strong focus on balancing this workload does not lead to significantly worse values regarding age difference and department adherence. Finally, a strong focus on weighting factor α leads to a decrease in all observed objectives. This is due to the fact that in such a constellation, the GLA heuristic always prefers incoming and future elective patients regardless of how good a current emergency patient might match with an elective patient when allocated to the same room, thus leading to a slightly higher overflow of emergency patients.

Table 4.11: Sensitivity analyses for patient-, doctor-, and nurse-specific objectives

	utilization	**overflow**	**age difference**	**same department**	**care level surplus**
base scenario	82.0%	100	5.2	95%	0.28%
scenario 1	82.0%	84	6.3	90%	0.57%
scenario 2	81.2%	480	3.3	80%	0.25%
scenario 3	82.0%	67	11.5	98%	0.37%
scenario 4	82.0%	92	5.2	94%	0.13%

4.5 Conclusion and further areas of research

Conclusion The present paper presents a decision model that can be applied for ad-hoc operational bed allocation in large hospital settings. Most of the previous literature on PBA focuses on the developed model of Demeester et al. (2010) and his published fictive example data sets, either by providing alternative and/or improved heuristic solution approaches for the problem defined and/or by adding certain aspects to the problem. In our joint project with a large German hospital covering all major disciplines, we identified a variety of additional aspects which to the best of our knowledge have not been dealt with in the literature currently available. Based on the real-life situation our decision support model incorporates three main stakeholders, namely patients, nursing staff, and doctors. The developed model integrates the planning of current emergency and elective patient arrivals, future elective patient arrivals, as well as anticipated future emergency patient arrivals. To the best of our knowledge, we are the first who take into account all relevant stakeholders, extended patient-patient room dependencies, overflow situations, and the anticipation of future emergency patients as well as the possibility of a frequent replanning, which accounts for the uncertainty being inherent in the system. The model and solution approach developed is designed to very quickly propose a meaningful bed allocation to the bed manager for every incoming patient at the time of their arrival, based on all the information known at that particular moment. We developed a greedy look-ahead (GLA) heuristic that is suitable and applicable for daily use as an efficient and quick support system. In the numerical results, we have shown that

i) the GLA heuristic greatly outperforms Gurobi's MIP solver in terms of computational time while delivering a solution quality of 96.8% and higher

ii) our GLA heuristic can also sufficiently solve large data sets from previous literature,

iii) on the basis of real hospital data the GLA heuristic improved the objectives of all stakeholders, e.g., the overflow was reduced by 96%,

iv) the objectives of the stakeholders are highly dependent on one another.

Finally, the modularity of our proposed approach regarding standard objectives and constraints of the typical stakeholders along with the ability to solve large problem instances renders our proposed approach applicable for large hospitals anywhere in the world which cater to most major disciplines and exhibit high emergency rates. As such it is not limited to the German setting.

Future areas of research Various opportunities exist for further research. Based on our decision model a survey on different sophisticated heuristics can be conducted, focusing in detail on the trade-off between runtime and solution quality. In addition, a more detailed investigation on the effects of uncertainty regarding emergency arrival ratios and LOS estimates can be undergone. This would include investigating different ways of modeling uncertainties for the multi-objective PBA problem. It is also imaginable to include further stakeholders such as, for example, bed transport services. The modeling and solution approach presented in this paper may be an appropriate starting point to address these open research questions.

5 Combining Machine Learning and Optimization for the Operational Patient-Bed Assignment Problem

Co-autors: Manuel Walther, Dominik G. Grimm, Alexander Hübner
Submission planed to *OR Spectrum*

Abstract This paper develops a multi-objective decision support model for solving the patient bed assignment problem. Assigning inpatients to hospital beds impacts patient satisfaction and the workload of nurses and doctors. The assignment is subject to unknown patient arrivals and lengths of stay, in particular for emergency patients. Hospitals therefore need to deal with uncertainty on actual bed requirements and potential shortage situations as bed capacities are limited. This paper contributes by improving the anticipation of emergency patients using machine learning approaches, incorporating weather data, time and dates, important local and regional events, as well as current and historical occupancy levels. Drawing on real-life data from a large case hospital, we were able to improve forecasting accuracy for emergency inpatient arrivals. We achieved an up to 17% better root mean square error when using machine learning methods compared to a baseline approach relying on averages for historical arrival rates. Second, we develop a new hyper-heuristic for solving real-life problem instances based on the pilot method and a specialized greedy look-ahead heuristic. When applying the hyper-heuristic in test sets we were able to increase the objective function by up to 3% in a single problem instance and up to 4% in a time series analysis compared to current approaches in literature. We achieved an improvement of up to 2.2% compared to a baseline approach from literature by combining the emergency patient admission forecasting and the hyper-heuristic on real-life situations.

5.1 Introduction

This paper deals with the patient bed assignment problem (PBA). This is the operational problem of allocating elective and emergency inpatients to specific rooms and beds within a hospital upon their arrival. The key challenge in PBA is the inherent uncertainty that governs most input parameters. The planning situation is unstable due to frequent changes, which may be caused by emergency patient arrivals, changes in treatment plans and a number of other factors. For example, large maximum care hospitals are a natural first point of contact for all emergency patients within their catchment area, which naturally leads to a high ratio of unknown emergency inpatient arrivals. Thus, when assigning inpatients to beds in such environments, it is very important to anticipate the number of imminent emergency patient arrivals as best as possible, as emergency and elective inpatients can occupy the same ward space. Several circumstances and external effects may drive the volume of emergency patients, e.g., seasons, weekdays, local events (e.g., county fairs, sports events). There may be different drivers for each discipline (e.g., snowy weather for trauma surgery, availability of family doctor for internal medicine). Real-life planning typically involves several hundred patients and beds, such that it is not uncommon to be faced with a completely changed set of input parameters due to several updates in the system during the planning horizon. Moreover, the PBA affects patient satisfaction (e.g., suitable room with adequate roommates), workload of nurses (e.g., a mix of work-intensive and easy-to-handle patients) and workload of doctors (e.g., own patients located in proximity). These may comprise some tradeoffs. For example, focusing only on patient satisfaction by putting optimal roommates together may be in conflict with the nurse workload. As such, the PBA is a multi-objective problem that considers the tradeoff between patient-, nurse-, and doctor-specific objectives while taking into account their respective constraints as well as infrastructural requirements.

The remainder of this paper is structured as follows: Section 5.2 defines the PBA problem, discusses related literature, and further elaborates on the specific contribution of this paper. Section 5.3 introduces the mathematical model and the hyper-heuristic framework developed. It is based on the “preferred iterative look-ahead technique” (pilot method) of Duin and Voß (1999) and Voß et al. (2005), which in part incorporates the greedy look-ahead heuristic described in Atkinson (1994) as a subheuristic. Section 5.4 provides several numerical examples based on actual hospital data and details a machine learning approach developed to better anticipate emergency inpatient arrivals. In addition, we combine these insights obtained from machine learning with a hyper-heuristic framework for solving the PBA efficiently for large problem instances. Finally, Section 5.5 presents a summary of the main results and outlines potential avenues for further research.

5.2 Problem description, related literature and contribution

This section summarizes the general problem, discusses its complexity and related literature.

5.2.1 General planning problem

Scope of the patient bed assignment problem It is important to distinguish between the patient admission and scheduling problem (PAS) and the patient bed assignment problem (PBA), as these expressions have been used with varying definitions in the literature. We consider the PAS as only dealing with the scheduling of elective patient admission dates (see e.g., Gartner and Kolisch (2014); Gartner and Padman (2019)), whereas the PBA tackles the problem of allocating a specific room and bed to a specific inpatient (see e.g., Demeester et al. (2010); Ceschia and Schaerf (2011); Schäfer et al. (2019)). For the PAS it is not necessary to know which bed exactly will be held available for a certain inpatient as long as it is guaranteed to a certain extent that a bed will be available (see e.g., Ceschia and Schaerf (2016)). The PBA is the downstream decision with regard to the PAS.

Figure 5.1 presents an example of the PBA. Two female emergency patients who have just arrived are planned to stay in beds 3 and 4. While bed 1 is theoretically available before bed 3, it is already "reserved" for a male elective patient scheduled to arrive on Friday and stay for several days. Consequently, the female patient planned to occupy bed 3 will have to wait in an overflow area (e.g., hallways, emergency or treatment rooms) until Saturday when bed 3 becomes available for her. For this example, it is considered more important that the elective patient arriving on Friday does not have to wait in an overflow area. Hence, it is crucial to determine at which time a specific physical room and bed is to be assigned to a inpatient and whether or not it should be possible to reserve such a bed. In essence, there is always a tradeoff between different PBAs, which at times leads to situations where it may be beneficial to the overall utility to deviate from a first-come-first-served rule.

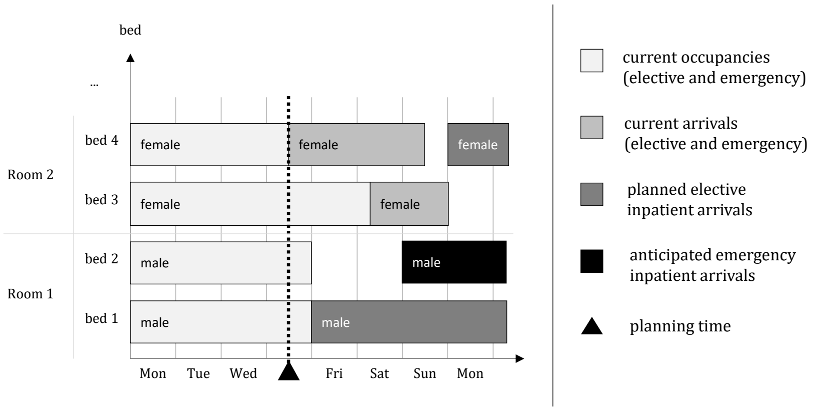

bed
...
bed 4
bed 3
bed 2
bed 1
Room 2
Room 1
female
female
female
female
female
male
male
male
male
Mon
Tue
Wed
Fri
Sat
Sun
Mon
current occupancies (elective and emergency)
current arrivals (elective and emergency)
planned elective inpatient arrivals
anticipated emergency inpatient arrivals
planning time

Figure 5.1: Illustration of the patient bed assignment problem

Objectives of patient bed assignment problem In general, patients want to have a room within a designated ward space that caters to their medical needs while avoiding unnecessary room transfers or having to wait in an overflow area. It is further desirable to combine similar patients, e.g., patients of similar age or with similar illnesses in terms of their specific medical condition and the severity thereof. In addition, elective patients typically do not accept that a room and bed within their respective department is not "reserved" for them upon their arrival, while emergency patients are more willing to accept having to temporarily stay in dedicated overflow areas. If staying in an overflow area does become necessary, patients wish to be transferred to a "regular room" as soon as possible. To facilitate doing rounds and patient visits, walking distances for doctors should be minimized. This can be achieved by grouping similar patients, i.e., patients associated with a specific department, into rooms. Compared to doctors, nurses can typically tend to a broader range of patients. However, they are typically dedicated to a specific ward, working in well-coordinated teams, and therefore cannot easily be transferred to other wards. Thus, balancing workload between wards is a key objective for nurses when assigning patients to beds (Schäfer et al., 2019).

Constraints of the patient bed assignment problem For the PBA, the following conditions have to be taken into account. First, non-ICU (intensive care unit) female and male inpatients are not allowed to be allocated to the same room. Second, certain medical conditions require patients to be in rooms, that are equipped with the necessary infrastructure, e.g., telemetry for selected cardiology patients. Third, it may be the case that a patient or several patients need to be isolated from other patients during their stay for medical reasons. Finally, there are usually no non-medically induced room transfers, meaning that assignments of patients who already physically occupy rooms associated with their designated department are treated as unchangeable. This is due to the fact that every physical room transfer entails significant additional work for hospital personnel (e.g., cleaning and sanitizing rooms, moving beds, reorganizing tasks) as well as unnecessary discomfort for the patient. In this context, the only exceptions are transfers due to medical reasons (e.g., transfers to and from the ICU).

5.2.2 Complexity of the patient bed assignment problem

In order to guarantee patient satisfaction and trouble-free process flow (i.e., avoid waiting times until inpatient admission as well as blocking emergency departments), bed mangers need real-time decision support. Furthermore, real-life planning situations are affected by many sudden changes (e.g., LOS update, no-shows and emergency patient admissions). Large hospitals in particular therefore require highly flexible planning systems, that are able to adapt to unexpected changes in real time. PBA complexity thus results from (1) being unable to precisely estimate the number of beds required and (2) the size of the problem of jointly planning hundreds of beds.

(1) Arrival and length of stay of patients Usually elective and emergency inpatients share the same ward space and bed capacities. This requires jointly planning the PBA for both types. Emergency inpatient arrivals are not known in advance and are stochastic, so they can only be estimated. Appropriately predicting which kind of emergency patients and how many are likely to arrive on a given day is a fundamental input to the PBA, particularly for large maximum care hospitals where up to 80% may be emergency patients. Simply predicting emergency patients based on historical averages will fall short, as – in addition to an inherent randomness – it seems highly probable that the actual number of emergency arrivals is dependent on a plethora of factors internal and external to the hospital, and cannot be explained solely by the time and date. For example, trauma surgery departments may experience an increase in emergency inpatients at the beginning of the cold season due to sidewalks that have frozen over, leading to more elderly people falling down and suffering a fracture. Furthermore, the LOS of a patient is always an informed estimate. Unforeseeable events such as sudden complications during surgery or treatment, faster recoveries, or patients who self-discharge against medical advice can potentially lead to a change in the LOS. Some disciplines exhibit high emergency arrival rates or are subject to more LOS updates. Finally, elective patients can also fail to show up for their planned inpatient stay. All this together leads to high volatility regarding future occupancy levels. In combination with the economic need for tight capacity and high occupancy levels, the volatility in patient volume inevitably leads to occasional overflow situations. In such cases, inpatients need to be temporarily assigned to overflow areas. Hallways, emergency or treatment rooms, or other wards outside a dedicated ward space may serve as buffers in such cases. Staying in such intermediate areas, however, is unpleasant for patients and will always entail additional work for nursing staff and doctors alike.

(2) Size of the problem To obtain better capacity utilization, departments of large hospitals now share ward space (see e.g., Essen et al. (2015) and Hübner et al. (2018)), which calls for jointly planning hundreds of beds and and efficient decision support. Whenever an elective or emergency patient is admitted to or discharged from wards, LOS are changed or no-shows of elective patients occur, or patients are reassigned from the overflow, the PBA needs to be updated. As such, the underlying planning problem has to be solved many times per day. To illustrate, one can for instance assume a scenario comprising a pooled capacity of 1,000 beds exhibiting an average utilization of 90%, with patients who stay three days on average. This would lead to an average of 300 inpatient arrivals and 300 inpatient dismissals per day, respectively. Further assuming an emergency ratio of 50% would mean that at least 150 of said arrivals are subject to fluctuations to the bed planning system beforehand. In addition, one can for example assume that 50% of the remaining elective PBAs, i.e., 75 arrivals, are somehow affected by a sudden change in LOS of any of the current occupants. In total, this would lead to an average of 225 additional events during the day, for which all future PBAs have to be recalibrated. Further changes in LOS updates for patients already occupying a room, no-shows of elective patients and overflow situations will increase the number of events. In overflow situations unexpected inpatient dismissals could then directly affect potential PBAs of any patients currently waiting within an overflow area.

5.2.3 Related literature

The problem at hand is related to decision models for the PBA and relies on estimating emergency patients. We structure the literature review in these two areas, and derive the associated open research areas in each section.

Decision models and related literature for patient bed assignment

The PBA has gained more and more attention mainly within the past decade. Key challenges dealt with in most contributions to this area of research can be seen in the computational complexity of typical problem sizes and the resulting need for heuristic solution approaches, as well as the underlying uncertainty and volatility of most parameters involved. The PBA at present has not yet received the same attention from the research community as classic OR applications in healthcare such as surgery scheduling or nurse rostering. Table 5.1 gives an overview of most of the recent contributions and highlights a set of key aspects related to the challenges mentioned above. With regard to the modeling approach followed, the table indicates whether a “static” or “dynamic” problem setting has been investigated. In this context, “static setting” refers to a hypothetical scenario in which every future arrival is known and no prior occupancies are considered, whereas in the “dynamic setting” prior occupancies are considered, while future arrivals are only known within a defined planning horizon. In addition, the table indicates whether “emergency patients” are considered in the modeling approach, i.e., whether or not the potential arrival of inpatients is considered, which cannot be known in advance. Furthermore, under “overflow possible” we indicate whether a specific modeling approach is designed to deliver feasible solutions to the PBA when faced with problem instances in which not enough beds are available for all arriving inpatients. “Uncertainty considered” refers to the modeling of volatility in patient parameters, e.g., future changes in LOS. Finally, the column “stakeholders” indicates for which group, i.e., patients, nurses, and doctors, objectives and constraints are included within the mathematical model. Table 5.1 further indicates analyses that have been undertaken in the different contributions. Here, the column “emergency forecast” indicates whether emergency inpatient arrivals were analyzed beyond the effects of using simple historical occupancy distributions. Furthermore, the column “time series” indicates whether the continuous application of a PBA algorithm over the course of several days or weeks was analyzed. In essence, this means analyzing the actual accumulated partial benefits or costs incurred by each patient stay in retrospect. Finally, the column “data sets used” indicates whether simulated data or real-life hospital data was used.

Table 5.1: Overview of decision models related to patient bed assignment

		Modeling approach					Analyses		
Contribution	Basic solution approach	Problem setting[1]	Emergency patients	Overflow possible[2]	Uncertainty[3]	Stakeholders[4]	Emergency forecast[5]	Time series[6]	Data sets used[7]
Demeester et al. (2010)	Tabu search	S	–	–	–	P	–	–	S
Bilgin et al. (2012)	Hyper-heuristic	S	–	–	–	P	–	–	S
Kifah and Abdullah (2015)	Great deluge	S	–	–	–	P	–	–	S
Turhan and Bilgen (2017)	Fix-and-optimize heuristic	S	–	–	–	P	–	–	S
Guido et al. (2018)	Matheuristic	S	–	–	–	P	–	–	S
Bastos et al. (2019)	Exact approach	S	–	–	–	P	–	–	S
Dorgham et al. (2019)	Genetic algorithm	S	–	–	–	P	–	–	S
Taramasco et al. (2019)	Metaheuristics	S	–	–	–	P	–	–	R
Ceschia and Schaerf (2011)	LNS	S,D	–	–	✓	P	–	–	S
Ceschia and Schaerf (2012)	LNS	S,D	✓	–	✓	P	–	–	S
Ceschia and Schaerf (2016)	LNS	S,D	✓	–	✓	P	–	–	S
Lusby et al. (2016)	Adaptive LNS	D	✓	–	✓	P	–	–	S
Vancroonenburg et al. (2016)	Specialized heuristic	S,D	✓	–	✓	P	–	–	S
Schäfer et al. (2019)	Greedy look-ahead heuristic	S,D	✓	✓	✓	P,N,D	–	✓	S,R
This Paper	Hyper-heuristic based on Pilot; ML for emergency forecast	S,D	✓	✓	✓	P,N,D	✓	✓	R

[1] S = static version of the PBA (every future arrival known, no prior occupancies)
D = dynamic version of the PBA (prior occupancy considered, arrivals only known within planning horizon)

[2] Situations in which not enough beds are available and in which patients have to spend time in an overflow buffer until a bed becomes available

[3] uncertainty regarding patient LOS and future admission dates

[4] objectives and constraints considered for patients (P), nurses (N), and doctors (Doc)

[5] emergency inpatient arrival forecast (analysis of effects beyond using simple historical averages)

[6] time series version of the PBA (analysis of effect that continuous application of a PBA-algorithm has over the course of several weeks)

[7] S = simulated problem instances, R = real hospital data

Static models for patient bed assignments The PBA was first introduced by Demeester et al. (2010). They consider a situation in which a hospital is initially empty and all future patient arrivals within a given time horizon are deterministically known as well as their respective parameters, e.g., actual LOS, gender, department adherence, individual infrastructural needs. The model is formulated as a static model where the assignments are only made once to populate the hospital. This static version is only a single problem instance in which all future arrivals are known, without considering any prior occupancies. In their model, patients are assigned to rooms such that an overall objective function based on violating patient-specific requirements is minimized. The model acknowledges gender-specific room assignment, assignment of patients to departments suited to their age, availability of relevant infrastructure, adherence to medical isolation and patient-specific room type preferences (e.g., single or double room). Patients are assigned to available rooms of a certain type while taking known admission and discharge dates of each patient into account. Capacity is assumed to be sufficient to accommodate all inpatients. As such, it does not allow for overflow situations, i.e., problem instances in which not enough beds are available for all inpatients cannot be solved. Furthermore, they do not consider nurse- and doctor-specific objectives and do not distinguish between emergency and elective patients. They apply a token-ring tabu search.

Several authors have since built on the model developed by Demeester et al. (2010) by providing alternative or improved solution approaches and/or by introducing new aspects to the PBA. Bilgin et al. (2012) use the model provided by Demeester et al. (2010) and solve the static version of the PBA by applying a hyper-heuristic approach using simulated annealing and a tabu search. Kifah and Abdullah (2015) and Bastos et al. (2019) also provide new solution approaches to the static version of the PBA model as proposed by Demeester et al. (2010). In particular, Kifah and Abdullah (2015) propose a variant of a generic algorithm, i.e., an adaptive non-linear great deluge heuristic, whereas Bastos et al. (2019) propose an MIP formulation and use sparcity conditions to find optimal solutions for some problem instances of Demeester et al. (2010). To decrease the computational complexity, Ceschia and Schaerf (2011) have proposed a reformulated version of the mathematical model originally proposed by Demeester et al. (2010). Specifically, Ceschia and Schaerf (2011) reformulate the model such that patients are only assigned to rooms rather than beds, as they consider the beds in each room to be identical. Based on this reformulated version, Turhan and Bilgen (2017), Guido et al. (2018), and Dorgham et al. (2019) have presented solution approaches to the PBA. For instance, Turhan and Bilgen (2017) also focus on improving the static version of the PBA problem and investigate the effects of using a fix-and-optimize heuristic. In addition, Guido et al. (2018) further investigate the impact of switching hard and soft constraints in the PBA and develop a matheuristic that focuses on providing tighter bounds on the search space. More recently, Dorgham et al. (2019) have proposed a further variant of a genetic algorithm combined with a hybrid simulated annealing approach. Taramasco et al. (2019) on the other hand have taken a slightly different modeling approach to the PBA. Specifically, they investigate a network of hospitals and divide the PBA into two stages. In a first stage patients are assigned to beds within a specific hospital, while in a subsequent second stage patients who cannot be assigned an adequate bed are redistributed among the other hospitals within the network. In addition, Taramasco et al. (2019) are one of the few who have investigated the static version of the PBA using real-life hospital data. To solve their model for large problem instances, they propose a metaheuristic, which is a composition of different specialized and evolutionary heuristics and approximate methods.

Investigating the static version of the PBA provides a valuable controlled test environment, which has been used in the literature to compare different modeling and solution approaches. However, for real-life applicability, a proposed modeling and solution approach for the PBA needs to be able to handle dynamic online planning situations, i.e., problem instances in which some beds are already pre-occupied and in which not all future arrivals are fully known to the system. In addition, for large hospitals using pooled ward capacities and experiencing high ratios of emergency arrivals, it is especially important to incorporate potential emergency inpatients into the bed assignment planning. First and foremost this requires having good emergency arrival forecasts. Furthermore, when pairing high occupancy rates with high emergency arrival rates, overflow situations are likely to arrive that have to be handled by the PBA system.

Dynamic models for patient bed assignments Ceschia and Schaerf (2011) are the first to provide an approach for adapting the PBA model and solution approach to the dynamic case. To this end, they include the notion of an individual “registration date” per patient, i.e., the date the arrival of the patient becomes known to the system. The number of days an arrival is known in advance can vary for elective patients and can be considered to equal zero for emergency arrivals. However, emergency patients are not treated any differently to elective patients once they are known to the system. In addition, Ceschia and Schaerf (2011) consider pre-occupancies, i.e., patients who are already in the hospital at the planning date, whereby each PBA that has happened before said date is considered fixed. To test their approach they draw on simulated data by Demeester et al. (2010) and adapt the information in a reasonable but arbitrary way. Furthermore, Ceschia and Schaerf (2011) provide an approach to investigating the uncertainty regarding the discharge date of a patient that is inherent in the dynamic problem setting. To assess the impact of different LOS they solve the PBA several times using different values for the discharge dates of all patients in the system. Specifically, they start by assuming that each patient leaves after one day and add a day to the LOS of each patient, respectively, until the actual discharge date is reached.

In their subsequent work (Ceschia and Schaerf (2012) and Ceschia and Schaerf (2016)) they further include uncertainty by factoring in flexible horizons and patient delays while also adding operating room constraints. Based on the work of Ceschia and Schaerf (2012), Lusby et al. (2016) further provide an alternative solution method to the PBA under uncertainty. Specifically, they develop an adaptive search procedure. Vancroonenburg et al. (2016) tackle the dynamic PBA setting by providing a first model that is designed to only assign those patients to a new room who have just arrived and physically require a bed. They use a graph-based approach in which they use maximal cliques to respect room capacity constraints. In addition, they suggest a second model in which they also assign patients to beds who are registered in the system but have not yet arrived. This approach uses "dummy rooms" that are only "open" to patients who have not yet arrived in order to ensure feasibility of the model in undercapacity situations. Schäfer et al. (2019) have developed a comprehensive model and a specialized solution approach for solving the PBA. Their model distinguishes between emergency and elective patients and incorporates their respective needs and constraints as well as those of doctors, nurses, and management. In addition, it is designed to handle ad-hoc overflow situations, should they arise. Finally, it incorporates and evaluates patient-patient dependencies with regard to rooms and wards.

For real life situations in large hospitals, it is important to have a decision support system that is proven to work in a dynamic online scheduling scenario. At a minimum, this requires a solution approach that can deal with ad-hoc overflow situations and emergency inpatient arrivals. In addition, the underlying volatility of patient LOS and emergency arrival rates typically requires several adaptations of future PBAs during any given day. The performance of any such support system can thus only be measured by retrospectively evaluating actual occupancy. To the best of our knowledge, Schäfer et al. (2019) are the only ones to have analyzed the performance of their modeling and solution approach over a time series. However, there is still a need for better parameter forecasting for the dynamic problem setting and testing.

Further literature related to patient bed assignment To complete the picture, we additionally review the following modeling papers to highlight further aspects, that are considered relevant to the PBA problem context. For instance, bed capacity related issues are addressed by the following authors. Essen et al. (2015) and Hübner et al. (2018) develop approaches to combine departments and wards to pool bed capacities. Vanberkel et al. (2012) use a queuing model to investigate the tradeoffs between centralizing hospital resources and decentralizing. Holm et al. (2013) use a discrete event simulation model to analyze patient flows and optimize the assignment of bed capacities between wards. Bekker et al. (2016) investigate the issue of partially flexible ward capacity and how much should be attributed to a general overflow area. Another example for handling overflow situations can be found in Herring and Herrmann (2012) who investigate the effects of deferring surgical patients while blocking surgical capacity for higher priority cases. Cotta (2011) investigate the effects of patient prioritization in a mass casualty scenario. With regard to patient admission, for instance, Gartner and Padman (2019) build on and extend Gartner and Kolisch (2014)'s approach to solving the PAS. They focus on the assignment of hospital resources and provide a mathematical program, that, among other things, includes flexible patient assignments to medical departments to account for multi-morbid patient clientele, as well as overtime availability of medical and nursing staff. Luscombe and Kozan (2016) provide a dynamic scheduling framework that relates to parallel machine and flexible job shop problems to provide a decision support model for patient assignment in emergency departments.

Open research with regard to modeling and solving PBAs As to the operational assignment of beds, the actual problem at hand is a dynamic online planning situation in which the PBA needs to be solved several times per day. That means at each point in time that an inpatient gets admitted or discharged or when any other change in the system merits moving patients from an overflow area to a regular bed. Alternative heuristics are required to address the dynamic problem. As pointed out above, a key performance indicator for any such heuristic is the retrospective assessment of actual occupancies over time. To the best of our knowledge, Schäfer et al. (2019) are the only ones to have provided such a time-series analysis using a deterministic greedy look-ahead heuristic. However, there is still a need to investigate more sophisticated heuristic approaches using different parameter settings, especially when using non-deterministic solution approaches.

Literature related to estimating emergency patients

One of the key drivers of uncertainty regarding bed management in large hospitals is the large ratio of emergency inpatient arrivals, which for certain medical specialties such as cardiology can surpass 80%. Carvalho-Silva et al. (2018) as well as Afilal et al. (2016) concern themselves with the problem of forecasting emergency arrivals at a hospital. Both use real-life hospital data and analyze their data using an autoregressive moving average approach.

Schiele et al. (2019) provide a model to anticipate resulting bed occupancy levels based on a given master surgery schedule. They consider different patient types and paths and make use of a neural network based approach to improve their prediction quality. In addition, several authors have dealt with forecasting emergency arrivals in general, e.g., outpatient arrivals, day clinic walk-ins, or emergency calls, as can be seen in a systematic review written by Wargon et al. (2009). More recently, Gul and Celik (2018) have reviewed and analyzed contributions on applications of statistical forecasting in emergency departments.

Open research with regard to estimating emergency patients for PBA As pointed out above, anticipating emergency arrivals as accurately as possible is key for the PBA. Our literature review shows that advanced methods to better anticipate emergency inpatient arrivals, e.g., deep learning, are rare in general and not available for the specific problem of assigning inpatients to beds. To this end, a broader investigation with combined effects such as detailed weather data, holidays, seasons or significant local events is required. This will allow the prediction of emergency arrivals more accurately compared to solely drawing on historical averages and distributions of patient arrivals. Such an approach is promising as it relies on publicly available data and as such is possible to be incorporated in existing planning systems. To the best of our knowledge, such an integrated approach to forecasting emergency inpatient arrivals for the PBA has not yet been proposed in the literature.

5.3 Modeling and solution approach

5.3.1 Model complexity, general idea of solution approach and model overview

Complexity and general idea This section gives an overview of the modeling approach and develops an efficient heuristic to account for the requirements of large hospital settings. The underlying problem of the PBA could be represented as a stochastic dynamic program. The dynamic setting of the problem arises from multiple events such as arrivals, discharges and no-shows of patients as well as changes in LOS. Here, each event represents a stage and the total number of inpatients constitutes the state space in each stage. To illustrate, when assuming the case of a large hospital with about 800 beds occupied on average, a planning horizon of 28 days and an average of over 500 events per day, this would result in more than 14,000 stages and a total state space of more than 11 million entries. The stochastic volatility arises from the fact that the total number and type of inpatients cannot be predetermined and are further subject to uncontrollable external influences (such as weather, patient recovery, treatment complications, etc.). In light of this, it becomes obvious that such a dynamic problem setting cannot be solved optimally, meaning that a heuristic approach is required if one wants to provide efficient and effective decision support in real-life settings. We approximate the dynamic problem as Schäfer et al. (2019) by solving a static model that is updated at each possible event. Ceschia and Schaerf (2011) propose a similar approach to test the performance of their static model in a dynamic setting. When solving the model, it allocates beds for patients (new inpatients and patients from overflow buffer), assigns patients to overflow, and reserves beds for patients (currently in overflow and future patient arrivals). As such, we subsequently solve single stages while considering future arrivals and discharges that are both already known and estimated. The model takes all the relevant information currently available into account for each of these individual stages.

Model overview The decision model is based on Schäfer et al. (2019). A multi-objective utility maximization problem quantifies patient-specific, doctor-specific, and nurse-specific objectives, while simultaneously considering medical, gender, and capacity constraints. The model builds in the possibility of using a buffer for situations where the number of beds is insufficient or beds may be blocked for patients arriving later. Table 5.2 summarizes the notation.

Table 5.2: Notation

Sets	
B	Set of beds, $B = \{1, 2, ..., b, ..., \|B\|\}$
D	Set of medical departments, $D = \{1, 2, ..., d, ..., \|D\|\}$
P	Set of inpatients, $P = \{1, 2, ..., p, ..., \|P\|\}$
R	Set of rooms, $R = \{1, 2, ..., r, ..., \|R\|\}$
T	Set of days within the planning horizon, $T = \{1, 2, ..., t, ..., \|T\|\}$
W	Set of wards, $W = \{1, 2, ..., w, ..., \|W\|\}$
Parameters	
$\alpha, \beta, \gamma, \delta$	Weights for basic and extended patient-, doctor- and nurse-related utilities, respectively
Ξ_p	Weight for patient types (e.g., elective vs. emergency patient)
a_p	Age of patient p
$A_{rt}^{\max}$ ($A_{rt}^{\min}$)	Maximum (minimum) age of all patients already occupying room r on day t
C_{wt}	Spare care capacity for caring further patients on ward w on day t
c_p	Care level required to accommodate patient p
D_{rt}	1 if all prior occupants of room r on day t belong to the same medical department; 0 otherwise
d_p	Medical department of patient p with $d_p \in D$
E_{bt}	1 if bed b is located in a room that is initially empty on day t; 0 otherwise
F_{rt}	1 if room r is initially empty on day t; 0 otherwise
g_p	-1 if patient p is male; 1 if patient p is female
i_p	$i_p = -1$ if patient p requires medical isolation; 1 otherwise
K_{br}	1 if bed b is in room r; 0 otherwise
L_{bw}	1 if bed b is in ward w; 0 otherwise
OF_p	Utility parameter of patient p depending on the time patient p already spent in overflow
Q_t	Time-dependent relevance value that arrivals/discharges will take place as anticipated/planned on day t
s_{bpt}	1 if bed b is available for patient p on day t; 0 otherwise

Continued on next page

Table 5.2 – *Continued from previous page*

Decision variable	
x_{bp}	1 if patient p is assigned to bed b; 0 otherwise
Auxiliary variables	
$a_{rt}^{\max}$ $(a_{rt}^{\min})$	Maximum (minimum) age of all patients p assigned to room r on day t
o_{wt}^{+}	Amount of exceeding the total care capacity on ward w on day t
y_{rt} (z_{rt})	1 if all patients assigned to an empty (partially occupied) room r on day t are from the same medical department; 0 otherwise

The objective function of Equation (5.1) maximizes the total utility U and consists of four terms that represent basic patient-specific objectives, extended patient-specific objectives, doctor-specific objectives and finally nurse-specific objectives. The four partial utilities are weighted by the factors α, β, γ, and δ. All four utility values depend on the binary assignment variable x_{bp} that represents whether a patient $p, p \in P$ is allocated to bed $b, b \in B$. The objective function and set of constraints is formulated as follows:

$$\begin{aligned}\text{maximize } U = \alpha \sum_{b \in B} \sum_{p \in P} (\mathrm{OF}_p + \Xi_p \sum_{t \in T} s_{bpt} Q_t) x_{bp} - \beta \sum_{r \in R} \sum_{t \in T} (a_{rt}^{\max} - a_{rt}^{\min}) \\ + \gamma \left[\sum_{r \in R} \sum_{t \in T} f_{rt} y_{rt} + \sum_{r \in R} \sum_{t \in T} (1 - f_{rt}) z_{rt} \right] - \delta \, (\sum_{w \in W} \sum_{t \in T} o_{wt}^{+})\end{aligned} \tag{5.1}$$

subject to

$$\sum_{b \in B} x_{bp} \leq 1 \qquad \forall p \in P \tag{5.2}$$

$$\sum_{p\in P} s_{bpt}x_{bp} \leq 1 \qquad \forall b \in B; t \in T \tag{5.3}$$

$$\sum_{t\in T} s_{bpt} \geq x_{bp} \qquad \forall b \in B; p \in P \tag{5.4}$$

$$K_{br}E_{bt}g_p s_{bpt}x_{bp} - K_{lr}E_{lt}g_h s_{lht}x_{lh} \geq -1 \qquad \forall b, l \in B; p, h \in P; r \in R; t \in T \tag{5.5}$$

$$K_{br}E_{bt}i_p s_{bpt}x_{bp} - K_{lr}E_{lt}i_h s_{lht}x_{lh} \geq -1 \qquad \forall b, l \in B; p, h \in P; r \in R; t \in T \tag{5.6}$$

$$a_{rt}^{\max} \geq A_{rt}^{\max} \qquad \forall r \in R; t \in T \tag{5.7}$$

$$a_{rt}^{\max} \geq K_{br}a_p s_{bpt}x_{bp} \qquad \forall b \in B; p \in P; r \in R; t \in T \tag{5.8}$$

$$a_{rt}^{\min} \leq A_{rt}^{\min} \qquad \forall r \in R; t \in T \tag{5.9}$$

$$a_{rt}^{\min} \leq \sum_{b\in B}\sum_{p\in P} A_{rt}^{\min}K_{br}s_{bpt}x_{bp} \qquad \forall r \in R; t \in T \tag{5.10}$$

$$a_{rt}^{\min} \leq K_{br}a_p s_{bpt}x_{bp} + A_{rt}^{\min}(1 - x_{bp}) \qquad \forall b \in B; p \in P; r \in R; t \in T \tag{5.11}$$

$$K_{br}d_p s_{bpt}x_{bp} - K_{lr}d_h s_{lht}x_{lh} \geq -\mathrm{M}(1 - y_{rt}) \qquad \forall b, l \in B; p, h \in P; r \in R; t \in T \tag{5.12}$$

$$\sum_{p \in P} \sum_{b \in B} K_{br} s_{bpt} x_{bp} \geq y_{rt} \qquad \forall r \in R; t \in T \tag{5.13}$$

$$K_{br} d_p s_{bpt} x_{bp} - D_{rt} \leq \mathrm{M}(1 - z_{rt}) \qquad \forall b \in B; p \in P; r \in R; t \in T \tag{5.14}$$

$$D_{rt} - K_{br} d_p s_{bpt} x_{bp} \leq \mathrm{M}(1 - z_{rt}) \qquad \forall b \in B; p \in P; r \in R; t \in T \tag{5.15}$$

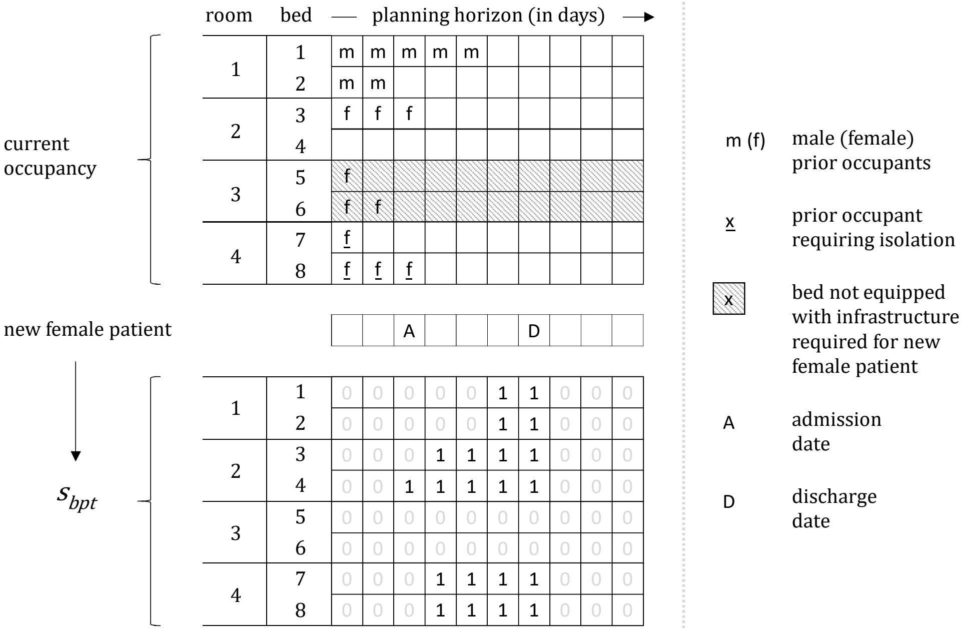
$$\sum_{p \in P} \sum_{b \in B} K_{br} s_{bpt} x_{bp} \geq z_{rt} \qquad \forall r \in R; t \in T \tag{5.16}$$

$$\sum_{b \in B} \sum_{p \in P} L_{bw} c_p s_{bpt} x_{bp} \leq C_{wt} + o^+_{wt} \qquad \forall t \in T; w \in W \tag{5.17}$$

$$o^+_{wt} \geq 0 \qquad \forall t \in T; w \in W \tag{5.18}$$

$$x_{bp}, y_{rt}, z_{rt} \in \{0, 1\} \qquad \forall b \in B; p \in P; r \in R; t \in T \tag{5.19}$$

The first term of the objective function in Equation (5.1) summarizes the basic patient-specific utility of assigning patient $p, p \in P$ to bed $b, b \in B$. Every assignment of a patient p to a bed b, i.e., $x_{bp} = 1$ generates a utility that accounts for the days that patient p is presumed to spend in bed b within the planning horizon T. The utility depends on the time the patient p already spent in the overflow (OF_p) in the past, a patient type-specific factor (Ξ_p), bed availability (s_{bpt}), and a relevance value (Q_t). The incorporation of an overflow value in the first part of the utility function allows patients already waiting in the overflow area to be assigned a higher preference than similar patients who have just arrived in the hospital. The second part of the utility function rewards the actual time that a patient p spends in bed $b \in B$. The parameter Ξ_p is a factor that makes it possible to distinguish between patient types, i.e., elective patients, emergency patients, or patients with special requirements. This factor may, for example, be used to ensure that elective patients are more likely to be assigned to a bed within their target ward upon arrival than emergency patients. In addition, patients returning from the ICU could be attributed an even higher value such that they will not be moved to an overflow area. The parameter s_{bpt} is set to 1 in the event that bed b is available for patient p on day t, and 0 otherwise. As non-medical room transfers are not allowed, s_{bpt} is determined at each event and is used to reflect not only bed availability but also bed compatibility by incorporating gender constraints (with respect to current occupants), infrastructural constraints, as well as medical isolation constraints (with respect to current occupants) for each possible patient bed combination. Figure 5.2 shows an example illustrating how parameter s_{bpt} is determined. The upper part represents the current occupancy and the lower part the determination of s_{bpt}. The parameter $s_{bpt} = 1$ if the respective bed is available for this patient on this day, otherwise there is no entry, meaning that $s_{bpt} = 0$. The example considers four rooms, each with two beds. A new female patient arrives on day 3 and is scheduled to be discharged on day 8. There are multiple options for allocating her to a bed. Male patients occupy room 1 with beds 1 and 2. Currently, the earliest availability of bed 1 and 2 for a female patient is day 6 after patient in bed 1 leaves. Therefore there are no entries in s_{bpt} for days 1 to 5. As she is scheduled to leave on day 8, day 8 to the end of the planning horizon has also no entry. Hence, assigning her to room 1 would result in spending at least two days in the overflow area. Bed 3 is available from day 4 and bed 4 is directly available. Beds 5 and 6 are not allowed to be used by this inpatient as this room is not equipped with essential medical infrastructure specifically required for this patient. Finally, female patients currently occupy both beds 7 and 8. They require medical isolation from non-quarantined patients for the duration of their stay, hence forcing the new patient to spend one day in the overflow area before moving into either of these beds, should she be allocated to one of them.


room
bed
planning horizon (in days)
current occupancy
new female patient
s_{bpt}
m (f) male (female) prior occupants
x̲ prior occupant requiring isolation
x bed not equipped with infrastructure required for new female patient
A admission date
D discharge date

room	bed										
1	1	m	m	m	m	m					
	2	m	m								
2	3	f	f	f							
	4										
3	5	f	▨	▨	▨	▨	▨	▨	▨	▨	▨
	6	f	f	▨	▨	▨	▨	▨	▨	▨	▨
4	7	f̲									
	8	f̲	f̲	f̲							

new female patient										
			A			D				

room	bed										
1	1	0	0	0	0	0	1	1	0	0	0
	2	0	0	0	0	0	1	1	0	0	0
2	3	0	0	0	1	1	1	1	0	0	0
	4	0	0	1	1	1	1	1	0	0	0
3	5	0	0	0	0	0	0	0	0	0	0
	6	0	0	0	0	0	0	0	0	0	0
4	7	0	0	0	1	1	1	1	0	0	0
	8	0	0	0	1	1	1	1	0	0	0

Figure 5.2: Example for determining parameter s_{bpt} for a new arriving female patient

Finally, Q_t is a parameter that reflects the time-dependent relevance of a bed assignment for patients on day t as anticipated/planned where Q_t is decreasing with increasing t. It gives an higher value to earlier arriving patients than those that come later in the planning horizon. Due to uncertainties it is quite reasonable that a patient, who is planned to arrive far in the future, will be reassigned to another bed at later planning periods, which may then even lead to a higher overall utility value for that patient. Equations (5.2) prevent double booking, i.e., a patient can only be allocated to a maximum of one bed. Equations (5.3) prevent overbooking, i.e., no two patients can be allocated to the same bed on the same day. Equations (5.4) ensure that a patient p can only be assigned to a bed b if bed b is available for this specific patient on at least one day during the stay, i.e., $s_{bpt} = 1$ for at least one $t \in T$. In addition, Equations (5.5) ensures that there are no mixed male and female rooms on any given day t. Using a similar approach, Equation (5.6) ensures that medical isolation requirements are respected.

Specifically, patients that need to be isolated due to infectious diseases, for example, may only be put into empty rooms or into rooms with patients that suffer from the same condition.

The second term of Equation (5.1) represents the extended patient-specific part. It evaluates the compatibility between different patients occupying one room. The goal is to minimize the differences between patients within rooms since it is desirable to combine similar patients. We use age difference as an indicator for the compatibility between patients (see also Schäfer et al. (2017)). Other indicators such as social status, education level, personal background etc. could also be applied in our model with the same logic. In particular, $a_{rt}^{\max} - a_{rt}^{\min}$ denotes the age difference between the oldest and the youngest patient in room r on day t. As such, both auxiliary variables $a_{rt}^{\max}$ and $a_{rt}^{\min}$ are dependent on x_{bp} as well as on the patients already occupying beds. $A_{rt}^{\max}$ ($A_{rt}^{\min}$) is set to the current maximum (minimum) age of all patients already occupying room r on day t. If room r is empty on day t, $A_{rt}^{\max}$ is set to a large integer value that represents the maximum possible age (e.g., 150), and $A_{rt}^{\min}$ is set to 0. Equations (5.7) and (5.8) ensure that the auxiliary variable $a_{rt}^{\max}$ reflects the maximum age of prior occupants and newly allocated patients in a room r on day t. Likewise, Equations (5.9) to (5.11) e ensure the same for $a_{rt}^{\min}$ while also making sure that $a_{rt}^{\min}$ equals $a_{rt}^{\max}$ in the event that room r is only occupied by one person or completely empty on day t.

The third term of Equation (5.1) rewards assigning only patients of the same department to specific rooms. Medical rounds for doctors are easier when several patients they are responsible for are in the same room. In addition, walking distances are reduced. Here we need to differentiate between empty and partially occupied rooms. This is indicated by the parameter f_{rt}, which is 1 if room r is empty on day t, and 0 otherwise. Two auxiliary variables y_{rt} and z_{rt} are applied:

- Empty rooms: The auxiliary variable y_{rt} is set to 1 if all patients assigned to an empty room r on day t are from the same medical department, which is achieved by Equations (5.12) and (5.13). Here, d_p is an integer value that depicts the medical department of patient p and M represents an arbitrary large integer value (“big M”).

- Occupied rooms: The auxiliary variable z_{rt} is set to 1 only if all patients assigned to room r are already from the same department. This is achieved by Equations (5.14) to (5.16). Here, D_{rt} is set to 1 if all prior occupants of room r on day t belong to the same medical department, and 0 otherwise.

Finally, the fourth term of the objective function (5.1) is used to balance the workload for nursing staff. This requires the matching of care requirements of patients and care capacity on wards. The specific number of “care units” for every patient p is quantified with c_p. This represents the effort and resources that go into taking care of that particular patient. The available number of nursing staff and thus, workforce or total “care capacity” per ward w and day t is predetermined due to shift schedules, staff rosters, and cannot easily be changed on short notice. Parameter C_{wt} represents the current spare capacity of a given ward w on day t for caring for newly arriving patients (i.e., available capacity, being the delta of the total capacity minus the capacity reserved for current patients in this ward). Exceeding a predefined care capacity per ward w on day t needs to be penalized. The amount by which the capacity of a ward w on day t is exceeded is represented by the auxiliary variable o^+_{wt}. Equations (5.17) and (5.18) link x_{bp} to o^+_{wt}.

5.3.2 Hyper-heuristic

This subsection develops the solution approach. Bed managers require a rapid system in everyday work that provides real-time decision support for each new event. An optimal solution approach is impracticable with respect to the combinatorial complexity of the PBA. Other approaches in the literature (see for example Demeester et al. (2010), Ceschia and Schaerf

(2011)) also had to resort to using heuristic approaches for the same reasons. Schäfer et al. (2019) propose a GLA heuristic that derived from the idea of Atkinson (1994). It is able to solve the problem time efficiently, but is vulnerable to ending up in a non-optimal solution. To circumvent these types of situations, we develop a hyper-heuristic framework based on the "pilot method" of Duin and Voß (1999). It supports greedy algorithms in avoiding local optimum traps. Duin and Voß (1999) and Voß et al. (2005) show that the pilot method is suitable for solving highly combinatorial problems (like the PBA), and that it performs competitively compared to well-known meta-heuristics. By only looking forward, the method iteratively weights all options before choosing the most promising. Further notation is delineated in Table 5.3.

Table 5.3: Expanded notation for the pilot method

a_0	Most promising element $u(a_0) \geq u(a)\ \forall a \in A$
A	Set of all possible choices a, so-called pilots
H	Subheuristic applied to assign remaining pilots $a \in A \setminus S_a$ (e.g., greedy heuristic)
N	Number of partial solutions considered at each iteration
S_a	Partial solution $S_a = a \cup X$
$u(a)$	Predetermined utility function $u : A \rightarrow \mathbb{R}$
X	Master solution, is iteratively created by adding the most promising element of an iteration $X = X \cup a_0$

General Algorithm An initial empty master solution $X = \emptyset$ is iteratively supplemented by an element $a \in A$, whereas A represents the set of all possible choices, so-called pilots. Based on the master solution X, a number of partial solutions N are generated by randomly drawing a pilot ($S_a = a \cup X$). Each partial solution is completed by the remaining pilots $a \in A \setminus S_a$ by applying a subheuristic H. Each solution can be evaluated using a predetermined utility function $u : A \rightarrow \mathbb{R}$. Let a_0 be the most promising element $u(a_0) \geq u(a)\ \forall a \in A$. The pilot a_0 gets included in the master solution $X = X \cup a_0$ and excluded from the remaining choices $A = A \setminus a_0$. Then the algorithm loops to create the next partial solution $S_a = a \cup X$ until a stop criterion is met (e.g., set of pilots is empty $A = \emptyset$, limitation of iterations). In our case, the utility is the total utility of the objective function of Equation (5.1), i.e., $u(a) = U$.

To speed up the computations we limit the solution space by only considering the set of relevant beds $\overline{B}$ and patients $\overline{P}$. The relevant beds considered include only those beds $b, b \in \overline{B}$ that are scheduled to be vacated within the planning horizon T. This means that beds that are already occupied by patients who have an estimated LOS exceeding the planning horizon are not included ($\overline{B} \subseteq B$). Likewise, only those patients $p, p \in \overline{P}, \overline{P} \subseteq P$ who are not yet occupying a bed b within their designated ward space and who require a bed at some point in time within the planning horizon T are considered. In particular, this includes patients who have just arrived, patients who are already waiting in the overflow area, as well as future elective patients already scheduled and anticipated future emergency patients, at some point within the planning horizon T. Limiting the sets for patients and beds is possible, as non-medical room transfers are not allowed. Algorithm 5.1 demonstrates the pilot method tailored to the PBA problem.

Subheuristic The subheuristic applied is based on the GLA heuristic developed by Schäfer et al. (2019). It sequentially calculates the potential added utility value with Equation (5.1) of each possible patient bed combination and also considers at this stage the constraints in Equations (5.2) to (5.19). Finally, it executes the most promising assignment. The additional notation to describe the subheuristic is shown in Table 5.4.

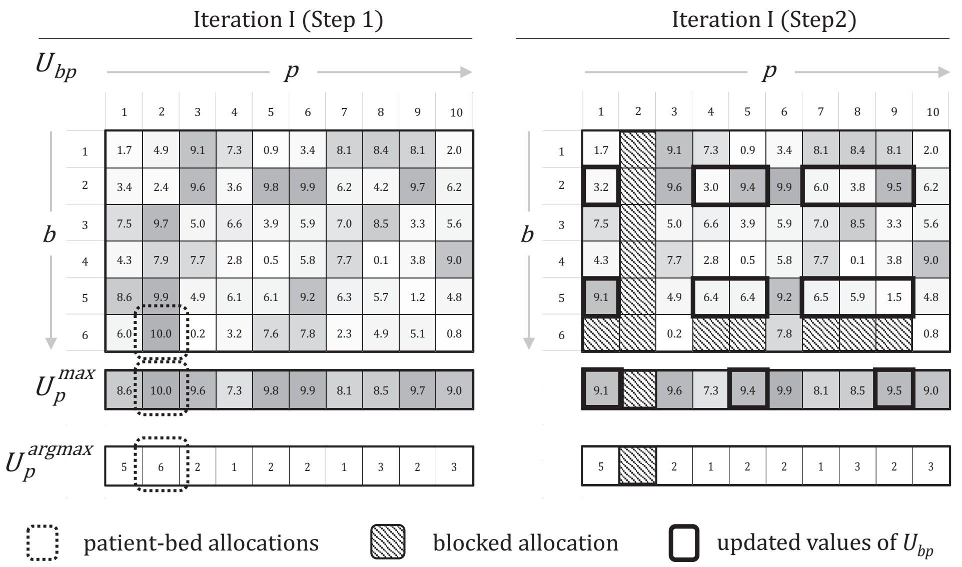
Algorithm 5.1 Pilot method for PBA

```
Require: $\overline{P}$, $\overline{B}$, $N$
Ensure: patient bed assignments $x_{bp}$
$x_{bp} \leftarrow \varnothing$
$A \leftarrow$ generatePossiblePatientBedAssignments($\overline{P}, \overline{B}$)
while ($|A| \neq 0$) do
    for $i \leftarrow 1, N$ do
        $a[i] \leftarrow$ random($A$)
        $pilot \leftarrow x_{bp} \cup a[i]$
        $\overline{B}'[i], \overline{P}'[i] \leftarrow$ updatePatientsAndBeds($\overline{B}, \overline{P}, a[i]$
        $pilotSolution[i] \leftarrow$ Subheuristic($\overline{B}'[i], \overline{P}'[i]$)
        $fitness[i] \leftarrow$ calculateFitness($pilotSolution[i]$)
    end for
    $j \leftarrow$ argmax($fitness$)
    $a_0 \leftarrow a[j]$
    $x_{bp} \leftarrow x_{bp} \cup a_0$
    $\overline{B} \leftarrow \overline{B}'[j]$
    $\overline{P} \leftarrow \overline{P}'[j]$
    $A \leftarrow$ updatePossiblePatientBedAssignments($A, a_0$)
end while
printPatientBedAssignments($x_{bp}$)
```

Table 5.4: Further notation for the subheuristic for PBA

U_{bp}	Partial utility that an assignment of patient $p, p \in \overline{P}$ to bed $b, b \in \overline{B}$ may add to the total utility U
U_p^{argmax}	Index value of the bed b that adds the maximum partial utility $\max(U_{bp})$ to the total utility U when patient p, $p \in \overline{P}$ will be allocated to this bed b
$U_p^{\max}$	Maximum partial utility that an assignment of patient p may add to the total utility U, $p \in \overline{P}$

Figure 5.3 illustrates the first iteration of the GLA heuristic. During an initialization process x_{bp} is set to zero and the utility matrix U_{bp} is calculated for all $p \in \overline{P}$ and $b \in \overline{B}$. The utility matrix U_{bp} represents partial utilities that can be added to the total utility function U (Equation (5.1)) by realizing a patient p to bed b assignment.



Iteration I (Step 1)

U_{bp}	1	2	3	4	5	6	7	8	9	10
1	1.7	4.9	9.1	7.3	0.9	3.4	8.1	8.4	8.1	2.0
2	3.4	2.4	9.6	3.6	9.8	9.9	6.2	4.2	9.7	6.2
3	7.5	9.7	5.0	6.6	3.9	5.9	7.0	8.5	3.3	5.6
4	4.3	7.9	7.7	2.8	0.5	5.8	7.7	0.1	3.8	9.0
5	8.6	9.9	4.9	6.1	6.1	9.2	6.3	5.7	1.2	4.8
6	6.0	10.0	0.2	3.2	7.6	7.8	2.3	4.9	5.1	0.8
U_p^{max}	8.6	10.0	9.6	7.3	9.8	9.9	8.1	8.5	9.7	9.0
U_p^{argmax}	5	6	2	1	2	2	1	3	2	3

Iteration I (Step2)

U_{bp}	1	2	3	4	5	6	7	8	9	10
1	1.7	▨	9.1	7.3	0.9	3.4	8.1	8.4	8.1	2.0
2	3.2	▨	9.6	3.0	9.4	9.9	6.0	3.8	9.5	6.2
3	7.5	▨	5.0	6.6	3.9	5.9	7.0	8.5	3.3	5.6
4	4.3	▨	7.7	2.8	0.5	5.8	7.7	0.1	3.8	9.0
5	9.1	▨	4.9	6.4	6.4	9.2	6.5	5.9	1.5	4.8
6	▨	▨	0.2	▨	▨	7.8	▨	▨	▨	0.8
U_p^{max}	9.1	▨	9.6	7.3	9.4	9.9	8.1	8.5	9.5	9.0
U_p^{argmax}	5	▨	2	1	2	2	1	3	2	3

Figure 5.3: Example for the GLA heuristic showing the steps of one iteration

If a bed b is not available at any time of the planned stay for the specific patient p, the partial utility value U_{bp} is set to zero. In Iteration I (Step 1), the most promising combination U_{bp} (highest utility value) is chosen, i.e., x_{bp} is set to 1 for patient 2 and bed 6 ($x_{62} = 1$). In the example, patient $p = 2$ is assigned to bed $b = 6$ as this yields the highest partial utility $U_p^{\max}$, with $U_p^{\max} = \max(U_{bp}), \forall b \in \overline{B}, \forall p \in \overline{P}$. To accelerate the process of finding the highest value during the iterations, two auxiliary variables are used to indicate the uppermost potential utility of a patient's assignment ($U_p^{\max}$) and the corresponding bed (U_p^{argmax}). This reduces the amount of values that need to be compared from $|\overline{P}| \times |\overline{B}|$ to $|\overline{P}|$ in each step.

The initial allocation of $x_{62} = 1$ has an effect on a series of potential allocation combinations x_{bp} of the remaining patients P and beds B. Subsequently, in Iteration I (Step 2), potential patient bed utilities U_{bp} that have been affected by a previous PBA in Step 1 get updated (black boxes in Figure 5.3). If necessary, $U_p^{\max}$ and U_p^{argmax} are redetermined. The following Iteration II also starts with the assignment of the most beneficial PBA. It will assign patient $p = 6$ to bed $b = 2$, as this has the highest utility U_{bp}, as can be seen on the right of Figure 5.3. In Iteration II, Step 2, the utilities of all remaining patient bed combinations will be updated. This will be continued until all patients are assigned. Algorithm 5.2 represents the iterative, procedural program flow.

Algorithm 5.2 Subheuristic: GLA heuristic for PBA

```
Require: $\overline{P}$, $\overline{B}$
Ensure: patient-bed assignments $x_{bp}$
$U_{bp} \leftarrow$ calculatePatientBedMatrix($\overline{P}$, $\overline{B}$)
$U_p^{\max} \leftarrow \max(U_{bp})$
$U_p^{\text{argmax}} \leftarrow \text{argmax}(U_{bp})$
while ($\max(U_{bp}) \neq 0$) do
    $p \leftarrow \text{argmax}(U_p^{\max})$
    $b \leftarrow U_p^{\max}[p]$
    $x_{bp} \leftarrow 1$
    $U_{bp} \leftarrow$ updatePatientBedMatrix($p$, $b$, $U_{bp}$, $\overline{P}$, $\overline{B}$)
end while
printPatientBedAssignments($x_{bp}$)
```

Applied Policies for Patient Bed Assignment To speed up the algorithm and tailor it to the PBA, different policies have been implemented and tested. First, at the start of each new pilot iteration the *filter policy* selects only a determined number of promising pilots. The vector $\text{argmax}(U_p^{\max})$ (see Algorithm 5.2) is used for this, the calculation taking place anyway to subsequently complete the partial solutions. Here, only those pilots with high expected additional utility values are considered. Second, the *drop policy* is applied, which executes the subheuristic H for only a predetermined fraction of the remaining options $a \in A \setminus X$. This can be guaranteed by only considering patients in the subheuristic who arrive within a certain period (shorter than the planning horizon). Finally, we also restricted the *evaluation depth*, i.e., only a subset of pilots $a \subseteq A$ are allocated by the pilot method. The remaining ones $a \in A \setminus X$ get assigned by the subheuristic H. The efficiency and applicability of the different policies are investigated in the numerical studies.

5.4 Numerical study

This section presents numerical studies. We draw upon real-life hospital data from a joint project with a large German hospital. First, we start in subsection 5.4.1 by presenting the machine learning approach used to anticipate emergency inpatient arrivals. Second, in subsection 5.4.2 we show the performance of the hyper-heuristic we have developed. Finally, in subsection 5.4.3 we analyze the impact of both the enhanced emergency inpatient arrival forecasting approach as well as the improved hyper-heuristic on the overall solution. All computational steps were carried out using Python 3.8 and R 3.6.

5.4.1 Anticipating emergency inpatient arrivals

In order to analyze potential influences on emergency patient arrivals, we have gathered metadata on various distinct features that were publicly available and which we suspected of having an impact on the emergency arrivals. These features relate to time and dates, weather data, important local and regional events, as well as historical and current occupancy levels (see Table 5.5). We then used this data in a machine learning approach to anticipate emergency inpatient arrivals based on a selection of the most significant features. The training data used spans across a time period of 2 years from 2014 to 2015, while our test and validation data is taken from 2016.

Identification of feature importance by medical department

We completed two steps to identify important features: (1) multicollinearity test to distinguish highly correlated features, and (2) identification of the explanatory power of remaining features with respect to emergency patient

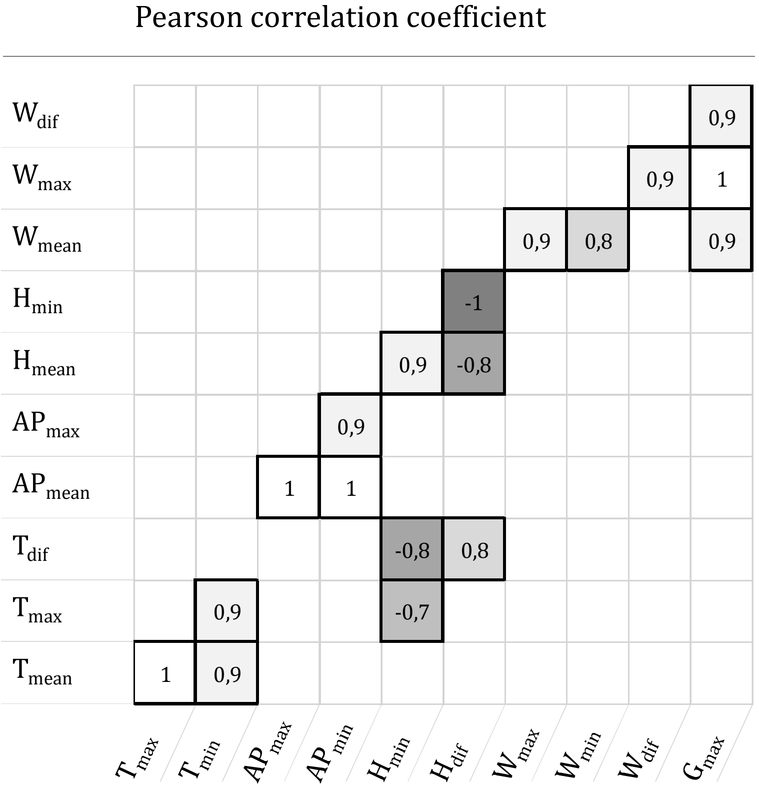
Table 5.5: Overview of factors and properties assessed regarding correlation with emergency inpatient arrivals

Factor	Feature
Time and Date	Weekday (WD_{Mon}, WD_{Tue}, ...) Season (Q1, Q2, Q3, Q4) School holidays ($\text{Hol}_{\text{School}}$) Bank holidays (Holiday) Post holiday weekday ($\text{WD}_{\text{postholiday}}$)
Weather	Temperature (T_{mean}, T_{min}, T_{max}, T_{dif}) Air Pressure (AP_{mean}, AP_{min}, AP_{max}, AP_{dif}) Humidity (H_{mean}, H_{min}, H_{max}, H_{dif}) Wind (W_{mean}, W_{min}, W_{max}, W_{dif}, G_{max}) Precipitation (Rain, Snow, Hail) Snow coverage (S_{cov}) Storm
Local and Regional Events	Fairs (County Fairs, Sport events)
Current Occupancy	Admissions of previous day (PrevAdmin)

arrivals. As each medical department has its own drivers, we undertake this investigation individually by department.

(1) In a first step, to avoid multicollinearity issues (see e.g., Guyon and Elisseeff (2003)), we determine the Pearson correlation coefficients (PCC) of each potential pairing of features listed in Table 5.5. Figure 5.4 gives an overview of all problematic pairings, i.e., all pairings wherein |PCC| >= 0.7. A simple example of this would be that the maximum temperature T_{max} strongly correlates with the minimum temperature T_{min}, e.g., minimum and maximum temperatures for any given day during summer time are typically higher than during winter time. Only one variable would be used for each of these pairings.

(2) In a second step, the remaining features have to be tested to determine their explanatory power regarding the number of patient arrivals on a given day. This is important for two reasons. First, simply looking at the direct



Pearson correlation coefficient

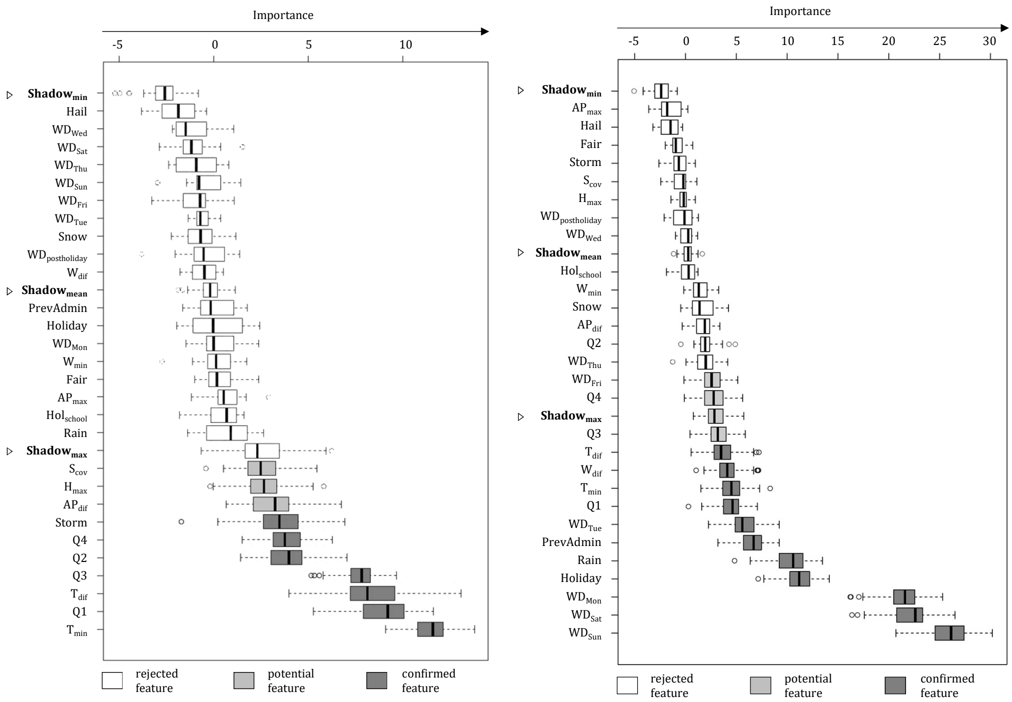
	T_{max}	T_{min}	AP_{max}	AP_{min}	H_{min}	H_{dif}	W_{max}	W_{min}	W_{dif}	G_{max}
W_{dif}										0,9
W_{max}									0,9	1
W_{mean}							0,9	0,8		0,9
H_{min}						-1				
H_{mean}					0,9	-0,8				
AP_{max}				0,9						
AP_{mean}			1	1						
T_{dif}					-0,8	0,8				
T_{max}		0,9			-0,7					
T_{mean}	1	0,9								

Figure 5.4: Measure of linear correlations between selected parameters

correlation between a given feature and the number of emergency arrivals in the test data can be misleading as this overlooks any potential effects that certain properties only have in combination (Guyon and Elisseeff, 2003). Second, machine learning algorithms tend to show a decrease in accuracy when the number of features used is significantly higher than optimal (see for example Kohavi and John (1997)). To this end, we make use of the "Boruta" package developed by Kursa and Rudnicki (2010). It consists of a feature selection algorithm based on the "random forest" classification method (Breiman, 2001). Its aim is to rank a set of features according to their respective predictive power regarding a specific classification variable, e.g., the number of emergency patient arrivals per day. This ranking is performed according to the individual "importance" of each feature, which is based on the average and standard deviation of the loss of accuracy of classification caused by the random permutation of attribute values between objects. A key idea here is to introduce so-called “shadow variables”, i.e., additional random variables, which are then included in the set of existing features. By adding randomness to the data set and collecting results from the ensemble of randomized samples, it is possible to reduce the misleading impact of random fluctuations and correlations.

The above-described feature selection process is undergone individually for every medical department, that has emergency arrivals. To give an example of this feature selection we present detailed results for two different departments, namely trauma surgery and gastroenterology, as can be seen in Figures 5.5a and 5.5b, respectively. For trauma surgery, the number of emergency inpatient arrivals is clearly correlated with the seasons (Q1 to Q4), with low temperatures ($\mathrm{T_{min}}$), as well as with the magnitude of intra-day temperature changes ($\mathrm{T_{dif}}$). Naturally, any feature that correlates with the number of emergency inpatient arrivals, in both the training data set and the test data set, can prove useful when anticipating such arrivals. However, the causality behind this correlation may only be guessed. In the case of emergency patients having had an accident that requires trauma surgery, it seems plausible that sudden drops of temperature, which lead to black ice on roads and sidewalks, or typical recreational activities pursued in winter (Q1), e.g., skiing, are responsible for this effect.

For the gastroenterology department, however, the picture looks quite different. Here, holidays, weekends and Mondays each exhibit a high explanatory correlation with regard to incoming emergency patients, whereas the temperature has a considerably lower influence when compared to the trauma surgery department. This could be due to a couple of different reasons. For instance, doctors and nursing staff we interviewed have reported that many gastroenterological illnesses often initially present with non-specific abdominal pain symptoms, which then intensify over the course of several days. This means that in comparison with a broken hip, for example, there



(a) Trauma surgery department **(b)** Gastroenterology department

Figure 5.5: Selected outcomes after application of the Boruta package

is no immediate need to get to a hospital, such that patients could opt to stay home on weekends. An alternative explanation could be that resident doctors' offices are typically closed on weekends and patients who are not yet aware of the severity of their illness will usually wait until the next workday to see their family doctor who might then immediately refer them to a hospital for further diagnosis and treatment.

Applying machine learning to estimate emergency patients

Estimating the number of future emergency patient admissions is inherently a regression problem. We therefore first applied (1) regression-based methods using the metadata described in Table (5.5). In addition, in a

(2) second step we applied a multilayer artificial neural network to also account for nonlinear dependencies. We used regularization methods in both approaches to avoid overfitting. Finally, (3) we used the test data to evaluate the generalization abilities of our trained models.

(1) Regression-based methods Ridge regression (RR) uses l_2-regularization (Hoerl and Kennard, 1970), whereas LASSO (LR) uses l_1-regularization (Tibshirani, 1996). l_2-regularization accounts for correlations between the input features, while l_1-regularization favors sparse solutions. Elastic Net (EN) is a regression-based method that combines l_1 and l_2 regularization (Zou and Hastie, 2005). Another class of regression models is Group-LASSO (GL), which allows individual features to be combined into groups (Yuan and Lin, 2006). All features of a group are penalized together, leading to whole groups being considered or neglected. We used 10-fold cross-validation to tune the hyperparameter λ for each approach, which controls the strength of the regularization. In the case of EN we iterated in a loop between 0 and 1 in 0.025 increments around the 10-fold cross validation structure to determine λ_2 as the second hyperparameter.

(2) Artificial neural network We used artificial neural networks (ANN) (Goodfellow et al., 2016; LeCun et al., 2015) to account for non-linear dependencies. An example architecture of an ANN is illustrated in Figure (5.6). We have evaluated several typologies of ANNs by varying the number of hidden layers between one to five. The best results have been achieved by applying a “32:16:8:4:2” network (the numbers are the number of neurons per hidden layer; hidden layers are separated by colons), the rectified linear unit (ReLu) as activation function, l_1 and l_2 regularization and the mean-squared error (MSE) loss function as well as the optimizer RMSprop. To avoid overfitting we have investigated the learning curve of training and validation loss. For tuning hyperparameters l_1 and l_2 we used a grid search algorithm.

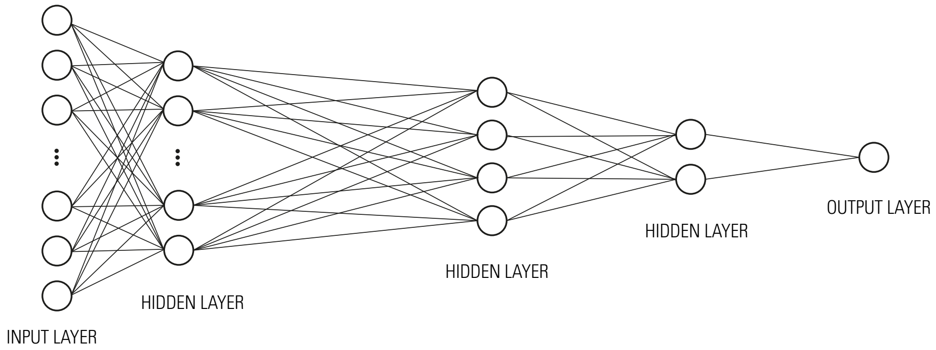


Figure 5.6: Example for the structure of a neural network including tree hidden layers

(3) Evaluation of performance on test data We applied the learned models to the test data from four departments at our case hospital that have a significant number of emergency patients. For example, orthopedics has almost no emergency patients. Table 5.6 summarizes the results and shows the root mean square error (RMSE), the machine learning model used that achieved the best performance, as well as the improvement achieved in comparison to historical averages. The historical averages serve as a baseline approach, and this is denoted as "Approach 1". This is compared with our above-described machine learning approach (denoted as "Approach 2 (ML)"). Table 5.6 shows that the machine learning approach outperforms "Approach 1". The ML approach leads to improvements of up to 17%, depending on the department, compared to the basic historical averages. The improvement gap of 12% between the departments may be explained by the department-dependent impact of the features. For instance, Figure 5.5b shows that the Gastroenterology Department incorporates three features that have a median importance scale greater than 20, whereas Figure 5.5a shows that the Trauma Surgery Department only barely surpasses the median importance of 10 in one feature. This is also in line with the Boruta analysis, where the importance of the features of Department 3 is rated lowest by far.

Table 5.6: Anticipation of emergency inpatient arrivals using machine learning

	Approach 1	Approach 2 (ML)					Max.	
		RR	LR	EN	GL	ANN	Improvement	
Department	RMSE	RMSE	RMSE	RMSE	RMSE	RMSE	[%]	Type
Department 1	4.888	4.331	4.309	4.183	4.103	**4.060**	16.939	ANN
Department 2	4.108	3.892	3.847	3.848	**3.675**	3.835	10.540	GL
Department 3	2.888	2.887	2.824	2.804	**2.743**	2.778	5.021	GL
Department 4	3.535	3.126	3.099	3.097	**3.067**	3.192	13.239	GL

5.4.2 Performance of the hyper-heuristic

In order to assess the solution quality of the hyper-heuristic proposed in this paper, we drew upon nine data sets. The available real data of one department cluster consisting of department 1 and department 2 (see also 5.4.1) based on actual patient movements between January 2016 and September 2016 will be considered. The department cluster consists of six wards with 24 beds each. Each data set is composed of 28 consecutive days and comprises an average of 648 unique patients. On average, 40% of patients are men and 60% women with an average age of 70 years and a length of stay of 6 days. All data sets reveal high ratios of emergency patients, e.g., up to 90%. We set the parameters in alignment with currently applied weights in our case hospital: $\alpha = 1$, $\beta = 0.1$, $\gamma, \delta = 2$, $q = 0.01$. Furthermore, the weighting factor Ξ_p was set to three distinct values depending on the patient type. Notably, these consist of $\Xi^{el} = 10$ for elective patients, $\Xi^{em} = 9$ for current emergency arrivals, and $\Xi^{an} = 4$ for anticipated emergency arrivals. Here, elective patients are preferred to current emergency patients, and these in turn are given preference vs. expected future emergency arrivals.

We have adjusted the existing data by eliminating all uncertainty factors for the sole purpose of monitoring the performance of the heuristics applied. Accordingly, emergency patients treated like elective patients and their exact admission are known in advance. Both patient types are no longer subject to LOS updates due to precisely known discharge times. Furthermore, patient no-shows are neglected. This means that the data sets considered are no longer affected by stochastic variations and are assumed to be deterministic.

Application to single problem instances We first assessed the performance of our hyper-heuristic for single problem instances. Using such a static version is a usual benchmark approach (see literature review in Section 5.2.3 above and for example in Bilgin et al. (2012); Guido et al. (2018); Dorgham et al. (2019); Ceschia and Schaerf (2011)). This approach excludes parameter-dependent (e.g., planning horizon, time-dependent relevance) performance differences caused by time series analysis. These parameters could lead to worse performance in the time series analysis and thus reduce the meaningfulness of hyper-heuristic performance despite better performance in all single problem instances. We tested several policies (see Section 5.3.2). In particular, we applied a *filter policy* with which we restricted the number of promising pilots to different predetermined amounts, which were determined based on their individual additional potential benefit to the utility function prior to an algorithm run-through. The best patient-bed assignments are drawn randomly from the five most promising patients. This was done to avoid unnecessary computational effort while at the same time ensuring that no potentially “lucrative” PBAs are overlooked. It should be noted that several potential PBAs of a single patient may have similar values and hence a wide variety of alternative promising PBAs exist. In addition, we applied a *drop policy* by limiting the application of the GLA subheuristic to only those patients that were known or anticipated to arrive within a certain number of days, which also leads to a significant reduction of computational time while retaining a high solution quality. Finally, we varied the *evaluation depth* by restricting the amount of subsequent PBAs obtained through the pilot method. To give an example, selecting only 10 pilots and a depth of 20 translates into applying the pilot method to determine the first 20 PBA, wherein for each of these 20 assignments the 10 most promising pilots will be chosen and evaluated using the GLA heuristic. Table 5.7 gives an overview of the solutions obtained. For each of the shown combinations of data set used, amount of promising pilots filtered (in lines), and evaluation depth (in columns), we have taken into account all single problem instances which emerged by executing the data sets. This results in around 2,000 single problem instances for each data set (i.e., around 288,000 in total), promising pilot and evaluation depth combinations. We did this in order to account for statistical distributions, which arise due to the inherent randomness associated with our implementation of the hybrid-heuristic.

The results obtained allow for drawing three main insights. First, by using the pilot method, it was possible to increase the solution quality in comparison to the GLA heuristic by up to 2.90% while achieving an average increase of 2.42% when considering 20 promising pilots combined with an evaluation depth of 20. This number can of course vary depending on the characteristics of the underlying patient clientele. It should be noted, however, that the effect observed is substantially the same across all nine data sets. Second, as is to be expected, increasing the evaluation depth as well as increasing the number of promising pilots both lead to an increase in solution quality. This is because it is more likely that better solutions will be found when broadening the search space as this increases the chance of finding solutions that are further away from standard GLA heuristic solutions. Here, it should be noted that, overall, the effect of increasing the evaluation depth has a higher impact on solution quality than increasing the number of promising pilots considered. A reason for this behavior could be seen in that even when using a low number of promising pilots considered, the pilots chosen exhibit the highest additional benefit to the overall utility function, respectively, which makes the underlying PBA more likely to be part of a good solution. Third, depending on the situation at hand, the acquired gain in solution quality due to a broader search space goes hand in hand with higher computational effort, which can be an important factor when requiring real-time PBAs in actual hospital situations. Roughly speaking, the total computation time for a single problem instance can

Table 5.7: Solution quality of Pilot method compared to GLA heuristic for single problem instances

DS[1]1	**depth**					DS 2	**depth**				
pilots[2]	5	10	15	20	Avrg	**pilots**	5	10	15	20	Avrg
5	0.66%	1.13%	1.49%	1.73%	1.25%	5	0.52%	0.99%	1.33%	1.49%	1.08%
10	0.76%	1.27%	1.60%	1.91%	1.38%	10	0.61%	1.12%	1.45%	1.62%	1.20%
15	0.79%	1.30%	1.59%	1.94%	1.40%	15	0.65%	1.17%	1.46%	1.63%	1.22%
20	0.83%	1.31%	1.66%	1.93%	1.43%	20	0.64%	1.21%	1.41%	1.61%	1.21%
Avrg	0.76%	1.25%	1.59%	1.88%	-	Avrg	0.60%	1.12%	1.41%	1.59%	-
DS 3	**depth**					DS 4	**depth**				
pilots	5	10	15	20	Avrg	**pilots**	5	10	15	20	Avrg
5	0.55%	0.98%	1.49%	2.05%	1.02%	5	0.65%	1.31%	1.98%	2.39%	1.56%
10	0.60%	1.06%	1.6%	2.27%	1.10%	10	0.78%	1.64%	2.15%	2.57%	1.76%
15	0.66%	1.18%	1.79%	2.20%	1.16%	15	0.81%	1.62%	2.35%	2.78%	1.85%
20	0.68%	1.28%	1.92%	2.38%	1.24%	20	0.79%	1.77%	2.28%	2.67%	1.85%
Avrg	0.62%	1.12%	1.70%	2.22%	-	Avrg	0.76%	1.58%	2.19%	2.60%	-
DS 5	**depth**					DS 6	**depth**				
pilots	5	10	15	20	Avrg	**pilots**	5	10	15	20	Avrg
5	0.68%	1.12%	1.51%	1.83%	1.28%	5	0.81%	1.51%	2.27%	2.54%	1.75%
10	0.78%	1.21%	1.64%	2.04%	1.41%	10	0.91%	1.73%	2.44%	2.85%	1.94%
15	0.79%	1.26%	1.78%	1.98%	1.44%	15	0.96%	1.82%	2.47%	2.79%	1.97%
20	0.84%	1.33%	1.80%	2.03%	1.49%	20	0.95%	1.91%	2.51%	2.90%	2.02%
Avrg	0.77%	1.23%	1.68%	1.97%	-	Avrg	0.91%	1.74%	2.42%	2.77%	-
DS 7	**depth**					DS 8	**depth**				
pilots	5	10	15	20	Avrg	**pilots**	5	10	15	20	Avrg
5	0.82%	1.43%	2.02%	2.40%	1.65%	5	0.95%	1.68%	2.22%	2.66%	1.86%
10	0.88%	1.64%	2.17%	2.62%	1.81%	10	1.06%	1.83%	2.43%	2.72%	1.99%
15	0.93%	1.76%	2.31%	2.70%	1.91%	15	1.11%	1.87%	2.51%	2.69%	2.03%
20	0.94%	1.77%	2.32%	2.79%	1.94%	20	1.12%	1.92%	2.52%	2.84%	2.08%
Avrg	0.89%	1.65%	2.21%	2.63%	-	Avrg	1.06%	1.82%	2.42%	2.73%	-
DS 9	**depth**					Total[3]	**depth**				
pilots	5	10	15	20	Avrg	**pilots**	5	10	15	20	Avrg
5	0.70%	1.19%	1.72%	2.33%	1.46%	5	0.70%	1.26%	1.78%	2.16%	1.48%
10	0.78%	1.43%	1.98%	2.55%	1.66%	10	0.80%	1.44%	1.94%	2.35%	1.63%
15	0.80%	1.47%	2.06%	2.52%	1.69%	15	0.83%	1.49%	2.03%	2.36%	1.68%
20	0.83%	1.52%	2.16%	2.63%	1.75%	20	0.84%	1.56%	2.07%	2.42%	1.72%
Avrg	0.78%	1.40%	1.98%	2.51%	-	Avrg	0.79%	1.44%	1.96%	2.32%	-

[1] data set used to extract problem instances

[2] amount of promising pilots filtered for further analysis

[3] total average over all analyzed problem instances

be estimated by adding up the total number of times the subheuristic has to run through all PBAs for a given single problem instance. To give an example, an evaluation depth of 10 combined with 10 selected pilots will add up to 100 applications of the subheuristic while an evaluation depth of 5 combined with 5 selected pilots will only require 25 run-throughs of the GLA heuristic, or 25% of the time. The runtime changes only proportional to the dimension of evaluation depth when multi-processing is applied. This roughly means that the runtime compared to the GLA heuristic is just multiplied by the evaluation depth. The GLA heuristic is typically solved in an average of less than one second for instances encompassing 124 beds.

Application to time series In addition to comparing the solution quality for single problem instances, we have undertaken analyses to compare the performances of both approaches over time. For this purpose, the data sets that have been cleared of uncertainties are also used. Furthermore, to investigate the scaling effect in relation to the department cluster size we divided the nine existing data sets with regard to the department cluster size stepwise by 24 beds from 24 to 120. To do this, the patients and beds are added depending on the division of the wards and their specific specialty. To test the hyper-heuristic approach developed, we use the top-performing settings from the single problem instance analyses (see Table 5.7), i.e., an evaluation depth of 20 combined with a selection of 20 promising pilots for each subsequent PBA.

The results of this analysis are presented in Table 5.8. Again, we have accounted for statistical effects of the stochastic search procedure by running the algorithm 20 times for each combination of data set and beds considered. Here, the results show an increase in total utility. The hyper-heuristic approach outperforms the GLA heuristic by 1.48% on average while achieving an increase of up to 3.86% for certain data sets. The utility increase of the hyper-heuristic vs. the GLA heuristic for the time series analyses in Table 5.8 is not as clearly predictable as for the single problem instance solution in Table 5.7. This is due to the settings of the hyper-heuristic (i.e., plan-

Table 5.8: Solution quality of Pilot method compared to GLA heuristic based using a time series analysis

Data		**24 beds**		**Data**		**48 beds**	
Set	Min	Avrg	Max	**Set**	Min	Avrg	Max
1	0.64%	1.62%	3.4%	1	1.54%	2.47%	3.58%
2	0.43%	0.75%	1.32%	2	-0.02%	1.69%	2.73%
3	-1.12%	-0.03%	0.68%	3	0.72%	0.99%	1.34%
4	1.08%	1.31%	1.54%	4	1.87%	3.18%	3.71%
5	0.14%	1.06%	2.60%	5	1.27%	2.06%	2.55%
6	1.89%	2.83%	3.48%	6	1.59%	1.81%	2.16%
7	1.66%	1.77%	1.96%	7	1.85%	2.56%	3.86%
8	0.82%	1.72%	2.44%	8	1.18%	2.87%	3.68%
9	-0.45%	0.46%	1.50%	9	1.21%	2.05%	2.89%
Avrg	0.57%	1.28%	2.10%	Avrg	1.25%	2.19%	2.95%
Data		**72 beds**		**Data**		**96 beds**	
Set	Min	Avrg	Max	**Set**	Min	Avrg	Max
1	0.79%	1.62%	2.39%	1	1.00%	1.69%	2.17%
2	1.32%	2.30%	3.18%	2	0.98%	1.55%	1.88%
3	1.47%	1.47%	1.47%	3	0.00%	0.23%	0.47%
4	0.67%	0.67%	0.67%	4	0.81%	1.17%	1.61%
5	1.75%	1.75%	1.75%	5	1.36%	1.65%	1.82%
6	0.68%	0.68%	0.68%	6	0.80%	1.68%	2.38%
7	2.25%	2.25%	2.25%	7	1.53%	1.71%	1.97%
8	1.52%	1.52%	1.52%	8	0.32%	0.84%	1.37%
9	1.02%	1.02%	1.02%	9	0.17%	0.58%	0.76%
Avrg	1.27%	1.47%	1.66%	Avrg	0.77%	1.23%	1.61%
Data		**120 beds**		**Data**		**Total**[1]	
Set	Min	Avrg	Max	**Set**	Min	Avrg	Max
1	1.30%	1.75%	2.11%	1	1.06%	1.83%	2.73%
2	0.88%	1.24%	1.57%	2	0.72%	1.51%	2.13%
3	1.03%	1.61%	2.06%	3	0.42%	0.85%	1.21%
4	-0.14%	0.56%	1.03%	4	0.86%	1.38%	1.71%
5	0.69%	0.91%	1.20%	5	1.04%	1.49%	1.98%
6	1.92%	2.30%	2.74%	6	1.38%	1.86%	2.29%
7	1.18%	1.48%	1.86%	7	1.69%	1.95%	2.38%
8	-0.05%	0.25%	0.52%	8	0.76%	1.44%	1.91%
9	0.76%	1.07%	1.42%	9	0.55%	1.04%	1.52%
Avrg	0.84%	1.24%	1.61%	Avrg	0.94%	1.48%	1.98%

[1] total average over all problem sizes

ning horizon, time-dependent relevance parameter Q_t). Furthermore, only patients within the planning horizon, that may overlap with the hospital stays of future elective patients (arrival exceeding planning horizon) are considered. In other words, even if the hyper-heuristic performs considerably better than the GLA heuristic for each single problem instance within the time series investigated, time-dependent parameter settings may eradicate the positive effect of the hyper-heuristic compared to the GLA heuristic for certain combinations of data sets and beds. This also explains some negative entries in the minimum values of Table 5.8. The hyper-heuristic outperformed the GLA heuristic in over 99% of the test instances.

5.4.3 Hyper-heuristic combined with enhanced emergency inpatient arrival forecasting

In this subsection, the impact of both the enhanced emergency inpatient arrival forecasting approach as well as the improved hyper-heuristic with regard to real data are analyzed. The nine data sets (6 wards with 24 beds each), including uncertainties, are used to do this. Each data set consists of around 2,000 unique events that take place over the course of 28 days. We denote the hyper-heuristic approach including the enhanced emergency patient admission data, which was achieved with machine learning, as *Hyper-Heuristic ML*. It is executed 20 times for all data sets and the average of all runs is reported. We apply two benchmarks:

- GLA Avg: The first is the GLA heuristic of Schäfer et al. (2019) where the arrivals of emergency patients have been estimated according to Approach 1 (see 5.4.1). This is exactly the approach in Schäfer et al. (2019). We normalize all values of the alternative approaches to this.
- GLA ML: The second is also based on the GLA heuristic, but the arrivals of emergency patients have also been estimated with machine learning.

Looking at the results of the analysis of the three methods in Table 5.9, the normalized values of the objective function give a first indication of the performance of our approach. It can be noted that the Hyper-Heuristic ML outperforms the GLA as well as the GLA ML approach in each data set. On average across all data sets the Hyper-Heuristic ML shows 1.4% better results than the GLA and beats the GLA ML by 0.4%. Even the minimum outcome of the Hyper-Heuristic ML for all data sets performs better than the GLA method. This makes the Hyper-Heuristic ML the most promising and reliable approach to solve the PBA problem.

Table 5.9: Solution quality of Hyper-heuristic ML compared to benchmarks using a time series analysis

Model	**Data Set**									**Total[1]**
	1	2	3	4	5	6	7	8	9	
GLA Avg	99.37%	102.13%	102.77%	107.8%	99.65%	98.32%	101.11%	95.05%	93.81%	100.00%
GLA ML	100.12%	102.22%	103.69%	107.83%	101.60%	100.39%	101.96%	96.27%	94.91%	101.00%
Hyper-heuristic ML	100.38%	102.74%	104.31%	108.25%	101.77%	100.46%	102.21%	97.12%	95.31%	101.40%

[1] total average over all data sets

5.5 Conclusion and further areas of research

Conclusion This paper develops and investigates improvements for the operational PBA. The model used has been developed in a joint project with a large German hospital covering all major disciplines and incorporates the objectives and constraints of the three main stakeholders, namely patients, doctors, and nursing staff. It integrates the planning of current emergency and elective patient arrivals, future elective patient arrivals, as well as anticipated future emergency patient arrivals. Two important aspects were tackled and improved in this paper.

- To tackle the uncertainty of emergency patient admissions, we applied machine learning techniques to estimate these more precisely. To this end, we used historic emergency inpatient data as well as metadata relating to time, date, weather forecasts, and local and regional events. We are the first to investigate and make use of the correlation of several external factors, such as weather data, to better anticipate emergency inpatient admissions.
- To enhance the performance of the solution approach we integrate the GLA heuristic into the Pilot method which consists of a hyper-heuristic framework.

Our numerical results have shown that machine learning approaches can outperform historical average approaches by up to 17% when it comes to predicting emergency inpatient arrivals. The underlying drivers for emergency inpatient arrivals differ strongly between departments due to the associated patient clientele, e.g., Trauma Surgery shows a higher dependency on weather data than Gastroenterology, which in turn is more strongly correlated with times and dates. Compared to the GLA heuristic, the hyper-heuristic developed can improve performance by up to 3% for single problem instances and up to 4% in a time series analysis. With respect to real data, the hyper-heuristic approach combined with sophisticated prediction of future emergency patient admissions by machine learning outperforms the GLA heuristic in a time series analysis by up to 2.2%.

Future areas of research Various opportunities exist for further research. For the problem shown, the existing solution methods can be further developed and different approaches can be pursued. The focus may be on enhanced anticipation of the input parameters, improvement of the heuristic methods or development of an optimal solution method. The estimate of input parameters focuses on both emergency and elective patients. Information on the progress the patient's recovery is making (e.g., LOS as well as type and probability of complications) can be anticipated for both patient groups. The approximation of time-related arrivals and patient characteristics (e.g., gender, age, and disease) is especially in focus for emergency patients, while no-show rates are interesting for elective patients. In the development of heuristics, the focus can be on runtime-related aspects, solution quality or the proportion of both by implementing and testing alternative approaches (e.g., meta-heuristics, matheuristics, exact approaches). Another topic of research interest is to integrate upstream and/or downstream processes in the decision model, such as admission scheduling of elective patients, operating room scheduling, bed transport services or staff rostering (cf. e.g., van Oostrum et al. (2008); Beaudry et al. (2010); Rachuba and Werners (2014); Aringhieri et al. (2015); Erhard et al. (2018); Thielen (2018); Séguin et al. (2019)). This integration makes it possible to obtain information about conflicts of interests of individual problems. In order to maximize profit, operating rooms should usually be booked to full capacity, although the hospital may not have suitable beds available for patients who have had surgery. Furthermore, the underlying mechanics of the PBA decision model are not limited to hospital settings alone. Further investigation could be made into identifying problem settings that have a similar scope. To give an example, the student-room assignment problem in hostels (Alfred and Yu, 2020) could potentially yield further areas of application.

6 Conclusion and outlook

Conclusion This doctoral thesis deals with the application of operations research methods in the field of service operations management. Two specific problem settings are addressed in cooperation with partners from the retail and healthcare sector.

In retail, together with one of Germany's largest retailers the assortment and shelf-space optimization is examined for two-dimensional shelves or tilted shelves, respectively. The model takes into account natural demand effects, namely stochastic, substitution and space elasticity. A heuristic has been developed due to the NP-hardness of the mathematical problem. This enables retailers to solve the defined two-dimensional shelf-space problem efficiently and makes it suitable for daily application. It supports retailers in designing a planogram for two-dimensional shelves by calculating optimal assortments and shelf quantities as well as the adjacently rectangular arrangement of each item's facings. In a case study driven on retailer's data, it is shown that profits may increase by up to 15% compared to the current assortment. Additionally, the study figured out that by neglecting natural demand effects retailers suffer from up to 80% lower profits.

To the best of our knowledge, in the previous literature, the demand effects (i.e., stochastic, substitution and space elasticity) in combination with the assortment decision have only been examined on the basis of regular (i.e. one-dimensional) shelves. With regard to two-dimensional shelves, only shelf-space optimization models that account for deterministic demand have been investigated so far. Accordingly, all other demand effects and assortment decisions are neglected.

In healthcare, jointly with a large German hospital, the patient-bed assignment problem is revisited from a new angle. Therefore, in contrast to previous literature the key stakeholders of the bed management process are incorporated, namely patients, doctors, and nursing staff. The model developed cater to all major disciplines. A greedy look-ahead (GLA) heuristic has been developed to be suitable and applicable for daily use as an efficient and quick support system in large hospitals. On the basis of real hospital data the GLA heuristic improved the objectives of all stakeholders, e.g., the time spent in overflow was reduced by 96%. Furthermore, to better anticipate emergency patient admissions machine learning techniques are used. These can outperform historical average approaches by up to 17% when taking into account historic emergency inpatient data as well as metadata relating to time, date, weather, local and regional events.

In contrast to the existing literature on the patient admission scheduling problem, we have redefined the problem as patient-bed assignment problem on basis of the real situation in a hospital. This cooperation enabled us to be the first researchers to study the problem on the basis of real data. Furthermore, we are the first to have investigated and demonstrated a correlation between weather data and emergency patient admissions.

Future areas of research As demonstrated by the examples of retailing and healthcare, there are considerable potentials for improvement in the field of service operations management. On the one hand, in most practice cases no or only primitive methods are used for process optimization. Therefore, simple methodologies may lead to sufficient improvements. On the other hand, the continuous further development of computational performance, as well as practice-oriented operations research methods, enable researchers to carry out more and more complex decision models. Exaggeratedly said, time-critical decisions for which nowadays a heuristic is still needed can soon be solved to optimality.

The rising complexity of a decision model usually goes hand in hand with a more accurate representation of the real model, for example, the integration of upstream or/and downstream processes. In case of the two-dimensional shelf-space optimization model the integration of shelf refilling, order management, inventory accuracy, backroom inventory, and delivery frequency is conceivable. Regarding the patient-bed assignment problem it may be the incorporation with admission, scheduling of elective patients, operating room scheduling, bed transport services or staff rostering.

Bibliography

Aastrup, J., Kotzab, H., 2009. Analyzing out-of-stock in independent grocery stores: an empirical study. International Journal of Retail & Distribution Management 37 (9), 765–789.

Afilal, M., Yalaoui, F., Dugardin, F., Amodeo, L., Laplanche, D., Blua, P., 2016. Forecasting the emergency department patients flow. Journal of Medical Systems 40 (7), 175.

Agrawal, N., Smith, S., 1996. Estimating negative binomial demand for retail inventory management with unobservable lost sales. Naval Research Logistics 43 (6), 839–861.

Alfred, R., Yu, H. F., 2020. Automated scheduling of hostel room allocation using genetic algorithm. In: Sharma, N., Chakrabarti, A., Balas, V. E. (Eds.), Data Management, Analytics and Innovation. Springer Singapore, Singapore, pp. 151–160.

Aringhieri, R., Landa, P., Soriano, P., Tànfani, E., Testi, A., 2015. A two level metaheuristic for the operating room scheduling and assignment problem. Computers & Operations Research 54, 21 – 34.

Atkinson, J. B., 1994. A greedy look-ahead heuristic for combinatorial optimization: An application to vehicle scheduling with time windows. The Journal of the Operational Research Society 45 (6), 673–684.

Bastos, L. S., Marchesi, J. F., Hamacher, S., Fleck, J. L., 2019. A mixed integer programming approach to the patient admission scheduling problem. European Journal of Operational Research 273 (3), 831 – 840.

Beasley, J., 2004. A population heuristic for constrained two-dimensional non-guillotine cutting. European Journal of Operational Research 156 (3), 601 – 627.

Beaudry, A., Laporte, G., Melo, T., Nickel, S., Jan 2010. Dynamic transportation of patients in hospitals. OR Spectrum 32 (1), 77–107.

Bekker, R., Koole, G., Roubos, D., 2016. Flexible bed allocations for hospital wards. Health Care Management Science 20 (4), 453–466.

Beliën, J., Demeulemeester, E., 2007. Building cyclic master surgery schedules with leveled resulting bed occupancy. European Journal of Operational Research 176 (2), 1185–1204.

Bell, D., 1976. The Coming Of Post-Industrial Society -, reissue Edition. Basic Books, United States.

Berlyne, D. E., 1958. The influence of complexity and novelty in visual figures on orienting responses. Journal of Experimental Psychology 55 (3), 289.

Bianchi-Aguiar, T., Carravilla, M. A., Oliveira, J. F., 2015. Replicating shelf space allocation solutions across retail stores. Working Paper University Porto, 1–24.

Bianchi-Aguiar, T., Hübner, A., Carravilla, M. A., Oliveira, J. F., 2019. Retail shelf space planning problems: A comprehensive review and classification framework. Working Paper University Porto, 1–30.

Bianchi-Aguiar, T., Silva, E., Guimaraes, L., Carravilla, M. A., Oliveira, J. F., Amaral, J. G., Liz, J., Lapela, S., 2016. Using analytics to enhance a food retailer's shelf-space management. Interfaces 46 (5), 424–444.

Bilgin, B., Demeester, P., Misir, M., De Vancroonenburg, W., Vanden Berghe, G., 2012. One hyper-heuristic approach to two timetabling problems in health care. Journal of Heuristics 18 (3), 401–434.

Bordoloi, S., Fitzsimmons, J., Fitzsimmons, M., 2018. Service Management - Operations, Strategy, Information Technology, 9th Edition. McGraw-Hill Education, New York.

Bortfeldt, A., Winter, T., 2009. A genetic algorithm for the two-dimensional knapsack problem with rectangular pieces. International Transactions in Operational Research 16 (6), 685–713.

Breiman, L., 2001. Random forests. Machine Learning 45 (1), 5–32.

Brown, W. M., Tucker, W. T., 1961. The marketing center: Vanishing shelf space. Atlanta Economic Review 9, 9–13.

Bundesagentur für Arbeit, 2019. Der arbeitsmarkt in deutschland 20. Blickpunkt Arbeitsmarkt 66 (2).

Campo, K., Gijsbrechts, E., Nisol, P., 2004. Dynamics in consumer response to product unavailability: do stock-out reactions signal response to permanent assortment reductions? the influence of culture on services. Journal of Business Research 57 (8), 834–843.

Carvalho-Silva, M., Monteiro, M. T. T., de Sá-Soares, F., Dória-Nóbrega, S., 2018. Assessment of forecasting models for patients arrival at emergency department. Operations Research for Health Care 18, 112 – 118, eURO 2016–New Advances in Health Care Applications.

Central Intelligence Agency, 2019. The CIA World Factbook 2019-2020 -, 2019th Edition. Skyhorse, United States.

Ceschia, S., Schaerf, A., 2011. Local search and lower bounds for the patient admission scheduling problem. Computers & Operations Research 38 (10), 1452–1463.

Ceschia, S., Schaerf, A., 2012. Modeling and solving the dynamic patient admission scheduling problem under uncertainty. Artificial intelligence in medicine 56 (3), 199–205.

Ceschia, S., Schaerf, A., 2016. Dynamic patient admission scheduling with operating room constraints, flexible horizons, and patient delays. Journal of Scheduling 19 (4), 377–389.

Chandon, P., Hutchinson, W. J., Bradlow, E. T., Young, S. H., 2009. Does in-store marketing work? effects of the number and position of shelf facings on brand attention and evaluation at the point of purchase. Journal of Marketing 73 (November), 1–17.

Chase, R., Jacobs, F., Aquilano, N., 2006. Operations Management for Competitive Advantage -, 11th Edition. McGraw-Hill/Irwin, Boston.

Cormen, T. H., Leiserson, C. E., Rivest, R. L., Stein, C., 2009. Introduction to Algorithms, 3rd Edition. MIT Press, Cambridge.

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

Cotta, C., Jul 2011. Effective patient prioritization in mass casualty incidents using hyperheuristics and the pilot method. OR Spectrum 33 (3), 699–720.

Curhan, R. C., 1972. The relationship between shelf space and unit sales in supermarkets. Journal of Marketing Research 9 (4), 406–412.

Curseu, A., van Woensel, T., Fransoo, J., van Donselaar, K., Broekmeulen, R., 2009. Modelling handling operations in grocery retail stores: An empirical analysis. Journal of the Operational Research Society 60 (2), 200–214.

DeHoratius, N., Raman, A., 2008. Inventory record inaccuracy: An empirical analysis. Management Science 54 (4), 627–641.

DeHoratius, N., Ton, Z., 2015. The role of execution in managing product availability. In: Agrawal, N., Smith, S. A. (Eds.), Retail Supply Chain Management. International Series in Operations Research & Management Science. Springer US, pp. 53–77.

Demeester, P., Souffriau, W., De Causmaecker, P., Vanden Berghe, G., 2010. A hybrid tabu search algorithm for automatically assigning patients to beds. Artificial Intelligence in Medicine 48 (1), 61–70.

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

Donselaar, K. H. v., Gaur, V., Woensel, T. v., Broekmeulen, R. A., Fransoo, J. C., 2010. Ordering behavior in retail stores and implications for automated replenishment. Management Science 56 (5), 766–784.

Dorgham, K., Nouaouri, I., Ben-Romdhane, H., Krichen, S., 2019. A hybrid simulated annealing approach for the patient bed assignment problem. Procedia Computer Science 159, 408 – 417, knowledge-Based and Intelligent Information & Engineering Systems: Proceedings of the 23rd International Conference KES2019.

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

Duin, C., Voß, S., 1999. The pilot method: A strategy for heuristic repetition with application to the steiner problem in graphs. Networks 34 (3), 181–191.

Düsterhöft, T., Hübner, A., Schaal, K. N., 2019. A practical approach to the shelf-space allocation and replenishment problem with heterogeneously sized shelves. European Journal of Operational Research.

EHI Retail Institute, 2014. Retail data 2014: Structure, key figures and profiles of international retailing. EHI Retail Institute, Köln.

Eisend, M., 2014. Shelf space elasticity: A meta-analysis. Journal of Retailing 90, 168–181.

Eltze, C., Goergens, S., Loury, M., 2013. Grocery store operations: Which improvements matter most? Akzente 1 (1), 74–81.

Epitropakis, M. G., Burke, E. K., 2018. Hyper-heuristics. In: Martí, R., Panos, P., Resende, M. G. C. (Eds.), Handbook of Heuristics. Springer International Publishing, Cham, pp. 1–57.

Erhard, M., Schoenfelder, J., Fügener, A., Brunner, J. O., 2018. State of the art in physician scheduling. European Journal of Operational Research 265 (1), 1 – 18.

Eroglu, C., Williams, B. D., Waller, M. A., 2013. The backroom effect in retail operations. Production and Operations Management 22 (4), 915–923.

Essen, T. J. v., Houdenhoven, M. v., Hurink, J. L., 2015. Clustering clinical departments for wards to achieve a prespecified blocking probability. OR Spectrum 37 (1), 243–271.

Fogel, L., Owens, A., Walsh, M., 1966. Artificial Intelligence Through Simulated Evolution. John Wiley & Sons.

Frank, R. E., Massy, W. F., 1970. Shelf position and space effects on sales. Journal of Marketing Research 7 (1), 59–66.

Fügener, A., Hans, E. W., Kolisch, R., Kortbeek, N., Vanberkel, P. T., 2014. Master surgery scheduling with consideration of multiple downstream units. European Journal of Operational Research 239 (1), 227–236.

Gartner, D., Kolisch, R., 2014. Scheduling the hospital-wide flow of elective patients. European Journal of Operational Research 233 (3), 689–699.

Gartner, D., Padman, R., 2019. Flexible hospital-wide elective patient scheduling. Journal of the Operational Research Society 0 (0), 1–15.

Gaur, V., Honhon, D., 2006. Assortment planning and inventory decisions under a locational choice model. Management Science 52, 1528–1543.

Geismar, H. N., Dawande, M., Murthi, B. P., Sriskandarajah, C., 2015. Maximizing revenue through two–dimensional shelf–space allocation. Production and Operations Management 24, 1148–1163.

Gilland, W. G., Heese, H. S., 2013. Sequence matters: Shelf–space allocation under dynamic customer–driven substitution. Production and Operations Management 22 (4), 875–887.

Glover, F., 1986. Future paths for integer programming and links to artificial intelligence. Computers & Operations Research 13 (5), 533 – 549, applications of Integer Programming.

Goldberg, D. E., 1989. Genetic Algorithms in Search, Optimization and Machine Learning, 1st Edition. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.

Goodfellow, I., Bengio, Y., Courville, A., 2016. Deep Learning. MIT Press, `http://www.deeplearningbook.org`.

Gross, C. N., Fügener, A., Brunner, J., 2017. Online rescheduling of physicians in hospitals. Flexible Services and Manufacturing Journal 7 (2), 1–33.

Gruen, W. T., Corsten, S., Bharadwaj, S., 2002. Retail out-of-stocks: A worldwide examination of extent, causes and consumer responses: Report. Grocery Manufacturers of America, Washingtion and D.C.

Guido, R., Groccia, M. C., Conforti, D., 2018. An efficient matheuristic for offline patient-to-bed assignment problems. European Journal of Operational Research 268 (2), 486 – 503.

Gul, M., Celik, E., 2018. An exhaustive review and analysis on applications of statistical forecasting in hospital emergency departments. Health Systems 0 (0), 1–22.

Guyon, I., Elisseeff, A., 2003. An introduction to variable and feature selection. J. Mach. Learn. Res. 3, 1157–1182.

Haller, S., 2015. Dienstleistungsmanagement. Springer Fachmedien Wiesbaden.

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

Hansen, P., Heinsbroek, H., 1979. Product selection and space allocation in supermarkets. European Journal of Operational Research 3 (6), 474–484.

Herring, W. L., Herrmann, J. W., Apr 2012. The single-day surgery scheduling problem: sequential decision-making and threshold-based heuristics. OR Spectrum 34 (2), 429–459.

Hertwig, R., Pachur, T., 2015. Heuristics, history of. In: Wright, J. D. (Ed.), International Encyclopedia of the Social & Behavioral Sciences (Second Edition), second edition Edition. Elsevier, Oxford, pp. 829 – 835.

Hoerl, A. E., Kennard, R. W., 1970. Ridge regression: Biased estimation for nonorthogonal problems. Technometrics 12 (1), 55–67.

Holland, J. H., 1992. Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence, 2nd Edition. MIT Press, Cambridge, MA, USA.

Holm, L. B., Lurås, H., Dahl, F. A., 2013. Improving hospital bed utilisation through simulation and optimisation: With application to a 40% increase in patient volume in a norwegian general hospital. International Journal of Medical Informatics 82 (2), 80 – 89.

Holzapfel, A., Hübner, A., Kuhn, H., Sternbeck, M., 2016. Delivery pattern and transportation planning in grocery retailing. European Journal of Operational Research 252 (252), 54–68.

Hopfield, J. J., 1982. Neural networks and physical systems with emergent collective computational abilities. Proceedings of the National Academy of Sciences 79 (8), 2554–2558.

Hopper, E., Turton, B., 2001. An empirical investigation of meta-heuristic and heuristic algorithms for a 2d packing problem. European Journal of Operational Research 128 (1), 34 – 57.

Hübner, A., 2011. Retail category management: Decision support systems for assortment, shelf space and price planning. Lecture Notes in Economic and Mathematical Systems. Springer, Heidelberg.

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

Hübner, A., Kuhn, H., Sternbeck, M. G., 2013a. Demand and supply chain planning in grocery retail: An operations planning framework.

International Journal of Retail & Distribution Management 41 (7), 512–530.

Hübner, A., Kuhn, H., Sternbeck, M. G., 2013b. Retail operations: Why and how retailers benefit from an integrative supply chain management perspective. In: Wimmer, T., Hucke, S. (Eds.), Inspiration, Ideas, Innovation. DVV Media Group, Hamburg, pp. 359–439.

Hübner, A., Kuhn, H., Walther, M., 2018. Combining clinical departments and wards in maximum-care hospitals. OR Spectrum 40 (3), 679–709.

Hübner, A., Kühn, S., Kuhn, H., 2016. An efficient algorithm for capacitated assortment planning with stochastic demand and substitution. European Journal of Operational Research 250 (250), 505–520.

Hübner, A., Schaal, K., 2017a. An integrated assortment and shelf-space optimization model with demand substitution and space-elasticity effects. European Journal of Operational Research 261 (1), 302 – 316.

Hübner, A., Schaal, K., 2017b. A shelf-space optimization model when demand is stochastic and space-elastic. Omega 69, 139–154.

Hübner, A., Schäfer, F., Schaal, K. N., 2020. Maximizing profit via assortment and shelf-space optimization for two-dimensional shelves. Production and Operations Management 29 (3), 547–570.

Hübner, A., Walther, M., Kuhn, H., 2015. Approach to clustering clinical departments. In: Matta, A. (Ed.), Conference Proceedings of Second International Conference on Health Care Systems Engineering. Springer Proceedings in Mathematics & Statistics, Berlin, pp. 111–120.

Hulshof, P. J. H., Kortbeek, N., Boucherie, R. J., Hans, E. W., Bakker, P. J. M., 2012. Taxonomic classification of planning decisions in health care: a structured review of the state of the art in or/ms. Health Systems 1 (2), 129–175.

Hulshof, P. J. H., Mes, M. R. K., Boucherie, R. J., Hans, E. W., 2016. Patient admission planning using approximate dynamic programming. Flexible Services and Manufacturing Journal 28 (1-2), 30–61.

Irion, J., Lu, J.-C., Al-Khayyal, F., Tsao, Y.-C., 2012. A piecewise linearization framework for retail shelf space management models. European Journal of Operational Research 222 (1), 122–136.

Keijzer, M., Merelo, J. J., Romero, G., Schoenauer, M., 2002. Evolving objects: A general purpose evolutionary computation library. In: Lecture Notes in Computer Science. Springer Nature, pp. 231–242.

Kellerer, H., Pferschy, U., Pisinger, D., 2004a. Knapsack problems. Springer, Berlin.

Kellerer, H., Pferschy, U., Pisinger, D., 2004b. Knapsack Problems. Springer.

Kenessey, Z., 1987. The primary, secondary, tertiary and quaternary sectors of the economy. Review of Income and Wealth 33 (4), 359–385.

Kifah, S., Abdullah, S., 2015. An adaptive non-linear great deluge algorithm for the patient-admission problem. Information Sciences 295, 573 – 585.

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

Kohavi, R., John, G. H., 1997. Wrappers for feature subset selection. Artificial Intelligence 97 (1), 273 – 324, relevance.

Kök, G., Fisher, M. L., Vaidyanathan, R., 2015. Assortment planning: Review of literature and industry practice. In: Agrawal, N., Smith, S. A. (Eds.), Retail Supply Chain Management. Vol. 223 of International Series in Operations Research & Management Science. Springer US, pp. 175–236.

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

Krajewski, L. J., Malhotra, M. K., Ritzman, L. P., 2016. Operations Management - Processes and supply chains, student Edition. Pearson, München.

Krishnamoorthy, C. S., Venkatesh, P. P., Sudarshan, R., 2002. Object-oriented framework for genetic algorithms with application to space truss optimization. Journal of Computing in Civil Engineering 16 (1), 66–75.

Kursa, M., Rudnicki, W., 2010. Feature selection with the boruta package. Journal of Statistical Software, Articles 36 (11), 1–13.

LeCun, Y., Bengio, Y., Hinton, G., 2015. Deep learning. nature 521 (7553), 436–444.

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

Lodi, A., Martello, S., Vigo, D., 1999. Heuristic and metaheuristic approaches for a class of two-dimensional bin packing problems. Informs Journal on Computing 11, 345 – 357.

Lusby, R. M., Schwierz, M., Range, T. M., Larsen, J., 2016. An adaptive large neighborhood search procedure applied to the dynamic patient admission scheduling problem. Artificial Intelligence in Medicine 74, 21 – 31.

Luscombe, R., Kozan, E., 2016. Dynamic resource allocation to improve emergency department efficiency in real time. European Journal of Operational Research 255 (2), 593 – 603.

Martínez-de Albéniz, V., Roels, G., 2011. Competing for shelf space. Production and Operations Management 20 (1), 32–46.

Mladenović, N., Hansen, P., 1997. Variable neighborhood search. Comput. Oper. Res. 24 (11), 1097–1100.

Nielsen, A., 2004. Consumer-centric category management: How to increase profits by managing categories based on consumer needs. John Wiley & Sons, Inc., New Jersey.

Pieters, R., Wedel, M., Batra, R., 2010. The stopping power of advertising: Measures and effects of visual complexity. Journal of Marketing 74 (5), 48 – 60.

Pisinger, D., 2005. Where are the hard knapsack problems? Computers & Operations Research 32 (9), 2271 – 2284.

Pisinger, D., Sigurd, M., 2007. Using decomposition techniques and constraint programming for solving the two-dimensional bin-packing problem. INFORMS Journal on Computing 19 (1), 36–51.

Pólya, G., 2014. How to Solve It. Princeton University Press.

Rachuba, S., Werners, B., 2014. A robust approach for scheduling in hospitals using multiple objectives. Journal of the Operational Research Society 65 (4), 546–556.

Rajaram, K., Tang, C. S., 2001. The impact of product substitution on retail merchandising. European Journal of Operational Research 135 (3), 582–601.

Range, T. M., Lusby, R. M., Larsen, J., 2014. A column generation approach for solving the patient admission scheduling problem. European Journal of Operational Research 235 (1), 252–264.

Rechenberg, I., Toms, B., Establishment, R. A., 1965. Cybernetic Solution Path of an Experimental Problem:. Library translation / Royal Aircraft Establishment. Ministry of Aviation.

Reiner, G., Teller, C., Kotzab, H., 2012. Analyzing the efficient execution on in-store logistics processes in grocery retailing – the case of dairy products. Production and Operations Management 22 (4), 924–939.

Ryzin, G. v., Mahajan, S., 1999. On the relationship between inventory costs and variety benefits in retail assortments. Management Science 45, 1496–1509.

Schaal, K., Hübner, A., 2018. When does cross-space elasticity matter in shelf-space planning? a decision analytics approach. Omega 244, 135–152.

Schäfer, F., Walther, M., Grimm, D. G., Hübner, A., 2020. Combining machine learning and optimization for the operational patient-bed assignment problem. Working Paper Technical University of Munich.

Schäfer, F., Walther, M., Hübner, A., 2017. Patient-bed allocation in large hospitals. In: Cappanera, P., Li, J., Matta, A., Sahin, E., Vandaele,

N. J., Visintin, F. (Eds.), Health Care Systems Engineering. Springer International Publishing, Cham, pp. 299–300.

Schäfer, F., Walther, M., Hübner, A., Kuhn, H., Dec 2019. Operational patient-bed assignment problem in large hospital settings including overflow and uncertainty management. Flexible Services and Manufacturing Journal 31 (4), 1012–1041.

Schiele, J., Koperna, T., Brunner, J. O., 2019. Predicting bed occupancy for integrated surgery scheduling via neural networks, working Paper.

Schmidt, R., Geisler, S., Spreckelsen, C., 2013. Decision support for hospital bed management using adaptable individual length of stay estimations and shared resources. BMC Medical Informatics and Decision Making 13 (3), 1–19.

Schwefel, H., 1965. Kybernetische Evolution als Strategie der experimentellen Forschung in der Strömungstechnik. Technische Universität Berlin, Hermann Föttinger-Institut für Strömungstechnik.

Séguin, S., Villeneuve, Y., Blouin-Delisle, C.-H., 2019. Improving patient transportation in hospitals using a mixed-integer programming model. Operations Research for Health Care 23, 100202.

Sharma, S., Abouee-Mehrizi, H., Sartor, G., 2019. Inventory management under storage and order restrictions. Production and Operations Management.

Smith, S. A., Agrawal, N., 2000. Management of multi-item retail inventory systems with demand substitution. Operations Research 48 (1), 50–64.

Sörensen, K., Glover, F. W., 2013. Metaheuristics. In: Gass, S. I., Fu, M. C. (Eds.), Encyclopedia of Operations Research and Management Science. Springer US, Boston, MA, pp. 960–970.

Sörensen, K., Sevaux, M., Glover, F., 2018. A history of metaheuristics. In: Martí, R., Panos, P., Resende, M. G. C. (Eds.), Handbook of Heuristics. Springer International Publishing, Cham, pp. 1–18.

Statistisches Bundesamt, 2019. Umsatzsteuerstatistik (voranmeldungen). Finanzen und Stuern 14 (8.1).

Tan, B., Karabati, S., 2013. Retail inventory management with stock-out based dynamic demand substitution. International Journal of Production Economics 145 (1), 78–87.

Taramasco, C., Olivares, R., Munoz, R., Soto, R., Villar, M., de Albuquerque, V. H. C., 2019. The patient bed assignment problem solved by autonomous bat algorithm. Applied Soft Computing 81, 105484.

Thielen, C., 2018. Duty rostering for physicians at a department of orthopedics and trauma surgery. Operations Research for Health Care 19, 80 – 91.

Tibshirani, R., 1996. Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society. Series B (Methodological) 58 (1), 267–288.

Turhan, A. M., Bilgen, B., 2017. Mixed integer programming based heuristics for the patient admission scheduling problem. Computers & Operations Research 80, 38–49.

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

van Oostrum, J. M., Van Houdenhoven, M., Hurink, J. L., Hans, E. W., Wullink, G., Kazemier, G., Apr 2008. A master surgical scheduling approach for cyclic scheduling in operating room departments. OR Spectrum 30 (2), 355–374.

Vanberkel, P. T., Boucherie, R. J., Hans, E. W., Hurink, J. L., Litvak, N., Apr 2012. Efficiency evaluation for pooling resources in health care. OR Spectrum 34 (2), 371–390.

Vancroonenburg, W., De Causmaecker, P., Vanden Berghe, G., 2016. A study of decision support models for online patient-to-room assignment planning. Annals of Operations Research 239 (1), 253–271.

Voß, S., Fink, A., Duin, C., 2005. Looking ahead with the pilot method. Annals of Operations Research 136 (1), 285–302.

Wargon, M., Guidet, B., Hoang, T. D., Hejblum, G., 2009. A systematic review of models for forecasting the number of emergency department visits. Emergency Medicine Journal 26 (6), 395–399.

Wäscher, G., Haußner, H., Schumann, H., 2007. An improved typology of cutting and packing problems. European Journal of Operational Research 183 (3), 1109 – 1130.

Xue, W., Caliskan Demirag, O., Chen, F. Y., Yang, Y., 2017. Managing retail shelf and backroom inventories when demand depends on the shelf-stock level. Production and Operations Management 26 (9), 1685–1704.

Yuan, M., Lin, Y., 2006. Model selection and estimation in regression with grouped variables. Journal of the Royal Statistical Society: Series B (Statistical Methodology) 68 (1), 49–67.

Zhang, L., Wong, T., 2015. An object-coding genetic algorithm for integrated process planning and scheduling. European Journal of Operational Research 244 (2), 434 – 444.

Zou, H., Hastie, T., 2005. Regularization and variable selection via the elastic net. Journal of the Royal Statistical Society: Series B (Statistical Methodology) 67 (2), 301–320.

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

Appendix

A UML

Figure A.1 determines the structure and notation of the GA and visualizes the main classes (bold heading), attributes (marked by minus) and operations (marked by plus). The connectors between the classes denote the interdependence of these and the associated numbers define the cardinality between the classes. One GA consists of one-to-many populations, one population consists of one-to-many individuals, one individual consists of one container, and one container consists of one-to-many rectangles. All these relationships are compositions, which means that no instance of a class can exist without an instance of the predecessor class. Furthermore, each rectangle has an item reference. The relationship type between rectangle and item is called an aggregation. It means that instances of these classes can be independently generated. Hence, the set of items N remains the same over the complete GA execution. The aggregation makes it possible to define the rectangles of each item i without duplication of the item set N.

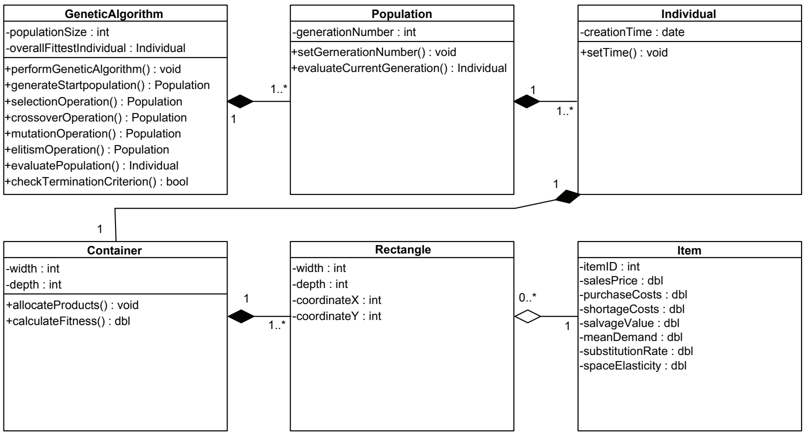

GeneticAlgorithm
-populationSize : int
-overallFittestIndividual : Individual
+performGeneticAlgorithm() : void
+generateStartpopulation() : Population
+selectionOperation() : Population
+crossoverOperation() : Population
+mutationOperation() : Population
+elitismOperation() : Population
+evaluatePopulation() : Individual
+checkTerminationCriterion() : bool
1..*
1
Population
-generationNumber : int
+setGernerationNumber() : void
+evaluateCurrentGeneration() : Individual
1
1..*
Individual
-creationTime : date
+setTime() : void
1
1
Container
-width : int
-depth : int
+allocateProducts() : void
+calculateFitness() : dbl
1
1..*
Rectangle
-width : int
-depth : int
-coordinateX : int
-coordinateY : int
0..*
1
Item
-itemID : int
-salesPrice : dbl
-purchaseCosts : dbl
-shortageCosts : dbl
-salvageValue : dbl
-meanDemand : dbl
-substitutionRate : dbl
-spaceElasticity : dbl

Figure A.1: *UML* class diagram of the object-coding genetic algorithm