NTNU
Norwegian University of
Science and Technology

An Optimization-Based Approach to Facility Layout Planning in Hospitals

Vilde Nielsen Kvillum
Anne Marit Rørholt Vigerust

Industrial Economics and Technology Management
Submission date: June 2018
Supervisor: Henrik Andersson, IØT
Co-supervisor: Anders Gullhav, IØT
Bjørn Nygreen, IØT

Norwegian University of Science and Technology
Department of Industrial Economics and Technology Management

Problem Description

The Facility Layout Problem is formulated and implemented with the purpose of being a decision-making tool in hospital layout planning. The model is solved with a two-stage iterative solution method, using mixed-integer programming on each stage. In addition to a technical study exploring different features of the implementation, a case study on the new hospital in Hammerfest is conducted in collaboration with Sykehusbygg HF.

Preface

This master's thesis concludes integrated Master of Science of Managerial Economics and Operations Research at the Department of Industrial Economics and Technology Management in Faculty of Economics at the Norwegian University of Science and Technology (NTNU). The thesis builds on the work performed in the specialization project TIØ4500 of the fall of 2017.

The thesis is written in collaboration with Sykehusbygg HF, a public enterprise involved in construction- and rebuilding projects of hospitals in Norway. The collaboration comprises the facility layout planning of hospital buildings, a complex process that considers locating hospital functions inside the footprint of a building. The information related to hospitals in this thesis is obtained from the projected new hospital in Hammerfest. This construction project is currently in the planning process.

We would like to express our sincere gratitude to our supervisors, Professor Henrik Andersson, Professor Bjørn Nygreen, and Postdoc. Anders N. Gullhav at NTNU for your valuable guidance and important discussions throughout the process. We would also like to thank Unni Dahl, Anneli Tyvold, and Rita Konstante at Sykehusbygg HF for their input through meetings and e-mail correspondence, and Siri Rørholt at Rambøll Trondheim for providing insight into the architectural approach of projecting a hospital building.

Trondheim, June 11th, 2018

Vilde NK   Anne Marit RV

Vilde Nielsen Kvillum   Anne Marit Rørholt Vigerust

Abstract

There are several major decisions involved when building a hospital. The purpose of the hospital needs to be established, and the associated departments that will constitute the hospital need to be determined. Also, the geographical location must be decided upon, and afterward the planning of the design of the hospital is executed. When building or rebuilding a hospital, a decision of great importance is the structure of the internal layout of the hospital. A well designed internal layout means reduced operational costs, avoidance of unnecessary workloads for employees, decreasing distances for both patients and employees, and as a consequence better abilities to save lives. New hospitals have to be built in a way that accounts for expected (and unexpected) future changes in demographics, adaptions to new technologies and discoveries in medicine (Sykehusbygg HF, 2017b).

This master's thesis is written in collaboration with Sykehusbygg HF, the public enterprise responsible for the planning of major hospital construction- and rebuilding projects in Norway. The goal of the thesis is to illustrate the use of mathematical optimization in generating possible layouts for hospitals based on relatedness between functions, in addition to other relevant considerations. The layouts are supposed to be a decision support tool in the process of planning hospital layouts at Sykehusbygg HF. This thesis formulates a facility layout problem for hospitals, where a diverse set of hospital *functions*, such as emergency departments, bed wards, polyclinics and medical imaging labs, has to be assigned unique locations on the *footprint* of a hospital building. The measure of interaction is expressed as values of *proximity*, parameters describing desired closeness defined between pairs of functions.

The footprint of the building is defined as a set of locations available for placement of functions. Each function can be spread across several different locations, as long as the locations are defined as neighboring locations. Each location can contain fractions of several different functions. Due to the way the footprint, functions, and the decision of placement of functions are defined in this thesis, the minimal

degree of discretization enables the model to easily account for realistic, non-rectangular buildings and functions of a continuous range of sizes. As far as the research of thesis is concerned, this approach is non-existent within the literature. The objective of the mathematical model has features similar to the quadratic assignment problem of allocating functions to locations considering the relations between functions. The objective seeks to develop a layout with minimum total distance between pairs of functions weighted by the pairs' proximity values. The problem is linearized and formulated as a mixed-integer program (MIP). Due to the objective function being dependent on interrelated placements of functions, the problem is tough to solve for real-world cases.

A case study is performed, where the model is tested on larger instances, including a data set from the ongoing planning process of the hospital in Hammerfest. In the case study, a two-stage solution method with exact programming methods in each stage is used. In the first stage, functions are assigned to floors, while the second stage handles the internal distribution of functions on each floor consecutively. Different iterative approaches to solving the floors in stage 2 are examined and evaluated. In addition, a resulting layout from using the two-stage approach on real data is proposed for Hammerfest Hospital and analyzed based on operational aspects.

The performance of the model is quantified by calculating the correspondence in average distances between functions placed and the proximity values between them. The results show a decreasing trend of distances between functions with increasing value of proximity, indicating that the considerations implemented are accounted for by the model. The results prove the model's ability to find layouts that proficiently account for the requirements for closeness between functions, emphasizing different proximity values and specific needs for locating functions.

This master's thesis illustrates how mathematical optimization can be exploited in hospital layout planning processes. After formulation and implementation, the iterative two-stage approach to the problem has successfully reached the goal of generating convenient layouts. In conclusion, with additional considerations of operational elements included, the model could directly facilitate the layout planning process and work as input to decision making.

Sammendrag

Det er flere viktige beslutninger som må tas når et sykehus skal bygges. Hensikten til sykehuset må fastslås, og de tilhørende avdelingene som skal være en del av sykehuset må bestemmes. Også den geografiske lokasjonen skal bestemmes, og derettes skal selve utformingen av sykehuset besluttes. Ved byggingsprosjekter eller ombyggingsprojekter av sykehus er også sykehusets indre utforming avgjørende. En godt utformet layout kan lede til reduserte driftskostnader, mindre arbeidsmengde for de ansatte, reduserte avstander for både pasienter og ansatte, og kan i ytterste konsekvens skape bedre grunnlag for å redde liv. Nye sykehus må bygges på en måte som står til forventede (og uforventede) fremtidige endringer i demografi, tilpasning til ny teknologi og oppdagelser innen medisin (Sykehusbygg HF, 2017b).

Denne masteroppgaven er skrevet i samarbeid med Sykehusbygg HF, en offentlig virksomhet som har ansvar for planlegging av store sykehusbyggings- og gjenoppbyggingsprosjekter i Norge. Målet med oppgaven er å illustrere bruken av matematiske optimeringsmetoder ved å generere mulige layouter i sykehus basert på behovet for nærhet mellom funksjoner, i tillegg til andre relevante hensyn. Layoutene skal fungere som et beslutningsverktøy i planleggingen av sykehuslayouter hos Sykehusbygg HF.

Masteroppgaven formulerer et bygningslayoutproblem (FLP) for sykehus, hvor et mangfolding antall *funksjoner*, for eksempel akutte avdelinger, sengeområder, poliklinikker og bildediagnostikk må tilordnes unike lokasjoner i bygningen. Målet på interaksjon mellom funksjonene er uttrykt gjennom *nærhetsverdier*, parametere som beskriver behovet for nærhet mellom et par av funksjoner, som i hovedsak består av hyppighet og viktighet av flyt mellom funksjonene.

Fotavtrykket til bygningen er oppdelt i et sett av lokasjoner som er tilgjengelige for plassering av funksjoner. Når funksjonene skal allokeres i bygningen kan hver funksjon spres på flere ulike lokasjoner, så lenge lokasjonene er definert som nærliggende nabolokasjoner. Hver lokasjon kan også inneholde fraksjoner av forskjel-

lige funksjoner. Måten fotavtrykk, funksjoner og metoden for å plassere funksjoner er definert i denne oppgaven gir minimal diskretisering og gjør det enkelt for modellen å ta hensyn til realistiske, ikke-rektangulære bygninger og funksjoner i mange ulike størrelser og fasonger. Denne tilnærmingen er ikke observert i litteraturen som er blitt studert i tilknytning til denne masteroppgaven. Objektivfunksjonen til den matematiske modellen har trekk lignende det kvadratiske tilordningsproblemet (QAP), som allokerer funksjoner til lokasjoner med tanke på nærhetsbehovet mellom funksjonene. Objektivfunksjonen tilstreber å utvikle en layout med kortest mulige avstander mellom par av funksjoner, vektet med parets nærhetsverdi. Problemet er linearisert og formulert som et blandet heltallsproblem. Grunnet den kvadratiske naturen til problemet er det svært vanskelig å løse for realistiske, store instanser.

I tillegg til et teknisk studie av aspekter med implementasjonen av modellen er et case-studie utført, der den matematiske modellen testes på større instanser, inkludert en instans basert på den pågående planleggingsprosessen til sykehuset som skal bygges i Hammerfest. I case-studiet benyttes en to-stegs løsning som bruker eksakte løsningsmetoder i hvert steg. I første steg allokeres funksjoner til de ulike etasjene av bygningen. I steg to blir den interne fordelingen på hver etasje gjort i egne sub-problemer. Ulike iterative tilnærminger til å løse etasjene i steg 2 er undersøkt og evaluert. I tillegg er layouten basert på data fra Sykehusbygg analysert basert på operasjonelle aspekter.

Modellens ytelse kvantifiseres ved å beregne korrelasjonen mellom gjennomsnittlig avstand mellom par av funksjoner og nærhetsverdiene mellom dem. Resultatene viser en avtagende trend av avstander mellom funksjoner med økt nærhetsverdi, noe som indikerer at målet til modellen er tilfredstillt. Resultatene viser modellens evne til å lage layouter som tar hensyn til behov for nærhet mellom funksjoner og andre behov spesifisert for allokering av funksjoner.

Denne masteroppgaven illustrerer hvordan matematiske optimeringsmetoder kan utnyttes i deler av sykehusplanlegging som angår layouten til sykehuset. Etter formulering og implementering av den matematiske modellen, har iterative to-stegs løsningsmetoder lykkes i å nå målet om å lage praktiske layouter. Avslutningsvis, når operasjonelle hensyn ivaretas, kan modellen brukes som et beslutningsverktøy i diskusjoner og planlegging av sykehusbygg.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The number of regional health institutions (hospitals with emergency department and obstetrics) in Norway has decreased from 34 in 2006 to 24 today (Helse- og omsorgsdepartementet, 2017; Braut, 2015). In addition, more patients are treated at each facility, making the need for efficient internal layouts of hospitals asserting (Statistisk Sentralbyrå, 2017).

The layouts of many hospitals today are results of processes of constant growth and development through the years. The hospitals are built with an original purpose, but have later typically been expanded in size, as the base of patients and need for treatment has increased and changed in line with a growing population. In addition to this, modern hospitals require a whole range of adaptions to new technologies and equipment that the original hospital buildings are not designed to accommodate.

The results of the frequent changes and reconstructions are poor layouts that may be sub-optimal in regards to the efficiency of the hospital. As many hospitals were built based on a patient basis different from today, these difficulties are hard to avoid without total reconstructions of the hospitals. The results and experiences from inefficient solutions in expanding older hospitals motivates a structured and targeted planning process when building a new, or fully renovating an old hospital (Sykehusbygg HF, 2017b).

To achieve the desired and needed structure of hospitals in Norway, new hospitals are built, and several existing hospitals undergo extensive renovation and extensions (Sykehusbygg HF, 2018c). The planning process when constructing or reconstructing a hospital is comprehensive and time-consuming. Based on experience, a myriad of considerations are taken into account. Creating a new hospital consists of several phases and a line of essential decisions regarding the characteristics and purpose. Size, complexity, number of departments, specialties, location and several other aspects regarding the hospital need to be resolved. After these decisions have been made, the layout of the hospital is to be determined (Sykehusbygg HF, 2017b).

Hospital Layout Planning, the handling of planning the internal layout of a hospital, is an area of great importance and high complexity. The results of a well-designed hospital are efficient transportation patterns, good experiences for patients and pleasant environments for treating people in need. On the contrary, a poorly planned hospital will cause higher operating costs, additional workloads for the employees, longer walking distances for both patients and employees, an inexpedient waiting time for patients and a weaker foundation for good patient experiences in general.

Research in the field of Facility Layout Problems (FLPs) suggests methods for idolizing optimization in several types of planning problems, including the Hospital Layout Planning problem. Using FLPs in this context opens up the need for incorporating considerations necessary for this specific environment of application. These considerations may not be seen in industrial applications, an area where FLPs are highly exploited in literature. Hospital Facility Layout Problems (HLPs) are not to a great extent exploited in the existing literature. The papers relevant to this topic typically account for parts of the complex planning problem or adapts substantial simplifications.

The purpose of this thesis is to illustrate a way of exploiting mathematical optimization in the layout planning process of a hospital. Ultimately, the model can be used to generate possible hospital layouts, emphasizing the varying needs for closeness between departments and provide different, comparable solutions based on the perspectives of interest groups. The model should allocate functions while aiming to minimize distances between the functions placed weighted by the relatedness between them. An appropriate solution method should be applied to generate layouts of sufficient accuracy in a reasonable amount of time. The layouts can contribute as a decision support tool in the hospital planning process, being adjustable according to hospital planners and architects qualitative considerations. The master's thesis is motivated by a collaboration with Sykehusbygg Helseforetak (Sykehusbygg HF), a public enterprise organizing major construction- and rehabilitation projects of hospitals in Norway. Sykehusbygg HF is currently in the process of planning a new hospital in the city of Hammerfest, and the plans for this hospital is used as data for the case study of this thesis. Today, the layout planning of hospitals at Sykehusbygg does not include the use of mathematical optimization (Sykehusbygg HF, 2017a).

The formulation of the model of this thesis divides the building of the hospital in several adjacent *locations*, and defines the departments and other units of the hospital as *functions* to be placed. The model operates with both allowing functions to be divided over different locations and for each location to contain parts of several different functions. As far as the literature study performed in this thesis is concerned, this part of the approach is novel within the field of facility layout planning. The approach gives flexibility that can hopefully capture sufficient realistic aspects in order to make the model applicable to real hospitals.

Challenges when planning hospitals involve considerations of actors with different views on the internal organization of the hospital. Relevant actors include patients, employees, management, researchers or politicians (Sykehusbygg HF, 2017a). When considering movement patterns of patients, access to important departments should be prioritized, and the distances between locations relevant to the patient should be minimized. From an employees' point of view, the travel distance between locations of his or her different work tasks should be as short as possible, so that focus can be given to core tasks rather than walking between different locations. Management and politicians may have economic prioritization in mind, working to keep costs low and areas flexible. The transportation of equipment, food, sheets, and medication also affects the overall efficiency of the hospital. The result of the variation in opinions of the interest groups along with the different function's requirements for closeness to one another, and the specifications given for each hospital planning project is a large amount of unstructured, sometimes partly contradictory data, which in turn leads to a complex planning problem.

The placements of functions and the resulting internal distances at the hospital determine the flows of patients, personnel, and materials, which in turn constitute the overall effectiveness of internal interactions at the hospital. The focus on needs for closeness and interactions with the goal of obtaining reasonable and realistic solutions is an approach adapted from the current planning processes of Sykehusbygg HF. Placing functions in different locations of a building with any building structure while considering important characteristics the locations must satisfy, differ from most of the current research on these types of problems. The need for closeness between functions related to each other is the most important considerations when planning the layout of hospitals (Sykehusbygg HF, 2017a).

The concepts discussed in this master's thesis is somewhat similar to the origin of the preceding Specialization Project performed in the Autumn of 2017 (Kvillum and Vigerust, 2017). The nature of the problem description of the master's thesis is similar to the one in the specialization project since both are performed during the planning process of Hammerfest Hospital. The model of the thesis differs from the model of the specialization project in modeling approach, and is generalized with features to be used on hospitals of all building structures, incorporates floors, and is developed with deliberation on solving the case study.

Chapter 2 provides a presentation of the background on hospital planning, the collaborating enterprise Sykehusbygg, and the Hospital in Hammerfest. Following, Chapter 3 presents relevant research on optimization methods and discusses the transferability of these methods to hospital planning problems. In Chapter 4, a description of the problem of this thesis is provided, with assumptions made concerning the real world problem. From this, Chapter 5 presents the mathematical model developed, and discusses assumptions regarding the modeling approach. Chapter 6 elaborates and presents the approach to solution method of this thesis. In Chapter 7, key aspects of the implementation of the model are presented, and a technical study that exploits the mathematical and technical features of the model and adaptions to different data is performed. A case study concerning the hospital currently being planned in Hammerfest is presented in Chapter 8, testing the model on real data obtained from this project through Sykehusbygg. Chapter 9 concludes on main findings of the thesis, and Chapter 10 suggests the focus for possible further research and extensions of the work.

Chapter 2

Background

This chapter introduces background information relevant to the thesis. The information is primarily based on an idea phase report and a project report created by Sykehusbygg concerning the planning process of the hospital of Hammerfest, along with meetings and correspondence with Sykehusbygg. A significant part of the information presented in this section is similar to what is presented in the preceding specialization project in the fall of 2017 since the project was based on an earlier stage of the planning process of Hammerfest Hospital. Also, an architect from Rambøll, with extensive experience in hospital planning and design of hospital buildings is consulted for external views and input. Section 2.1 introduces terminology used in this thesis. Following, Sykehusbygg and their planning process when building hospitals are presented in Section 2.2, while Section 2.4 present a brief introduction to the hospital in Hammerfest.

2.1 Terminology

Hospital layout planning - Hospital layout planning refers to the part of the planning process of a hospital considering the planning of the internal layout of the building(s). This process follows the decisions of location, footprint, and the number of floors of the hospital building. The layout of the hospital is designed by deciding where to place *functions* in the building to achieve efficient operation of the hospital.

Functions - A function is in this thesis defined as a department or a specific section of the hospital serving a particular purpose and is to be placed in the hospital building as one, somewhat compact unit. Examples include the emergency department, polyclinics and bed wards. Parts of the hospital such as kitchenettes, toilets, and storage rooms are assumed included in the gross areas of the functions and are therefore not defined as separate functions.

Proximity - Proximity is defined as the requirement for closeness between pairs of functions and is given as a number on a chosen scale (zero to ten in this thesis). A high number indicates a high requirement for closeness. The proximity values are to be based on the relation between the functions, mostly considering the importance and magnitude of the flows of patients, doctors, nurses, materials and information between the functions. The proximity values vary depending on different *perspectives* on the need for interaction and closeness between functions.

Perspectives - Perspectives are used to describe the different point of views regarding what proximity values pairs of functions requires. Each perspective includes a set of proximity values for relevant pairs of functions. The values of proximity are to be given different weights that depend on the perspective in focus, to produce layouts with various considerations. Different perspectives could be linked to different stakeholders, or to varying views on economic concerns.

2.2 Sykehusbygg HF

The public part of the Norwegian health care system is organized in four centrally controlled health regions based on geographic location. Sykehusbygg HF (Sykehusbygg Helseforetak), hereby referred to as Sykehusbygg, is a public enterprise owned by the health regions. Sykehusbygg was established in October 2014 and contributes to significant construction- and rehabilitation projects concerning hospital buildings with a total budget above 500 million NOK (Sykehusbygg HF, 2017c). Sykehusbygg has 90 employees, located at the central office in Trondheim, and a local office in Oslo. Sykehusbygg is currently involved in eleven hospital projects, where eight of the projects are in the planning process, and three are under construction (Sykehusbygg HF, 2018c). One of the purposes of Sykehusbygg is to develop and sustain knowledge on innovative and efficient construction of health-care buildings through learning, innovation, experience and transfer of knowledge. This is achieved through close involvement in all similar processes of a significant size. As previously mentioned, Sykehusbygg does not apply mathematical optimization nor formulates the planning process as mathematical models today. Their methods are based on an extensive basis of experience and knowledge from earlier projects and meetings and discussions with involved actors.

2.3 The planning process

The process of planning and building a hospital is as mentioned comprehensive and involves several different phases. First, the size and complexity of the hospital, along with purposes and tasks the hospital needs to fulfill are defined. Some hospitals are meant to have special functions, and these should be included in the initial planning process. Early in the planning process, a project group that includes Sykehusbygg and actors relevant to the hospital in question, typically doctors and nurses, representatives for patients, government and management, is established. If a new hospital is to be built, the geographic location is an important choice to be made. Due to a steady decrease in the number of hospitals in Norway (Helse- og omsorgsdepartementet, 2017), the question of locations of new hospitals is a one highly debated. However, this thesis does not concern this debate other than a recognition of importance in the relationship between geographic location and operation.

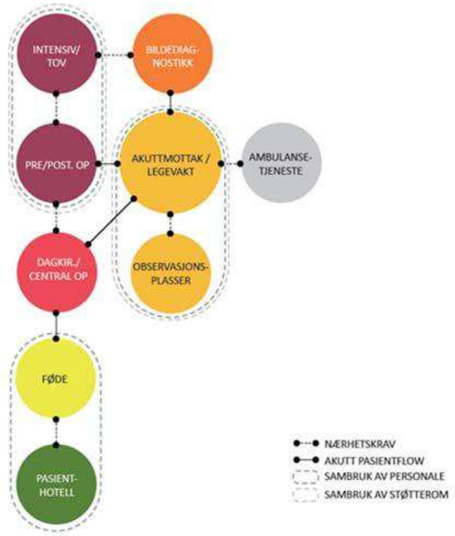
After establishing geographic location, the footprint and the shape of the building need to be chosen. Hospital buildings are in general complicated structures with many special needs and requirements. Their layout and architecture often come as a result of the traits of their designated locations, regulations and demands from the authorities, and the economic boundaries. Many demands need to be satisfied when building a hospital. Following the development of hospital buildings through the years, regularly shaped buildings have been forced to give way to more irregular, peculiar footprints, often providing efficient flows of patients and goods (Rambøll Norge AS, 2018).

When central decisions regarding purposes of the hospital and what functions to include in the hospital, geographical location and footprint are made, the next phase consists of determining where functions inside the hospital should be located, and the relation between them. This phase is the focus of this thesis. The process of planning the internal layout of a hospital is based on insight gathered from relevant stakeholders including employees, patients and others affected by the organizing and use of the hospital building. This insight is combined with experience from past projects and used as the background for discussions. The results of the discussions work as guidelines for the hospital architects, who develop a suitable layout for the hospital (Sykehusbygg HF, 2017b).

Sykehusbygg works systematically on retrieving information from stakeholders of the hospital. Population growth, disease development, medical and technological development, and interaction with other medical services are also important aspects to consider in projecting a hospital (Sykehusbygg HF, 2017c). Functions have to be placed in appropriate locations to make the flows of patients, staff, and material efficient. Vital functions need to be expediently located in the building, and in relation to each other. With the right placement of functions, the hospital could also save significant operation costs. When performing the process of placing functions inside the building, it is impossible to comply the wishes of everyone involved, and the process of deciding which considerations to emphasize is a significant part of the planning process.

Sykehusbygg uses a tool of proximity measures to get an overview of the need for closeness between functions and to distinguish and prioritize different perspectives on the internal organizing. Information from stakeholders contains descriptions of important interactions and flows between functions in the hospital based on their point of view. Sykehusbygg translates this information into executive diagrams that illustrate requirement for proximity, called proximity diagrams. Proximity diagrams regarding the emergency functions and bed wars are shown in Figure 2.1. The diagrams describe each function's need for relation to other functions, with flows of different importance and aspects of the relationships illustrated for different parts of the hospital.

An essential responsibility of Sykehusbygg is evaluating the possible solutions from the different point of views. Perspectives vary when focusing on various considerations. The distances the patients need to travel should be weighted against the distances the doctors and nurses travel between workstations. The perspectives are used in combination to make decisions, trying to ensure all essential considerations are taken into account. Further into the planning process, architects are involved. The architects hold knowledge from planning a long line of hospital buildings, and use their experience from earlier projects, along with information and requirements from Sykehusbygg, to design the structure of the building.



(a) Emergency department

(b) Bed wards

Figure 2.1: Examples of Proximity Diagrams from Sykehusbygg

2.4 The Hospital of Hammerfest

The hospital in Hammerfest is a part of the health region Helse Nord RHF, the Northern region of the Norwegian health care system. The current hospital was built in 1956 and have been modernized and extended several times from the 70's until today. These sporadic developments and extensions have resulted in somewhat weak logistics and inefficient use of area and resources. The ceiling heights are too low, and the buildings are not suitable for modern single bedroom sleeping areas, or modern infrastructure and equipment. To meet the requirements for a modern hospital towards 2030-2040, a decision was made to build a new hospital, move the hospital area to a new location, and three different alternatives for the building design have been evaluated. In contrast to one of the other options for Hammerfest Hospital, which consists of more rectangular shapes, which is traditionally commonly used for hospitals, and a decision of a bow-shaped footprint as shown in Figure 2.2 has been made for the hospital (Sykehusbygg HF, 2017b).

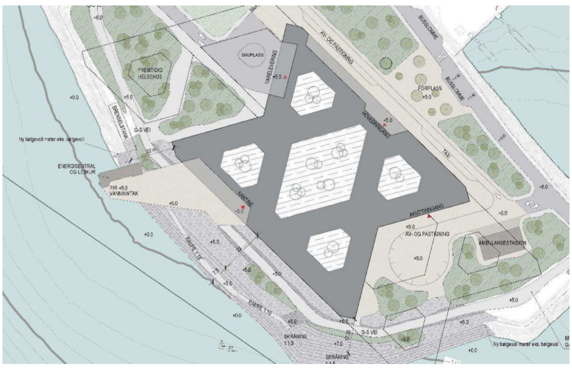


Figure 2.2: Location and footprint of the new hospital of Hammerfest

Sykehusbygg has decided the location of the hospital, the design of the building, and has suggested a layout for the inside of the building. These plans are elaborated in a concept report made for the hospital of Hammerfest (Sykehusbygg HF, 2018b). The information retrieved from Sykehusbygg includes a list of departments that will be included in the hospital, the layout suggested by architects and proximity diagrams for some groups of functions. Hammerfest hospital is a hospital with emergency functions, and also, other functions like bed wards, intensive care, imaging, and polyclinics are included in the list of functions. In conjunction with the planning of the hospital in Hammerfest, six participation groups have been created, each with different perspectives on the ideal organizing of the hospital. All groups have opinions needed to be recognized by Sykehusbygg in the planning process, and all have different perspectives on the importance of proximity between functions in the hospital.

The new hospital to be built in Hammerfest is used as motivation for this thesis and forms the foundation for the case study performed in Chapter 7, where real data of the hospital is exploited. As the hospital is currently in a phase of the planning where the internal allocation of functions is relevant, the aim is for the work of this thesis to contribute with insights and suggestions. Besides, the work already done by Sykehusbygg to establish and collect the needs for proximity from the stakeholders gives a foundation for retrieving the data needed for the model.

Chapter 3

Literature

The focus of this chapter is to present literature related to the hospital layout planning problem. In Section 3.1, relevant literature is discussed through a classification of different aspects of the relevant problem. In Section 3.2, the model developed in this master's thesis is positioned in relation to the classification. The concepts of the problem, and hence the relevant literature, are in some aspects similar to the preceding specialization project performed in the fall of 2017 (Kvillum and Vigerust, 2017). Where relevant, the traits of this earlier developed model are discussed as a part of the classification.

3.1 Classification of Facility Layout Problems

The Facility Layout Problem (FLP) involves the relative placement of units in a facility layout (Tari and Neghabi, 2015). The FLP is a complex combinatorial optimization problem aiming to minimize the impact of transportation between units, by expediently allocating the units on a specified area. FLPs are widely applicable in several sectors, but in return, they are both complex and technically challenging. The many dimensions and resulting high number of variables make FLPs hard and time consuming to solve. An FLP is, as shown in Section 3.1.2, often expressed as having the objective function of a Quadratic Assignment Problem (QAP). Additional aspects deviating from the standard QAP are included based on the nature and choices of formulations of the various problems in the literature.

Using FLPs in hospital environments constitute a specific type of FLP called Hospital Facility Layout Problems (HFLP), in this thesis named the Hospital Layout

Problem (HLP). This literature review discusses FLPs in general, and HLPs in specific where relevant literature exists. FLPs vary on several aspects, and different approaches for classification of the problems have been proposed. Drira et al. (2007) developed a framework with the purpose of categorizing and describing FLPs. The classification used in the following sections are partly based on the work of Ahmadi et al. (2017) who have extended the framework of Drira et al. (2007). The framework discusses environment of application, formulation approaches of the objective function for the FLP, the state of the problem (static/dynamic), representation of areas, number of stages, and the solution methods for the problem. The goal of the classification is to obtain a greater understanding of the HLP and reasoning the choices of formulation and solution methods for the HLP of this thesis.

3.1.1 Environment of Application

FLPs are widely used in several sectors such as industrial, chemical, electronics and offshore. The FLP originates from industrial applications, and caused by the useful application of FLPs in locating machines for streamlining production facilities; these are the practical applications most exploited among FLPs (Drira et al., 2007). Most relevant to this thesis is the use of FLPs in relation to hospitals. Elshafei (1977) introduced the application of FLPs in hospital planning and formulated the problem of locating hospital departments in order to minimize distance traveled by patients and goods. This is the problem referred to as an HLP.

The planning of hospitals is mostly based on experience and knowledge of hospital planners and architects. Vos et al. (2007) discuss the application of mathematical optimization in hospital planning, and the appropriateness of using it as a tool for supporting efficient operating of hospitals from a logistics point of view. FLPs are well suited and could be equally relevant for hospital planning problems as for other applications, but the models are not as widely used or documented for these purposes (Ahmadi et al., 2017). In the industrial context, direct economic- and efficiency-benefits are recorded, and FLPs can repeatedly be used over time for improvement processes, as documented by Tompkins et al. (2010). HLPs may in many cases require a higher degree of information processing than other FLPs, as hospitals provide additional complexity of flows compared to for example a production line. Anyhow, if acquiring a sound basis of data, often by extrapolating the data to incorporate factors of the future, modeling the FLP either with exact data or with a simplification of parameters, a model for hospitals can be quite valuable. As an example, Helber et al. (2016) presented research and application of optimization on a hospital in Germany, developing an approach for hospital planning by using FLPs. This paper is probably the most relevant and transferable part of the literature to the problem of this thesis considering a hospital environment.

3.1.2 Objective Function

Allocating functions with specified sizes to locations of the hospital footprint has traits similar to a two-dimensional packing problem (Lodi et al., 2002). The packing problem seeks to allocate a set of objects of certain sizes and shapes into a limited area. Christensen et al. describe the geometric bin packing problem, consisting of a collection of rectangular items with a defined size. The model aims to pack all rectangles into a bin with a minimal number of unit squares. Christensen et al. describe an orthogonal packing case that avoids overlapping of items and to some extent takes into account the relative placement of the elements, which is an essential aspect of the FLP. The objective of the packing problem is to minimize waste of objects, by fitting as many objects as possible into the area defined. Unlike the packing problems, the internal distances between allocated functions are crucial for the HLPs. In addition, all the functions of the HLP need to fit inside the layout of the hospital, and "waste" is in many ways not an option since all functions of the hospital need to fit. On top of this, the objects, in this thesis referred to as functions, may not have a predefined shape, as is common in packing problems.

Kusiak and Heragu (1987) evaluated different approaches for modeling the FLP, including QAPs, Quadratic Set Covering Problems (QSP), Integer Linear Programming Problems (ILP) and Mixed Integer Linear Programming Problems (MILP). Despite differences in structure, a decision variable relating the placement of one function to another is common in all formulations. Although the QAP is considered by many to be the most reasonable and natural formulation of an FLP (Kusiak and Heragu, 1987), traits of the other approaches can be used as supplements to improve or simplify the overall formulation of a problem.

Koopmans and Beckmann (1957) was the first to formulate the facility layout problem of locating functions based on flows between them (Kusiak and Heragu, 1987). They modeled the FLP as a Quadratic Assignment Problem (QAP). The QAP belong to a class of combinatorial optimization problems (Burkard, 1984) and describes an assignment problem where the objective depends on the relation between the location of several elements. This gives the problem a second-degree objective function where the location of functions is interdependent, and the objective is dependent on products of pairs of binary variables. The works of Elshafei

(1977) and Helber et al. (2016) are examples of formulating HLPs with features of QAPs, and the advantages of this approach are seen through the ability of the QAP as an objective function to directly, or by including relatively small adjustments, capture the nature of different FLPs. However convenient, the formulation of the QAP includes a large number of binary variables connected through the placements of different elements, making the problem-size extensive and the computational time of the model possible massive (Kvillum and Vigerust, 2017). Also, a QAP in its general form assigns one element to one location binary, leaving little room for functions or locations of different sizes (Bazaraa, 1975).

The Quadratic Set Covering Problem (QSP) is defined as the problem of locating objects on a given area *(...) in such a way as to minimize the total interaction weighted by the distance* (Bazaraa, 1975). In contrast to the QAP, the QSP is here able to handle functions of varying size by dividing the available area in a grid and define the functions as a configuration of a number of the grid elements. In other words, the QSP opens for functions taking several parts of the location (grid elements) but does not accommodate several functions taking a share of one defined grid element. The QSP is an extension of the Set Covering problem (SCP). Murray et al. (2010) studied linear Location Set Covering problems (LSCP) and presented two versions of these, an implicit, and one explicit. Whereas the LSCP-Explicit model considers the facilities in a location by specific sets of facility combinations, while the LSCP-implicit model allows more than one function to cover each node/demand by a percentage of that node (Murray et al., 2010). Both functions covering several nodes and nodes containing fractions of several functions are highly relevant when considering hospital planning where neither locations or functions are of regular sizes and shapes, but these two traits have not been found combined in a quadratic problem in the literature.

Caused by the challenges of solving problems having the quadratic nature of the FLP, linearizing the structure has proven useful to be able to solve the problem using simplex-based methods. By connecting the binary variables of placement of pairs of functions, a new set of binary variables describing relations of placement between two functions is created, and the problem can be solved as an Integer Linear Programming Problem (ILP). This method was used by Helber et al. (2016) when modeling the HLP, formulating the objective as a QAP with linearization making it an ILP. The earlier mentioned desirable possibilities of a node to be shared between different functions each taking a percentage of the node, would, in turn, produce a MILP (Mixed Integer Linear Programming Problem) including both binary and continuous variables.

Most FLPs define the cost variable of the problem as transportation efforts (Helber et al., 2016), using registered data of transportation over time. This approach gives numerical values on the frequency and accordingly the induced importance of interactions between functions. Extending the understanding of transportation costs as a *distance based measure*, not only considering transportation of goods, Meller et al. (1998) interpret the cost variable in FLPs to include different flows. In a hospital, the interactions and flows are complex, and data on internal transportation can be limited or hard to obtain. This hampers the process of prioritizing and systematizing interactions between functions, and the need for an executive cost giving comparable values arises. In the preceding specialization project (Kvillum and Vigerust, 2017), the objective accounted for distances weighted by *proximity values*, values indicating the need for closeness between the functions placed.

3.1.3 Time Aspect

FLPs can be categorized as either dynamic or static by whether the problem incorporates changes over time or not. In a static problem, parameters such as flows between functions are stable over the planning horizon, while a dynamic problem must be flexible or at least take into account flows that may vary over time (Arnolds and Nickel, 2015). Drira et al. (2007) state that most work done on FLPs incorporates a static model. However, the dynamic approach is used in industrial application to develop robust layouts, exemplified by Balakrishnan et al. (2003), Braglia et al. (2003) and Kouvelis et al. (1992). Here, the flow is divided into different periods, and separate layouts for each period of the time horizon are made (Drira et al., 2007).

HLPs, in general, have dynamic aspects that tend to change over time such as patient basis, number of employees and needs for medical equipment. Even when faced with the need for static decisions, hospital planners aim to make hospitals flexible to variations in needs over time. Rooms or departments can be built to possibly accommodate multiple purposes over time, which is a method of including some degree of dynamics in the model (Sykehusbygg HF, 2017a). Also, seasonal, weekly and daily variations in patient basis may occur, often due to the location of the hospital regarding climate, and due to seasonal variations in diseases and damages (Sykehusbygg HF, 2017b). Hospitals are projected partly based on statistics and forecasts, attempting to predict needs of the hospital in the future. As rebuilding is costly, it is sought avoided by incorporating future changes as the flexibility of areas when planning the hospital (Sykehusbygg HF, 2017c). The primary goal of the HLP is to make cost-effective decisions today, while still retain as much flexibility in the layout as possible to keep the cost of future changes to a minimum. Hence, despite the dynamic aspects of the planning of a relatively permanent hospital building, the solution of the HLP itself, when executed, is static.

3.1.4 Representation of Areas

There are several ways of formulating and modeling the area of buildings and the area of the units to be placed in a building in FLPs. The problem is, as seen in the survey by Ahmadi et al. (2017), almost without exceptions discretized to some degree, either by dividing the building into areas or by giving the functions a certain shape and/or size.

For the FLP to locate functions in a building, the characteristics of the building need to be known. The representation of the footprint by its size, shape, and areas available for placement of functions are an essential part of the problem formulation. Discretization of the footprint is done by dividing the footprint into regular or irregular areas of equal or different shape and size. When discretizing a footprint, a grid of even division into grid elements is frequently used. This is a natural consequence of the simplifications induced by the facility often being defined by an x times y measure (Drira et al., 2007) and the transferred simplifications on the representation of functions to be placed. The approach of the footprint given as a grid is used in the specialization project (Kvillum and Vigerust, 2017), representing the building by a number of equal sized and shaped grid elements configured in the shape of the building's footprint.

Most of the literature suggests a building structure having a rectangular nature. However, in cases where the footprint has a shape inadequate for dividing into quadratic grid elements, or where this is not suited with the representation of functions explained below, a continuous approach with little or no division of the footprint is necessary (Shayan and Chittilappilly, 2004; Ahmadi and Jokar, 2016). Both discrete and continuous approaches induce possibilities and disadvantages when it comes to the shape of functions to be placed in the building, and opens up for different solution methods (Ahmadi et al., 2017). A discrete approach has a simpler formulation and can be easier to solve, but comes with certain simplifications regarding obtained solutions. A continuous approach can be argued to have abilities to capture cases closer to reality and being more flexible; however, they will usually require increased computational effort and are more complicated to formulate compared to the discrete approach.

Similarly to the footprint, Ahmadi et al. (2017) distinguish between continuous and discrete approaches to the representation of functions. According to Ahmadi et al. (2017), the choice of representation results from evaluating specifications of the functions of the problem; mainly their size and whether they have required shapes, in addition to considering traits of the footprint and the available space for placing functions. Described by Afrazeh et al. (2010) and Kaku et al. (1988), in a discrete representation, the functions' areas are defined as a number of grid-elements, and the functions' shapes are constituted by arranging the grid elements in different ways (Kvillum and Vigerust, 2017). Though quadratic shapes are the ones exploited in literature (Shayan and Chittilappilly, 2004), the continuous representation is described as giving the function's area independently of grid elements, the sizes and shapes of the functions may differ, and be decided by the designated placement in the building.

Afrazeh et al. (2010) describe the FLP as a discrete representation of function areas, omitting the challenges of different shapes and sizes of elements by having equally sized functions with identical shapes. Helber et al. (2016) constructed a model with functions of different sizes giving all the functions a rectangular shape. As for most hospitals, and accordingly HLPs, having equal sizes for all functions is not a possibility, and by this comes a need for deciding on the shape of the functions to be placed, based on their size and their final location in the building (Sykehusbygg HF, 2017a).

In the specialization project, a fully discrete approach was exploited. In the discrete option where both the footprint and the functions to be placed are divided into equal quadratic elements, the final shapes of the functions in the building are given as configurations of the grid elements or as rectangular, equally-shaped figures (Ahmadi and Jokar, 2016; Kvillum and Vigerust, 2017). In Figure 3.1 a layout resulting from a model of a hospital with the shape divided in a grid from the specialization project is shown. The groups of grid elements with the same color are functions placed on the grid with a configuration of elements based on size and suitable position by the model.

Both Ahmadi and Jokar (2016) and Shayan and Chittilappilly (2004) consider the footprint as a continuous area without division in smaller areas, limited only by the footprint's dimensions. The models presented in these works also treat all the functions as equally shaped rectangles, but with unequal size. Murray et al. (2010) discuss LSCP-Implicit (mentioned earlier in Section 3.1.2) that introduces the ability for multiple functions to cover a percentage of different areas (blocks) available for placement, representing a semi-continuous approach. Drira et al. (2007) indicate an increasing degree of complexity with increasing degree of continuity, as continuity opens for a more substantial number of more diverse solutions. Because of the complexity, a fully continuous model formulation is not likely to

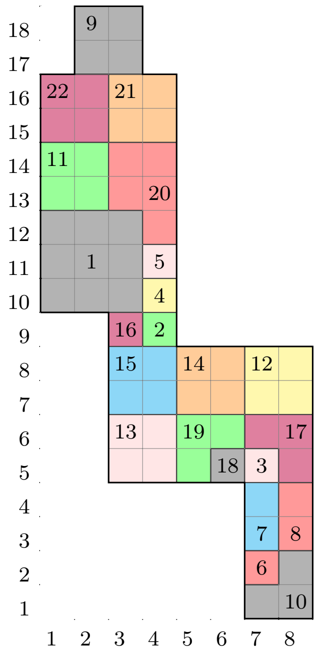


Figure 3.1: Discrete approach to footprint and functions

be expedient, but some degree of continuity could be sought to maintain possible good solutions (Ahmadi et al., 2017).

3.1.5 Number of Stages

Many facility layout problems are modeled with similarities to assignment problems (Elshafei, 1977; Helber et al., 2016). The assignment problem can be modeled in one or several stages, and a problem with more stages is preferred by many due to the simplifications it may induce the process of solving the FLP. Whether a problem consists of a single or multiple stages often correlates with whether the building has one or several floors (Che et al., 2017). As mentioned, the complexity of the FLP is expected to proliferate with the size of footprint and number of elements to locate (Farahani et al., 2012), and an extension from one to several floors has a similar effect on the problem's size and complexity. Though a multi-floor problem does not necessarily require a multi-stage solution method, the division of stages may in many instances follow the logic of several floors, where one of the stages assigns elements to floors, and the other takes care of the distribution within a floor. The FLP is a complex combinatorial problem, and thus separating the problem in stages can make it easier to solve.

In the work of Helber et al. (2016) on a hospital in Germany, functions are first assigned to separate floors, and in a second stage, an allocation process is performed for each floor in the building. Bernardi and Anjos (2013) perform a similar distinction of stages, where the mixed-integer linear program in the first stage takes into account the interaction of elements between floors, minimizing the global interaction/distance measures. The second stage optimizes the layout of each floor separately. Ahmadi and Jokar (2016) incorporate three stages, whereas the first stage assigns departments to floors with a MIP model, the second stage uses nonlinear programming model to specify the relative position of the departments on each floor, and finally, a third stage uses nonlinear programming to determine the final layout.

The division of the model in two stages in the work of Helber et al. (2016) and Bernardi and Anjos (2013) are both incorporated in the formulation of the problem, and as a simplification for the solution method. While the advantage is connected to reducing complexity, the disadvantage of this approach is the possible limitation of information used in the two stages. To simplify the problem, parts of the information are used in the first stage, and other parts are used in the second stage (Helber et al., 2016). The result is a lack of consideration of distances between units in different parts (floors) of the building in the internal assigning process of the second stage. This may not be a problem at all, but considering the application on hospitals, interactions between all units, in all different parts of buildings could be significant. This is especially important since the distance of traveling some floors with an elevator may be as efficient or even much quicker than moving from one side of the building to the other on the same floor.

When considering a building with several floors, the role of transportation between floors by elevators and/or stairs becomes applicable. In general, if solving the layout for each floor independently, only one location of an elevator/staircase will be accounted for. Because of the lack of information on relative placements of functions on other floors and therefore which elevator is the favorable elevator, the distance to locations on other floors can only be accounted for by a distance to the closest elevator and a distance incurred by taking the elevator (Bernardi and Anjos, 2013). To be able to account for a choice between several elevators, solving the different floors after the functions are assigned to a floor need to be somewhat interdependent. If solving for all floors simultaneously, the choice between several elevators when calculating distances between locations is easily accounted for, but solving in one stage would in many cases be, as mentioned, a massively comprehensive problem. A goal of the formulation of the problem and further the division in two stages would be to, to some degree, account for the relative placements of functions on other floors when locating functions internally on a floor in the second stage (Che et al., 2017).

3.1.6 Solution Methods

When deciding the solution method for solving the FLP, there is a trade-off between obtaining the optimal solution, which requires a significant amount of computational time and effort for problems above a certain size, and obtaining a solution that creates a layout that is *good enough*, which could be performed in a shorter amount of time. There are several methods suggested for solving FLPs, and most of the previous work, including the majority of the cases described in Ahmadi et al. (2017), as seen in Table 3.1, argue that applying heuristics to the problem is a convenient approach. The extensive use of heuristics results from the high complexity of the FLP when the problem reaches a certain size (Singh and Sharma, 2006).

Singh and Sharma (2006) classify heuristic algorithms in two groups, where *construction* algorithms create a solution from scratch, and improvement algorithms improve an initial solution. Also, heuristics can be classified in distance-based and adjacent-based improvement heuristics. The first, distance-based improvement heuristics, is highly applicable to the general formulation of FLPs, whereas the other, adjacent-based heuristics, only accounts for adjacent functions, and is therefore not applicable to the general formulation of the FLP (Singh and Sharma, 2006). In addition to heuristics for solving the FLP, meta-heuristics are commonly used to approximate solutions to FLPs of large scales.

Another simplification of the solution method, which is performed by many, is as mentioned in Section 3.1.5 to divide the problem into multiple stages, by first allocating functions to larger areas (e.g., buildings or floors) and then allocating the functions to the optimal locations internally in the areas, separately. As explained in Section 3.1.5, a division in stages can simplify the solution process, allowing for models with a significant amount of data (large buildings, several floors, many functions), but this can be at the expense of the quality of the solution and the possibility of reaching the optimal solution. However, as will be discussed later, in hospital planning the difference between an optimal and a solution close to optimal may be insignificant in terms of operating the hospital. Whether to implement two stages is a trade-off between the need for an optimal solution and the available computational time and computational capacity. For a real-life hospital with size significantly larger than the ones sufficient for testing, a full single-stage approach with an exact solution may not be possible.

When constructing a model from scratch, exact solution methods can be used for testing the model's characteristics, preferably in smaller instances. Also, using mathematical programming methods is attractive caused by the simple incorporation of aspects such as fixing functions and varying footprints of the building (Ahmadi and Jokar, 2016). The mathematical model of the specialization project (Kvillum and Vigerust, 2017) was solved in one stage using an exact solution method, incorporating all relations in that stage. However, this model consisted of one single floor. An extension of the model of the specialization project to several floors as preparatory work before developing the model of this thesis, caused much longer calculation times and a requirement for computational capacity gradually moving out of reach for available computational sources.

3.2 Classification of the work in this Thesis

In this section, the model of this master's thesis is positioned in relation to the above-presented classification framework and features of the problem. The related literature, including the preceding specialization project, and this thesis are shown in Table 3.1. The FLP described in this thesis concerns health care, more specifically hospital layout planning, making the problem an HLP. Ahmadi et al. (2017) pointed out that the health-care sector is one of the areas lacking adequate research in the context of mathematical optimization. The choices made in developing the model of this thesis is based on what is considered reasonable with the goal of achieving a thorough understanding of the problem. As the preceding literature-study showed, there are several possible ways of describing the same problem, the choices made in this thesis consist of what seems appropriate for the application. An attempt to provide plausible reasons for the choices made in relation to hospital applications are made, both to encourage a critical view of the model and to point to possible further improvements.

The goal of the model of this thesis is to incorporate transportation flows between pairs of functions that are to be placed in a hospital building. The problem is formulated as a mixed-integer linear program (MILP), and due to its objective function, the problem has similarities with the quadratic assignment problem, the QAP. The model assigns all functions to locations with the goal of minimizing the sum of distances between functions, multiplied by the relatedness (proximity) between pairs of functions. The model of this thesis acquires the QAP by placing the center of each function to one location, binary. The rest of the function can be placed in the same location, or acquire other, neighboring locations, similar to the QSP. Unlike the general formulation of both QAP and QSP, each location can contain fractions of several functions as continuous variables.

As a result of both the lack of accurate data of flows in the hospital and the methods Sykehusbygg uses when planning the building of a hospital, a measure different from actual flow data is used in the objective. The interactions and flows of patients, employees, and materials are emphasized in a priority value referred to as *proximity values* used to prioritize placement of functions in relation to each other. Moseley (1963) early on introduced a type of priority values obtained from traffic and transportation data, but the use of this approach has been limited in later years (Shayan and Chittilappilly, 2004). The proximity values of this thesis will in a similar way as transportation costs described in literature prioritize closeness of functions that have important relations.

Considering that the model of this thesis does not incorporate flows directly, but rather by proximity values based on today's interactions and a projection of future relations, planning the hospital incorporates a static nature as the possibilities for comprehensive changes are limited after the hospital is built.

A modeling approach that can be used regardless of the footprint of the building and associated design is desirable as hospitals are designed in many shapes, also non-rectangular (Rambøll Norge AS, 2018). Faced with buildings not easily divided into rectangular or quadratic blocks, the decision on the shapes of the functions to be placed may come as a result of the location assigned, rather than as an individual choice for each function. The area of each function to be placed does not necessarily match the area of an available location (dependent on how the division of the footprint is made). Such flexibility raises a need for a model having the earlier suggested possibilities of both allowing functions to be spread over several locations, and locations to be shared between parts of several functions.

The hospital building is in this thesis is represented by a set of nodes having a certain corresponding area capacity. The nodes represent the locations available for placing functions, and the area and shape available in each location differs throughout the building as a result of the building design. Distances between functions placed are calculated between the locations where the center of the function is placed. Unlike in the QSP, the locations of the building are not defined by a number of blocks, but rather as arbitrary shapes fitting the building, not defined by a grid. Even though not based on a grid, the division in locations of the building represents a discretization of the problem.

The functions to be placed in the model are represented by continuous areas, not by a number of grid elements. The shape of a placed function depends on the location(s) the function is designated to. For the area of a function to be spread across several nodes, the nodes need to be adjacent. The functions can cover only parts of a location, or cover multiple locations wholly or partly. Hence, while binary decision variables define which locations each function are located in or centered in, continuous variables represents the fraction of each node the function covers.

Different excerpts of literature describe aspects similar to some of the methods used to represent areas of locations and functions in the model of this thesis. However, the composition of modeling choices with footprint divided in locations that could hold more than one function, and functions that could be spread across different locations, along with the definition of groups of locations the functions can be spread across have not been found in the studied literature.

The model of this thesis incorporates buildings with several floors and several elevators, for each location calculating the shortest distances to all other locations, choosing the elevator giving the shortest distance between locations of different floors.

The HLP of this thesis is modeled in both one and two stages, the latter with the purpose of simplifying the solution approach and solving the model in reasonable time for larger instances. Both the first stage and the different sub-problems of solving each floor are solved with exact MILP software, making this a heuristic approach based on solving exact sub-problems. The purpose of the mathematical model introduced in this thesis is to work as a decision tool when planning the hospital. Because of this, a solution not optimal, but a result of some simplifications will often be sufficient as input in the planning process. The probability of the solution (whether optimality has been reached or not) being adjusted based on experience and other traits not captured in the mathematical data, is high. In addition to this, the model is based on proximity values defined from a basis of unsorted knowledge and experience, with a risk of being inconsistent, incorrect or inaccurate compared to flow data in less complex applications such as in industrial buildings.

The studied HLPs in the literature that incorporates two stages distributes functions to floors in stage 1 and then allocated functions within floors in stage 2 (Helber et al., 2016; Ahmadi et al., 2017). With this solution method, functions of a high requirement of proximity to each other are prioritized to be placed on the same floor, with consecutive little consideration given to functions on other floors in stage 2. The solution approach of this thesis exploits the two-stage approach and considers the relations to functions on other floors when optimizing each floor in the second stage. This ensures reasonable computation times while giving solutions that contemplate the relations between all functions of the hospital.

Table 3.1: Classification of relevant facility layout problems in literature

References	Environment	State	Objective Function	Areas	Floors	Stages	Solution Method	Relation Parameter
This report	**Hospital**	**Static**	**QAP**	**Continuous/ Discrete**	**Multiple**	**Two-stage**	Two-stage Decomposition	**Proximity values**
Afrazeh et al. (2010)	Industrial	Static/Dynamic	QAP	Discrete	–	–	–	Flows
Ahmadi and Jokar (2016)	General	Static	–	Continuous	–	Three-stage	–	–
Balakrishnan et al. (2003)	Industrial	Dynamic	QAP	–	–	–	Heuristics (Genetic algorithm)	Flows
Bernardi and Anjos (2013)	–	State	–	–	–	Two-stage	Mathematical Programming	–
Braglia et al. (2003)	Industrial	Dynamic	QAP	–	–	–	–	Flows
Elshafei (1977)	Hospital	Static	QAP	–	–	–	Heuristics	Flows
Helber et al. (2016)	Hospital	Static	QAP	Discrete	–	Two-stage	Heuristics (Fix-and-optimize)	–
Kaku et al. (1988)	General	Static	QAP	Discrete	–	–	Heuristics	Flows
Khosravian et. al. (2013)	Industrial	Dynamic	–	–	–	–	Heuristics (Simulated annealing)	Flows
Kouvelis et al. (1992)	Industrial	Static/Dynamic	–	–	–	–	Heuristics	Flows
Kvillum and Vigerust (2017)	Hospital	Static	QAP	Discrete	Single	Single	Exact methods	Proximity values
Meller et al. (1998)	General	–	–	–	–	–	–	Flows
Moseley (1963)	–	–	–	–	–	–	–	Priority values
Shayan and Chittilappilly (2004)	–	–	–	Continuous	–	–	Heuristics (Genetic algorithm)	Flows

Chapter 4

Problem Description

In the following, the hospital layout problem of this thesis is described, with important characteristics outlined. *Functions* with varying sizes, features, and relations to other functions are to be placed at different *locations* inside a hospital building. The hospital is assumed new, and not as a part of a reconstruction project of an existing building. Besides, the hospital can be of varying size and shape independent of any specific geometry, and consist of one or more independent buildings. The problem may easily be applied to several buildings, but in this thesis, the application is directed towards one building. The area and footprint of the building, along with the number of floors and layout of each floor can be determined in advance and is considered known to the model. This includes the location of corridors and hallways, atria, and elevators. The area requirement of each function to be placed and their need for *proximity* (closeness) to other functions are considered known. In addition to the mentioned specifications, any special requirements such as functions demanding a particular location, for instance near entrances, windows or a certain floor, are considered specified and must be taken into consideration when designing the model.

Each function interacts with several other functions, and the objective is to minimize the distances between functions that have a high need for proximity. In addition to minimizing distances, it is likely that there exists a placement cost for locating functions to different locations. However, for this thesis, lack of information on placement costs and focus directed towards minimizing distances with respect to relations are causing the placement cost to be assumed equal for all functions to any location and is hence neglected in the model.

The locations of the building where the functions are to be placed are predefined with an area. Each function should desirably be placed in one coherent unit. No functions can overlap in area, and no location can hold a sum of functions with larger area than the location's total area. All functions must be assigned to one or several locations, and in total, the whole area of each function must be assigned a location. The area of the footprint consists of both locations where functions can be placed and building structure and design such as hallways, corridors, atria, terraces, and windbreaks where functions cannot be placed. The area of functions in the problem is given as a gross value that includes area needed for shared facilities and support functions such as storage rooms, kitchens, and toilets, but excludes the hallways and corridors as these are assumed a part of the footprint.

Interactions between functions are summarized in a proximity value for each pair of functions. The objective to minimize the distances between functions is with respect to requirements for proximity. A high proximity value means that the pairs of functions have a high frequency and/or importance of interactions, or strong relations to one another. Where relevant, the proximity values between each pair of functions are given for different perspectives, representing, e.g., different stakeholders' opinions on the needs for interactions between functions. The perspectives can be assigned priorities by weighting the proximity values.

Chapter 5

Mathematical Model

In this chapter, the hospital layout problem is formulated mathematically. As mentioned, the model incorporates a quadratic assignment problem. This type of problem refers to the assignment of several functions to a set of locations simultaneously, accounting for their interactions with each other. In Section 5.1, the principles of how the model is constructed and special features of the mathematical problem are explained. Section 5.2 presents the sets, indices, parameters, and variables of the model. Sections 5.3 and 5.4 describe the objective function and the constraints of the model, respectively.

5.1 Modelling Principles

The assignment process of this model aims to make the interaction between hospital functions as efficiently as possible. This is done by assigning functions to locations, minimizing the distances between functions that require proximity. Special requirements, such as a need for placement at specific locations are taken into account when assigning functions. The area of the footprint of the hospital building is divided into several locations of different shapes and sizes that together constitute the overall footprint, exemplified in Figure 5.1. Each location is represented by a *node*, and when placed at a node, a function covers parts of or the whole corresponding area of that location. The area of the functions and the area of the locations corresponding to each node are pre-defined. Each function has an area independent of the division of areas of different locations. Hence, a function can be distributed over several different locations, or only cover parts of a location. The function's center is placed at the node of one location, and a function can cover one or more locations, partly or fully, including the center location. However, for a function to be allocated to more than one location, these locations need to be defined as a part of a set of adjacent *neighbour locations*, that is determined based on the layout. These neighbors are nodes that lie adjacent to, or close to the center location. This is to ensure that a function is not spread over a wide selection of different locations in the building, but is centralized in nearby areas. The output of the model consists of the nodal placement of each functions' centers, and the coverage each function has of this location and all other locations it is present in.

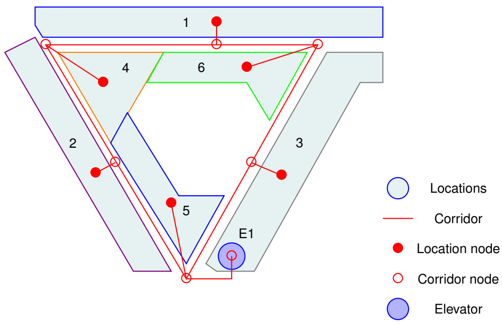


Figure 5.1: Representation of footprint with locations and nodes

Distances between locations are calculated between the center placement of pairs of functions and are presented as parameters in the model. The unit for distance is based on the footprint and corresponds to the actual distances in the building. Distances are in advance calculated between all pairs of nodes of the locations and are input to the model. When the model is executed, distances between pairs of functions are represented by the distances between the nodes where the center of the functions is placed. As shown in Figure 5.1 the distances are calculated from one location node to the corridor nodes and to the other location nodes. For locations on different floors, the distances are calculated the same way, but also through a node that represents the nearest elevator (and accounting for waiting time) with the distance corresponding to the distance units traveled. The appropriate elevator will be chosen when calculating the distances, and the distances are calculated as the shortest path. The distance measure is hence not divided in horizontal and vertical distance, but are pre-calculated based on the actual distances between all nodes. When placing the centers of functions, the rest of the function (if not all placed in the center node) is placed in neighbor locations of that center location. Corridor nodes and elevator nodes are only a part of the calculation of distances and are not further used in the model. If not limited by other parameters, functions are free to take any amount of a location, and the rest of the area of the location is available to other functions. Functions cannot overlap, and the total area of functions placed at a node is restricted by the area available in the node.

Each pair of functions is given a *proximity value* based on their degree of dependency, from considerations of flow, effectiveness, and interactions. The higher the value of proximity between two functions, the higher the closeness between them is prioritized in the assignment process. The proximity values are multiplied by the distance between pairs of functions when they are assigned, and thus the value of the objective function will increase if functions with strong relations are not placed close together. Different *perspectives* of the relations between functions give different proximity values, and the perspectives can be prioritized by weighting them in the objective function.

5.2 Notation

Below, the notation of the model is outlined. The notation is structured as follows: Sets have uppercase letters, and indices are presented with lowercase letters. Parameters have capital letters, and the variables have lower case letters. Subscripts denote indices to the sets, parameters, and variables. Superscripts of capital letters are used for specification of certain sets or parameters.

5.2.1 Sets and Indices

Table 5.1 summarizes the sets included in the model.

Table 5.1: Sets and indices of the model

Set	**Description**	**Indices**
$\mathcal{F}$	Functions	$f \in \mathcal{F}$
$\mathcal{N}$	Nodes (representing locations)	$i, j \in \mathcal{N}$
$\mathcal{N}_i^{NE}$	Nodes in the neighbourhood of node i	$j \in \mathcal{N}_i^{NE}$
$\mathcal{R}$	Perspectives of proximity values	$r \in \mathcal{R}$

The set $\mathcal{F}$ represents all the functions that need to be placed at the footprint of the hospital building. The nodes $\mathcal{N}$ represent areas in the building where functions can be placed. The nodes in $\mathcal{N}_i^{NE}$ are pre-determined as neighbour nodes to a node i. Perspectives $\mathcal{R}$ are used in the calculation of proximity values and represents different sets of proximity values dependent on the prioritization of proximity values between functions. Perspectives appear in the objective function to select target values.

5.2.2 Parameters

The parameters of the model are shown in Table 5.2.

Table 5.2: Parameters of the model

Parameter	**Description**
A_f	Area required for function f
Q_i	Area available in node i, included allowed margin
D_{ij}^N	Distance between node i and node j
α_r	Weight given to perspective r
P_{fgr}	Proximity of functions f and g for perspective r
R_i	Max. number of functions allowed placed in node i
C_i	Max. number of centers allowed placed in node i
M_f	Max. number of nodes function f can be spread over
δ	Min. fraction of a location a function needs to cover
γ	Min. fraction of a function allowed to cover a location

A_f represent the area of the functions that are to be placed in the hospital. The area corresponding to each node available for functions is represented by Q_i. The distance measure D_{ij}^N represents the distance between the nodes i and j. α_r are weights given to different perspectives of proximity values. Values of proximity, P_{fgr} are defined between function f and g in perspective r. R_i are limits on the number of functions that can be placed at each node, while C_i limits the number of function centers that can be placed in a node. M_f are limits on how many nodes each function f can be spread across. The parameter δ represents a lower bound on the fraction of a node i each function f needs to take, while γ represent a lower bound of the fraction of a function f allowed to cover a location.

5.2.3 Variables

The variables of the model are presented in Table 5.3.

Table 5.3: Variables of the model

Variable	Description
t_{fi}	Fraction of area of node i covered by function f
w_{fi}	$= 1$ if function f exists in node i
x_{fi}	$= 1$ if function f is centered at node i /=0 otherwise
y_{figj}	$= 1$ if function f has its center in node i and function g has its center in node j /=0 otherwise

t_{fi} describe the fraction of the area corresponding to the node a function covers. w_{fi} represent if a function f is placed in a node i, and x_{fi} describe the placement of the center of a function f in a node i. y_{figj} are relation variables between pairs of functions f and g, and takes on the value one if f is placed in i and g is placed in j.

5.3 Objective function

$$min\ z = \sum_{f\in\mathcal{F}} \sum_{g\in\mathcal{F}|f<g} \sum_{i\in\mathcal{N}} \sum_{j\in\mathcal{N}} (\sum_{r\in\mathcal{R}} \alpha_r P_{fgr}) D^N_{ij} y_{figj} \tag{5.1}$$

The objective function (5.1) aims to allocate functions $\mathcal{F}$ to locations represented by nodes i and j in $\mathcal{N}$. The goal is to minimize distances D^N_{ij} between the center of functions with respect to proximity values that are given for pairs of functions for different perspectives $\mathcal{R}$.

5.4 Constraints

Following are the constraints included in the model. The set $i \in \mathcal{N}$ represent the nodes that are available for placement of functions, in other words, the functions that are not yet locked to a node. Specifications of groups of nodes, characteristics that are relevant for placement of functions are preprocessed in the creation of the variables.

5.4.1 Assignment Constraints

Thee assignment constraints included in the model are presented below.

$$\sum_{i\in\mathcal{N}} x_{fi} = 1 \qquad f \in \mathcal{F} \tag{5.2}$$

$$x_{fi} - w_{fi} \leq 0 \qquad f \in \mathcal{F};\ i \in \mathcal{N} \tag{5.3}$$

$$x_{fi} + x_{gj} - y_{figj} \leq 1 \qquad f, g \in \mathcal{F};\ i, j \in \mathcal{N} \tag{5.4}$$

$$w_{fi} \leq x_{fi} + \sum_{j\in\mathcal{N}_i^{NE}} x_{fj} \qquad f \in \mathcal{F};\ i \in \mathcal{N} \tag{5.5}$$

$$\sum_{f\in\mathcal{F}} t_{fi} \leq 1 \qquad i \in \mathcal{N} \tag{5.6}$$

$$t_{fi} - w_{fi} \leq 0 \qquad f \in \mathcal{F};\ i \in \mathcal{N} \tag{5.7}$$

$$\sum_{i\in\mathcal{N}} Q_i t_{fi} \geq A_f \qquad f \in \mathcal{F} \tag{5.8}$$

$$t_{fi} - min\left\{1, \frac{\gamma A_f}{Q_i}\right\} x_{fi} \geq 0 \qquad f \in \mathcal{F};\ i \in \mathcal{N} \tag{5.9}$$

$$t_{fi} \geq \delta w_{fi} \qquad f \in \mathcal{F};\ i \in \mathcal{N} \tag{5.10}$$

$$C_i - \sum_{f\in\mathcal{F}} x_{fi} \geq 0 \qquad i \in \mathcal{N} \tag{5.11}$$

$$R_i - \sum_{f\in\mathcal{F}} w_{fi} \geq 0 \qquad i \in \mathcal{N} \tag{5.12}$$

$$M_f - \sum_{i\in\mathcal{N}} w_{fi} \geq 0 \qquad f \in \mathcal{F} \tag{5.13}$$

Constraints (5.2) make sure that all functions' centers are assigned to a node i. Constraints (5.3) relate the placement of a function in a node to the placement of a center so that a part of the function exists in the center node. Constraints (5.4) are linearization constraints that relate the allocation of pairs of functions to nodes, given that there is a relation between that pair of functions. Constraints (5.5) are neighbor constraints, and ensure that functions are somewhat connected when placed across several nodes. If a function occupies more than one node, all nodes occupied need to be either the center node or neighbor nodes to the center node. Distances between functions are calculated from the node chosen as the center, so spreading functions only to nodes that are close to the center ensures that the calculation of distances is quite accurate. The purpose of the neighbor constraint is to prevent a function to be placed in a widespread manner and cover nodes that are not neighbors.

Constraints (5.6) ensure that no more than 100% of the area that corresponds to a location is covered by the functions placed there. Constraints (5.7) give a relation between t_{fi} and w_{fi}, so that if a function takes a fraction of a node, w_{fi} is set to be 1. Constraints (5.8) make sure that for all nodes, the area placed in the nodes occupied is greater or equal to the area of the function. Constraints (5.9) ensure that if the center of a function f is placed in node i, the area of function f placed in node i is larger than or equal to the a minimum fraction γ of the area of that function. This is used to ensure that a large part of the function is placed in the center node, where the distances are calculated from, and hence make sure that the placement of that function does not cause divisions of the function on a widespread area.

Constraints (5.10) force the fraction t_{fi} to be greater or equal to a parameter δ if a function is placed at that node. This can be relevant to reduce the possibility of fractions of insignificant size to be places in nodes. Constraints (5.11), (5.12), and (5.13) are all restrictions on how functions f are allocated in nodes i. Constraints (5.11) restrict the number of functions with center in a node i by the parameter C_i. Constraints (5.12) ensure that that the maximum number of functions that can be present in each node does not exceed R_i and Constraints (5.13) ensure a restricted division of each function to different nodes by the parameter M_f.

5.4.2 Valid inequalities

$$\sum_{i \in \mathcal{N}} \sum_{j \in \mathcal{N}} y_{figj} = 1 \qquad f, g \in \mathcal{F} \tag{5.14}$$

$$x_{fi} - \sum_{j \in \mathcal{N}} y_{figj} = 0 \qquad f, g \in \mathcal{F};\ i \in \mathcal{N} \tag{5.15}$$

Valid inequalities are incorporated in the model to make the formulation tighter. Constraints (5.14) are added to the formulation of the model to reduce the size of the problem in a similar way as a valid inequality. The sum of y_{figj} for each pair of functions must equal 1, and therefore only one variable is created for each pair of functions. This is given that the pair of functions has a relation, hence the proximity value P_{fgr} is larger than zero. Constraints (5.15) ensure that only one relation is created between each pair of functions. For the constraints to exist, function f needs to be placed in location i, function g needs to be placed in location j, and the functions need to have a relation.

5.4.3 Variable definitions

$$x_{fi} \in \{0,1\} \qquad f \in \mathcal{F};\ i \in \mathcal{N} \tag{5.16}$$

$$w_{fi} \in \{0,1\} \qquad f \in \mathcal{F};\ i \in \mathcal{N} \tag{5.17}$$

$$y_{figj} \in \{0,1\} \qquad f,g \in \mathcal{F} \mid f < g;\ i,j \in \mathcal{N} \mid i \neq j \tag{5.18}$$

$$t_{fi} \geq 0 \qquad f \in \mathcal{F};\ i \in \mathcal{N} \tag{5.19}$$

Constraints (5.16), (5.17) and (5.18) are binary constraints. In the mathematical model, y_{figj} are given as binary variables while relaxations of y_{figj} are used in the implementation of the model, reducing the complexity of the model, but giving equivalent results as the mathematical model, as the values of y_{figj} is given by the binary x_{fi}. Constraints (5.19) are definitions of the fraction variables t_{fi} limiting the lower value of the variable.

Chapter 6

Solution Method

Caused by the size and complexity of the problem of this thesis, introducing some form of decomposition seems inevitable in trying to obtain a good solution. With the building(s) of a hospital naturally divided in sections by the floors, an initial assignment process for distributing functions to different floors seems reasonable. As shown in Section 3, the literature points to this division as a relevant solution method. The model of this thesis is divided into two stages where the first stage assigns functions to floors, and the second stage distributes the functions to specific locations of their respective floor. Section 6.1 presents the notation of the two stages. In Section 6.2 and 6.3, respectively, the formulation of the two stages is presented.

Stage 1 of the model allocates functions to floors based on their overall relation to other functions. The goal is to locate functions with high degree of interaction (expressed by proximity value) on the same floor. Stage 1 does not handle the internal placements of functions on each floor, but avoids exceeding the allowed area of functions placed in each floor and making sure the set of functions are feasible to place there in regards of the areas and neighbor nodes on the floor. This stage does not account for distances between each location on the separate floors, but rather the distances between floors, making it expedient, from the objective point of view, to allocate functions of high proximity to the same floor.

In the second stage of the model, the internal allocation process of each floor is handled. This stage is similar to the original model presented in Section 5, but differs in the way of accounting for functions on other floors. While a one-stage model can be used for handling all relations between the functions placed simultaneously, the second stage has the option of being solved iteratively. In that

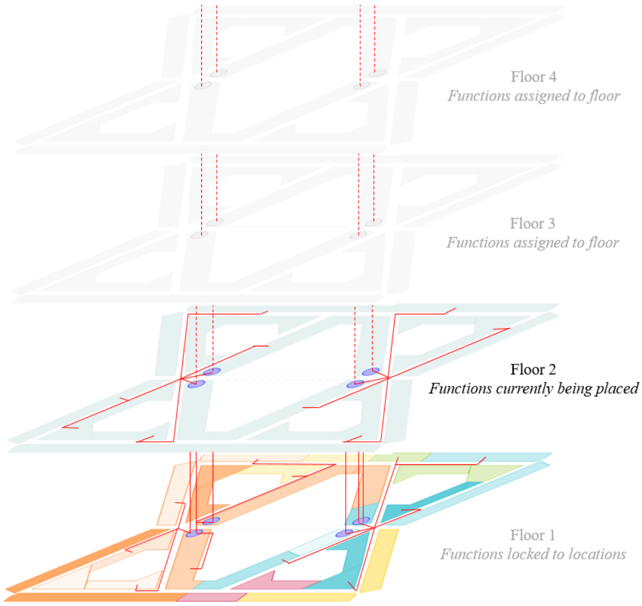


Figure 6.1: Method of iteratively accounting for functions on other floors

case, one floor is solved at the time. For each floor, the functions are allocated to locations considering their relationship and interaction to other functions by their proximity value.

Stage 2 distributes functions over one floor at a time, accounting for functions already placed at other floors in previous iterations. In addition, the floors not yet solved are included by accounting for the relation the functions on the floor of the iteration has to the functions assigned to those floors by stage 1 of the solution process. The order of the floors solved has to be chosen from considerations of convenience, logistics and a prioritization of the importance of different measures.

For a solution method solving the floors in order from bottom to top, the process of solving the second floor is described below and illustrated in Figure 6.1. Here, floor 2 is solved based on both the internal information on the floor and information about functions on other floors, both the floors already solved and the floors not yet decided. The placement of the functions on the floor in question is a result of the relations between pairs of functions on that floor, relations to functions that are locked to the previously solved floors, and aggregated relations to functions assigned to floors, but not yet placed throughout the floor. The functions placed on the previously solved floors (here floor 1) are accounted for with distances directly from their specified placement. Caused by the unknown layout of placements on floors that are not already decided; the functions on other floors (here higher in the building than the floor in question, floor 2) are collected in shared area nodes resembling whole floors, and the relations to functions in the floor in question (floor 2) are calculated to that node, through the nearest elevator.

6.1 Notation

Below, the notation for both stages is presented. The affiliation to stages is outlined for both the common notation and the notation specific for each stage. The notation is structured as in the preceding mathematical model. Significant shares of the notation are equal to the original model, as this model is closely connected to stage 2, and only the notation unique to the two-stage structure is outlined and explained.

6.1.1 Sets and indices of the solution method

Table 6.1 summarizes the sets included in the two-stage solution method.

Table 6.1: Sets and Indices of the solution method

Set	Description	Indices	Stage
$\boldsymbol{\mathcal{E}}$	**Floors of the building**	$\boldsymbol{e, d \in \mathcal{E}}$	**1 & 2**
$\mathcal{F}$	Functions	$f \in \mathcal{F}$	1
$\boldsymbol{\mathcal{F}_e^E}$	**Functions located on floor** $\boldsymbol{e}$	$\boldsymbol{f \in \mathcal{F}^E}$	**2**
$\mathcal{N}$	Nodes	$i, j \in \mathcal{N}$	1
$\boldsymbol{\mathcal{N}_e^E}$	**Nodes located on floor** $\boldsymbol{e}$	$\boldsymbol{i \in \mathcal{N}_e^E}$	**2**
$\mathcal{N}_i^{NE}$	Nodes in the neighbourhood of node i	$j \in \mathcal{N}_i^{NE}$	1 & 2
$\mathcal{R}$	Perspectives of proximity values	$r \in \mathcal{R}$	1 & 2

The set $\mathcal{E}$ consists of the floors of the building. $\mathcal{F}^E$ represent all the functions placed in floor e by stage 1. Similarly, $\mathcal{N}_e^E$ consist of the nodes located on floor e of the building. Both these sets are results from stage 1 of the model, and are a part of stage 2 of the solution method.

6.1.2 Parameters

The parameters used in the two-stage approach, both in stage 1 and stage 2, are shown in Table 6.2.

Table 6.2: Parameters of the solution method

Parameter	Description	Stage
A_f	Area required for function f	1 & 2
α_r	Weight given to perspective r	1 & 2
C_i	Max. number of centers of functions f allowed in node i	2
D^E_{ed}	Distance between floor e and floor d	1
D^N_{ij}	Distance between node i and node j	1 & 2
δ	Min. fraction of a location a function needs to cover	1 & 2
γ	Min. fraction of a function allowed to cover a location	1 & 2
M_f	Max. number of nodes function f can be spread across	2
P_{fgr}	Proximity of functions f and g for perspective r	1 & 2
$\boldsymbol{B_{fir}}$	**Aggregated proximity value for perspective $\boldsymbol{r}$ for function $\boldsymbol{f}$ in $\boldsymbol{\mathcal{F}^E_e}$ to functions on other floors**	**2**
Q_i	Area available in node i	1 & 2
R_i	Max. number of functions allowed placed in node i	2

Stage 1 is added with the purpose of dividing functions over floors, and as functions are not yet assigned to a location distance to/from node to an elevator is not included in this stage, the only distance relevant in this stage is the vertical distance between floors. The parameter D^F_{ed} is added as the distance between floor e and d. In stage 1, P_{fgr} are the proximity values between all functions of the problem. In stage 2, P_{fgr} are the proximity values between all functions that are placed at floor e. B_{fir} are aggregated proximity values used in stage 2 to functions in other floors. For each function f to be placed at node i at floor e, B_{fir} consist of proximity values times distance to functions on other floors than the floor executed in stage 2.

6.1.3 Variables

The variables of the solution method are presented in Table 6.3. As with sets and parameters, the differences from the original mathematical model are the variables added in association with the two-stage approach.

Table 6.3: Variables of the solution method

Variable	Description	Stage
$\boldsymbol{p_{fe}}$	**= 1 if function $\boldsymbol{f}$ is placed on floor $\boldsymbol{e}$**	**1**
t_{fi}	Fraction of area of node i covered by function f	2
$\boldsymbol{v_{fegd}}$	**= 1 if function $\boldsymbol{f}$ is placed on floor $\boldsymbol{e}$ and function $\boldsymbol{g}$ is placed on floor $\boldsymbol{d}$ /=0 otherwise**	**1**
$\boldsymbol{w_{fi}}$	**= 1 if function $\boldsymbol{f}$ exists in node $\boldsymbol{i}$ /=0 otherwise**	**1 &2**
$\boldsymbol{x_{fi}}$	**= 1 if function $\boldsymbol{f}$ is centered at node $\boldsymbol{i}$ /=0 otherwise**	**1 & 2**
y_{figj}	= 1 if function f has its center in node i and function g has its center in node j /=0 otherwise	2

p_{fe} are variables indicating which floor e the function f is placed on. The variables v_{fegd} are floor relations, taking the value 1 if function f is placed on floor e and function g is placed on floor d. w_{fi} and x_{fi} are relaxed in the implementation of stage 1, while they are binary as a part of stage 2.

6.2 Stage 1

The goal of both stages of the model combined is to allocate functions considering their interactions and relations expediently. Stage 1 determines the distribution of functions on different floors, collecting as many functions with important relations on each floor as possible. Even though stage 1 distributes the functions on different floors, consideration is given to the area and features of the locations on each floor. The model ensures that the distribution takes into account the specifications of the nodes and functions, making sure that the total function area allocated to each floor does not exceed the total area of the locations.

6.2.1 Objective function

$$min\ z = \sum_{f \in \mathcal{F}_e^E} \sum_{g \in \mathcal{F}_e^E f<g} \sum_{e \in \mathcal{E}} \sum_{d \in \mathcal{E}} (\sum_{r \in \mathcal{R}} \alpha_r P_{fgr}) D_{ed}^E v_{fegd} \tag{6.1}$$

The objective function of stage 1 (6.1) aims to allocate all functions $\mathcal{F}$ to floors e and d in $\mathcal{E}$. The goal is to minimize distances D_{ed} between the floors where functions are places with respect to proximity values that are given for pairs of functions for different perspectives $\mathcal{R}$. Stage 1 allocates functions only to floors and does not account for internal distances in a floor.

6.2.2 Assignment Constraints

The assignment constraints included in the first stage of the model are shown below.

$$\sum_{i \in \mathcal{N}} x_{fi} = 1 \qquad f \in \mathcal{F} \tag{6.2}$$

$$x_{fi} - w_{fi} \leq 0 \qquad f \in \mathcal{F};\ i \in \mathcal{N} \tag{6.3}$$

$$w_{fi} \leq x_{fi} + \sum_{j \in \mathcal{N}_{NE_i} x_{fj}} \qquad f \in \mathcal{F};\ i \in \mathcal{N} \tag{6.4}$$

$$\sum_{f \in \mathcal{F}} t_{fi} \leq 1 \qquad i \in \mathcal{N} \tag{6.5}$$

$$\sum_{i \in \mathcal{N}^e} \sum_{f \in F} A_f x_{fi} \leq \sum_{i \in \mathcal{N}^e} Q_i \qquad e \in \mathcal{E} \tag{6.6}$$

$$p_{fe} + p_{gd} - v_{fegd} \leq 1 \qquad f, g \in \mathcal{F};\ e, d \in \mathcal{E} \tag{6.7}$$

$$\sum_{e \in E} p_{fe} = 1 \qquad f \in \mathcal{F} \tag{6.8}$$

$$p_{fe} \geq \sum_{i \in \mathcal{N}_e^E} x_{fi} \qquad f \in \mathcal{F};\ e \in \mathcal{E} \tag{6.9}$$

Constraints (6.2) make sure that all functions' centers are assigned to a node. Constraints (6.3) relate the placement of the center of a function with the placement variable x_{fi}, to ensure that a part of the function exists in the center node. Constraints (6.4) are neighbour constraints. If a function occupies several nodes, they all need to be either the center node or neighbor nodes to the center node. Constraints (6.5) ensure that no more than 100% of the area that corresponds to a node is covered by the functions placed there. Constraints (6.6) limit the total area of functions placed on a floor to the total area of the nodes on this floor.

Constraints (6.7) are linearization constraints that relate the allocation of pairs of functions on floors to each other. Constraints (6.8) ensure all functions are allocated to one floor. Constraints (6.9) relate the floor variable to the placement of centers, and forces p_{fe} to be one if a function f is center placed in a node i on floor e.

6.2.3 Valid inequalities

$$\sum_{e \in \mathcal{E}} \sum_{d \in \mathcal{E}} v_{fegd} = 1 \qquad f, g \in \mathcal{F} \tag{6.10}$$

Valid inequalities can be incorporated into the model to make the formulation tighter. Constraints (6.10) give the sum of v_{fegd} for each pair of functions equal to 1, and therefore only one relation variable is created for each pair of functions on different floors.

6.2.4 Variable definitions

$$v_{fegd} \in \{0,1\} \qquad f,g \in \mathcal{F} \mid f < g;\ e,d \in \mathcal{E} \mid e \neq d \tag{6.11}$$

$$p_{fe} \in \{0,1\} \qquad f \in \mathcal{F};\ e \in \mathcal{E} \tag{6.12}$$

$$x_{fi} \geq 0 \qquad f \in \mathcal{F};\ i \in \mathcal{N} \tag{6.13}$$

$$w_{fi} \geq 0 \qquad f \in \mathcal{F};\ i \in \mathcal{N} \tag{6.14}$$

$$t_{fi} \geq 0 \qquad f \in \mathcal{F};\ i \in \mathcal{N} \tag{6.15}$$

Constraints (6.11) and (6.12) are binary constraints. Constraints (6.13), (6.14) and (6.15) are defined to be positive or zero. As stage 1 does not account for internal placements on each individual floor, the variables x_{fi} are relaxed, removing the binary demands. By still forcing the sum of x_{fi} to be 1 for a function on a floor, the allocation of functions to floors is ensured, and each function is assigned to one floor.

In the mathematical model, v_{fegd} are given as binary variables. On the contrary, a relaxation of v_{fegd} is used in the implementation of stage 1, allowing the variables to be continuous. The reason is the way each function are prevented from spreading across different floors by the definition of neighbors restricted to the same floors, and because v_{fegd} is given by p_{fe}. The relaxation reduces the complexity of the model, and by using constraints to control the sum of the variables for each function, the right functionality of the model is still ensured.

6.3 Stage 2

Stage 2 of the model considers the internal allocation of functions on a floor. This stage is executed when stage 1 is already performed, and the allocation of functions to different floors is done. The solution process is designed to be iterative, solving one floor at a time, while using the information from already solved floors. Stage 2 could also be used to solve each floor simultaneously, only taking into consideration which functions are allocated to the other floors while solving each floor. Relations to functions on other floors are accounted for either directly between functions for functions already assigned to a specific location on a floor, or by distances to their designated floor for functions not already given a specific placement on a floor. Following, stage 2 of the model is presented, as a model for the nodes of each floor $i \in \mathcal{N}_e^E$ with the functions $f \in \mathcal{F}_e^E$ placed there.

6.3.1 Objective function

$$min\ z = \sum_{f \in \mathcal{F}_e^E} \sum_{i \in \mathcal{N}_e^E} \sum_{\substack{g \in \mathcal{F}_e^E \\ f<g}} \sum_{j \in \mathcal{N}_e^E} (\sum_{r \in \mathcal{R}} \alpha_r P_{fgr}) D_{ij}^N y_{figj} + \sum_{f \in \mathcal{F}_e^E} \sum_{i \in \mathcal{N}_e^E} (\sum_{r \in \mathcal{R}} \alpha_r B_{fir}) x_{fi} \tag{6.16}$$

The objective function (6.16) of stage 2 aims to allocate functions $\mathcal{F}_e^E$ placed on each floor to locations represented by nodes i and j in $\mathcal{N}_e^E$ on floor e. The goal is to minimize distances between the center of functions with respect to proximity values that are given for pairs of functions for different perspectives $\mathcal{R}$. In addition to accounting for the distances and proximity values of functions on the same floor or functions already placed, distances to function allocated to other floors but not yet placed are included with aggregated proximity values to each function to be placed as B_{fir}. These values contribute to prioritizing which functions need to lie close to the elevator on the floor in question.

6.3.2 Assignment Constraints

Following are the constraints included in the second stage of the model. The set $i \in \mathcal{N}_e^E$ represent the nodes that are available for placement of functions on floor e, after functions locked to nodes are placed to their respective locations. Specifications of nodes, like for instance window locations, are pre-processed in the creation of the variables. For each iteration (floor of the building) solved, the input data is modified by defining locations and distances customized to the floor in question.

$$\sum_{i \in \mathcal{N}_e^E} x_{fi} = 1 \qquad f \in \mathcal{F}_e^E \tag{6.17}$$

$$x_{fi} - w_{fi} \leq 0 \qquad f \in \mathcal{F}_e^E;\ i \in \mathcal{N}_e^E \tag{6.18}$$

$$x_{fi} + x_{gj} - y_{figj} \leq 1 \qquad f, g \in \mathcal{F}_e^E;\ i, j \in \mathcal{N}_e^E \tag{6.19}$$

$$w_{fi} \leq x_{fi} + \sum_{j \in \mathcal{N}_i^{NE}} x_{fj} \qquad f \in \mathcal{F}_e^E;\ i \in \mathcal{N}_e^E \tag{6.20}$$

$$\sum_{f \in \mathcal{F}_e^E} t_{fi} \leq 1 \qquad i \in \mathcal{N}_e^E \tag{6.21}$$

$$t_{fi} - w_{fi} \leq 0 \qquad f \in \mathcal{F}_e^E;\ i \in \mathcal{N}_e^E \tag{6.22}$$

$$\sum_{i \in \mathcal{N}_e^E} Q_i t_{fi} \geq A_f \qquad f \in \mathcal{F}_e^E \tag{6.23}$$

$$t_{fi} - min\left\{1, \frac{\gamma A_f}{Q_i}\right\} x_{fi} \geq 0 \qquad f \in \mathcal{F}_e^E;\ i \in \mathcal{N}_e^E \tag{6.24}$$

$$t_{fi} \geq \delta w_{fi} \qquad f \in \mathcal{F}_e^E;\ i \in \mathcal{N}_e^E \tag{6.25}$$

$$C_i - \sum_{f \in \mathcal{F}_e^E} x_{fi} \geq 0 \qquad i \in \mathcal{N}_e^E \tag{6.26}$$

$$R_i - \sum_{f \in \mathcal{F}_e^E} w_{fi} \geq 0 \qquad i \in \mathcal{N}_e^E \tag{6.27}$$

$$M_f - \sum_{i \in \mathcal{N}_e^E} w_{fi} \geq 0 \qquad f \in \mathcal{F}_e^E \tag{6.28}$$

$$\sum_{e \in E} p_{fe} = 1 \qquad f \in \mathcal{F}_e^E \tag{6.29}$$

$$p_{fe} - \sum_{i \in \mathcal{N}_e^E} x_{fi} \geq 0 \qquad f \in \mathcal{F}_e^E;\ e \in E \tag{6.30}$$

Constraints (6.17) make sure that all functions' (F_e^E) centers are assigned to a node i. Constraints (6.18) relate the placement of a function in a node to the placement of a center, so that a part of the function exists in the center node. Constraints (6.19) are linearization constraints that relate the allocation of pairs of functions to nodes, given that there is a relation between that pair of functions. In the implementation of the model, the y_{figj}-variables are only created if there is a relation between f and g. Constraints (6.20) are neighbor constraints, and ensure that functions are connected when placed across several nodes. If a function occupies different nodes, all nodes occupied need to be either the center node or neighbor nodes to the center node. Distances between functions are calculated from the node chosen as the center, so spreading functions only to nodes that are close to the center ensures that the calculation of distances is quite accurate.

Constraints (6.21) ensure that no more than 100% of the area that corresponds to a node is covered by the functions placed there, and constraints (6.22) give a relation between t_{fi} and w_{fi}, so that if a function takes a fraction of a node, w_{fi} is set to be 1. Constraints (6.23) make sure that for all nodes, the area placed in different nodes is equal to the area of the function.

Constraints (6.24) ensure that if the center of a function f is placed in node i, the area of function f placed in node i is larger than or equal to the a minimum fraction γ of the area of that function. Constraints (6.25) force the fraction t_{fi} to be greater or equal to a parameter δ if a function is placed at that node. Constraints (6.26), (6.27), and (6.28) are all restrictions on how functions f are allocated in different nodes i. Constraints (6.26) restrict the number of functions with center in a node i by the parameter C_i. Constraints (6.27) ensure that that the maximum number of functions that can be placed at each node does not exceed R_i and Constraints (6.28) ensure a restricted division of each function to different nodes by the parameter M_f. Constraints (6.29) and (6.30) connects the floor variable to the center placement of functions on the designated floors.

6.3.3 Valid inequalities

$$\sum_{i \in \mathcal{N}_e^E} \sum_{j \in \mathcal{N}_e^E} y_{figj} = 1 \qquad f, g \in \mathcal{F}_e^E \tag{6.31}$$

$$x_{fi} - \sum_{j \in \mathcal{N}_e^E} y_{figj} = 0 \qquad f, g \in \mathcal{F}_e^E;\ i \in \mathcal{N}_e^E \tag{6.32}$$

As in stage 1, valid inequalities is incorporated into the model to make the formulation tighter. Constraints (6.31) are added to the formulation of the model to reduce the size of the problem similarly as a valid inequality. The sum of y_{figj} for each pair of functions must equal 1, and therefore only one variable is created for each pair of functions. Constraints (6.32) ensure that only one relation is created between each pair of functions.

6.3.4 Variable definitions

$$x_{fi} \in \{0,1\} \qquad f \in \mathcal{F}_e^E;\ i \in \mathcal{N}_e^E \tag{6.33}$$

$$w_{fi} \in \{0,1\} \qquad f \in \mathcal{F}_e^E;\ i \in \mathcal{N}_e^E \tag{6.34}$$

$$y_{figj} \in \{0,1\} \qquad f, g \in \mathcal{F}_e^E \mid f < g;\ i, j \in \mathcal{N}_e^E \mid i \neq j \tag{6.35}$$

$$t_{fi} \geq 0 \qquad f \in \mathcal{F}_e^E;\ i \in \mathcal{N}_e^E \tag{6.36}$$

Constraints (6.33), (6.34) and (6.35) are binary constraints. In the mathematical model, y_{figj} are given as binary variables while relaxations of y_{figj} are used in the implementation of the model, reducing the complexity of the model, but giving equivalent results as the mathematical model, as the values of y_{figj} is given by the binary x_{fi}. Constraints (6.36) are definitions of the fraction variables t_{fi} limiting the upper value of the variable.

Chapter 7

Computational Study

The Hospital Facility Layout Problem is implemented in the commercial software FICO® Xpress Optimization Suite 8.3, Xpress IVE version 1.24.18, Xpress optimizer version 31.01.09 and Mosel version 4.6.0. All instances in the computational study and the following Case Study in Section 8 are solved using nodes in the "Solstorm" cluster at NTNU with operating system CentOS 6.8 with Intel E5 (3.40GHz) and 512GB RAM.

In Section 7.1, aspects of the implementation are discussed, including an explanation of choices regarding special characteristics of locations and functions. In Section 7.2, a technical study is executed on the model to discuss features and characteristics of the implementation of the model, and the chosen solution method. Section 7.3 summarizes the impact of different aspects found in the technical study.

7.1 Implementation of the mathematical model

This section considers the implementation of the mathematical model presented in Chapter 5, with important aspects of the chosen implementation provided. The input needed for the model is presented in Section 7.1.1, and in the following Sections further explanations of the locations and functions, and the calculation of distances are given.

7.1.1 Data Description

Table 7.1 displays the content of the data sets used in the implementation of the model. Essential information like area and characteristics of the building design and the functions that are to be placed, along with distances between different parts of the building are presented in the data sets. A division of the footprint into locations of specified sizes, and limitations on the capacity of how many functions each location can house, in addition to a limit on how many, and which locations each function can be spread across when placed are included. Values regarding the need for relatedness between functions are defined on a scale from zero to ten (ten indicating a high need for relatedness) and work as the driver of the objective of the model. The proximity values are symmetric for each pair of functions; hence the relationship is equal in both directions. Several sets of proximity values can be included and weighted with different emphasis. Additional functionality like specifications on where some functions need to be placed, both related to floor and position on each floor are included and can be used to force some aspects of the solution.

Table 7.1: Content of the data sets

Set	**Content**
Building	Shape of footprint and floor plans, total area of the building
Locations	Associated nodes, locations available for placement, neighbour nodes capacity of functions/centers of functions, increase in area of locations
Distances	All-to-all distances between each location (represented by a node) through corridors and elevators
Functions	Required area, number of nodes the function is allowed to spread across
Specifications	Locked Placements, special requirements for locations the functions are to be placed in (e.g., windows)
Proximity	Proximity values between pairs of functions Weights on perspectives of proximity

7.1.2 Locations

The building is divided into several locations, all represented by a node indicating the center, as shown in Figure 7.1. The locations are distributed throughout several floors, and each location has pre-defined areas. Due to the puzzle of placing functions of different sizes throughout floors with certain areas, an area increase parameter of the area of locations is introduced. The area increase allows each location to be a tiny bit larger than the actual locations, introducing the needed flexibility to be solvable. Each location has pre-defined sets of neighbor nodes,

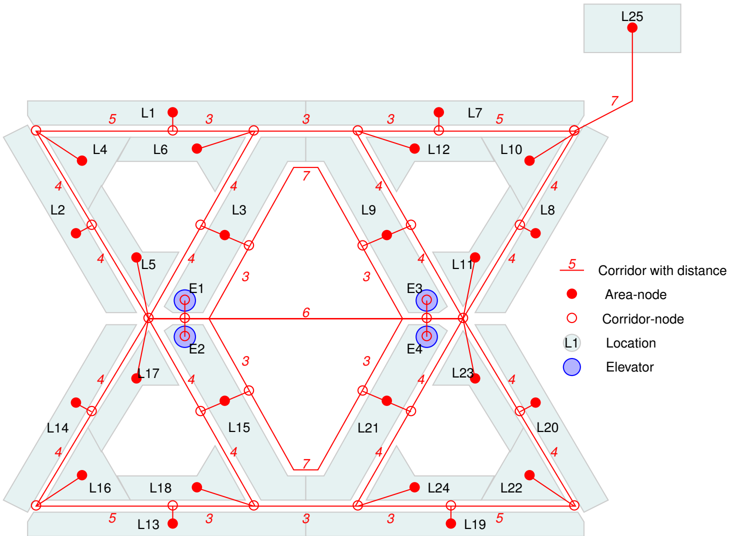


Figure 7.1: Footprint with locations and distances

listing locations that are adjacent and hence can be occupied by the same functions. Some locations are included in sets that have specific characteristics, for example, locations with windows, and locations on different floors. The purpose is to enable functions to match the specifications on the locations. The number of functions that can share the same location is limited for each location *(MaxFunctionsCapacity)*, and a restriction on the number of functions centers is induced on each location (*MaxCentresCapacity*).

7.1.3 Functions

Functions that are to be placed inside the building all have a specified area and can be placed in one or several locations, limited by the parameter *MaxSpreadFunctions*. These parameters ensure that functions are not divided into many small fractions throughout the hospital, but rather in a compact space so that the locations are connected, even though placed in separate locations. Functions can have specific criteria matching the characteristics of the location, which means they are limited to be placed in one of those nodes. For example, some functions can be required to be placed on the ground floor, or in a location that has a window. Besides, functions can be locked to locations, by either locking the center to a location or locking the presence of the function in that location, or by defining a minimum coverage fraction for a function in a location.

7.1.4 Distances

As shown in Figure 7.1, all locations are connected through a web of corridor nodes. All distances between adjacent nodes are defined and showed in the figure, and the distances between all the location nodes are calculated with an all-to-all shortest path algorithm (Floyd-Warshall) using MATLAB (R2016b, 64-bit). Each location has distance one to the nearest corridor node.

In a hospital building, both horizontal distances on each floor and vertical distances between floors have to be accounted for. Horizontal distances are given in a unit where the proportions are equivalent to the real distances. Between floors, elevators are the main link. The building may include stairs in different locations than the elevators, but as a simplification, distances are calculated through the elevators only, since some flows cannot go through the stairs while all flows of the hospital can travel by elevator. The distances between floors consist of one distance unit per floor traveled, in addition to two extra distance units accounting for the expected waiting time by the elevator. These penalty units are only added once, independent of the number of floors traveled.

The "cost" of waiting for an elevator is meant as a tool for weighting the distances of locations on different floors conveniently compared to horizontal distances on each floor. Initial studies of the model are done on varying this value, for an instance with three floors and one quadrant, the results showing that the final layout is independent of small variations of this measure.

7.2 Technical Study

This section presents test instances showing different characteristics with the purpose of illustrating the impact of varying features of the model. As several adjustments available in the implementation are relevant when using the model on a full-scale hospital case, the impact of different modifications is highly relevant. Table 7.2 summarizes the instances of the technical study. The modifications tested include varying the footprint of the hospital and the number of floors, inducing different locked functions, varying the capacity of locations and spread of functions and varying the definitions of neighbors. Tests are performed on giving varying weights to different perspectives on proximity and varying the allowed area increase. Lastly, valuable tests regarding solution method are performed.

The technical study is initiated by a *Base Case*, designed with basic characteristics that are described in Section 7.2.2. Throughout the tests, the other instances are

Table 7.2: Summary of test instances in the technical study

Instance	Footprint F/Q	Locked	Capacity /Spread	Neigh-bours	Pers-pective	Area increase	Stages	Section
Base Case	**1/3**	**-**	**2(3)/2(3)**	**Normal**	**P1**	**1%**	**1S**	7.2.2
1F1Q	1/1	-	2(3)/2(3)	Normal	P1	1%	1S	7.2.3
1F2Q	1/2	-	2(3)/2(3)	Normal	P1	1%	1S	”
1F4Q	1/4	-	2(3)/2(3)	Normal	P1	1%	1S	”
2F1Q	2/1	-	2(3)/2(3)	Normal	P1	1%	2S	”
3F1Q	3/1	-	2(3)/2(3)	Normal	P1	1%	2S	”
4F1Q	4/1	-	2(3)/2(3)	Normal	P1	1%	2S	”
LockFloor	3/1	1 Func.	2(3)/2(3)	Normal	P1	1%	1S	7.2.4
LockNotF	3/1	1 Func.	2(3)/2(3)	Normal	P1	1%	1S	”
LockSameF	3/1	2 Func.	2(3)/2(3)	Normal	P1	1%	1S	”
LockSpec.	1/3	1 Func.	2(3)/2(3)	Normal	P1	1%	1S	”
Lock_w	1/3	w lock	2(3)/2(3)	Normal	P1	1%	1S	”
Lock_t	1/3	t lock	2(3)/2(3)	Normal	P1	1%	1S	”
Lock_x	1/3	x lock	2(3)/2(3)	Normal	P1	1%	1S	”
Cap_4_5	1/3	-	4(5)/3(4)	Normal	P1	1%	1S	7.2.5
Cap_3_4	1/3	-	3(4)/2(3)	Normal	P1	1%	1S	”
Cap_2_3	1/3	-	1(2)/2(3)	Normal	P1	1%	1S	”
Spread_1_2	1/3	-	1(2)/2(3)	Normal	P1	1%	1S	”
Spread_3_4	1/3	-	3(4)/2(3)	Normal	P1	1%	1S	”
Spread_4_5	1/3	-	4(5)/2(3)	Normal	P1	1%	1S	”
LessNeighb	1/3	-	2(3)/2(3)	Fewer	P1	1%	1S	7.2.6
MoreNeighb	1/3	-	2(3)/2(3)	More	P1	1%	1S	”
P_75/25	1/3	-	2(3)/2(3)	Normal	80/20	1%	1S	7.2.7
P_50/50	1/3	-	2(3)/2(3)	Normal	50/50	1%	1S	”
P_25/75	1/3	-	2(3)/2(3)	Normal	20/80	1%	1S	”
P2	1/3	-	2(3)/2(3)	Normal	0/100	1%	1S	”
Area+0	1/3	-	2(3)/2(3)	Normal	P1	0%	1S	7.2.8
Area+5%	1/3	-	2(3)/2(3)	Normal	P1	5%	1S	”
Area+10%	1/3	-	2(3)/2(3)	Normal	P1	10%	1S	”
Area+15%	1/3	-	2(3)/2(3)	Normal	P1	15%	1S	”
One-Stage_1Q	3/1	-	2(3)/2(3)	Normal	P1	1%	1S	7.2.9
Two-Stage_1Q (sim.)	3/1	-	2(3)/2(3)	Normal	P1	1%	2S *sim*	”
Two-Stage_1Q (it.)	3/1	-	2(3)/2(3)	Normal	P1	1%	2S *it*	”
One-Stage_2Q	3/2	-	2(3)/2(3)	Normal	P1	5%	1S	”
Two-Stage_2Q (sim.)	3/2	-	2(3)/2(3)	Normal	P1	5%	2S *sim*	”
Two-Stage_2Q (it.)	3/2	-	2(3)/2(3)	Normal	P1	5%	2S *it*	”

evaluated in relation to the Base Case. In Table 7.2, the changes in relation to the Base Case of all tests are outlined. Each type of modification is done consecutively with a set of various instances, keeping other traits than the ones in focus equal to those of the Base Case. All the instances of this technical study are smaller and less comprehensive than a real-world case would be. Most of the instances consist of one single floor, and the role of stage 1 (allocating functions to floors) is then diminished. Most of the technical instances are consequently solved to optimality in one stage.

For each test performed, results of instances with similar features are summarized in tables for each section. These include computation time to optimal solution, and objective values obtained, in addition to information relevant to each test.

7.2.1 Data for the Technical Study

The data for the technical study includes a footprint equal for all test instances, except for those with the purpose of testing varying footprints and number of solution stages. Also, a range of functions of varying sizes is defined. The majority of the data of the tests are constant and equal to the input data of the Base Case. Table 7.3 summarizes the key figures of the data used in the Base Case, which is also the data of the tests unless otherwise is specified. The following tests inspect variations of different aspects of these data, outlined in Table 7.2

Table 7.3: Key figures of the Base Case

Data	Specifications	
Footprint	Description	1 Floor, 3 Quadrants (+ 1 location)
	Total area [m^2]	3340
	Areas per locations [m^2]	250 200 250 110 110 110 250 200 250 110 110 110 250 200 250 110 110 110 250
	Max Functions capacity (per quadrant)	2 2 2 2 1 1 2 2 2 2 1 1 2 2 2 2 1 1 2
	Max Centres capacity (per quadrant	2 2 2 1 1 1 2 2 2 1 1 1 2 2 2 1 1 1 2
	Area increase (extra area available)	1%
	Waiting-cost elevator (distance units)	2
Functions	Number	13
	Total area [m^2]	3340
	Areas per function [m^2]	350 250 50 380 343 244 248 195 351 272 208 199 250
	Max Spread	3 3 2 3 3 2 2 2 3 3 2 2 3
Proximity values	Perspective 1	100 %
	Perspective 2	0 %

7.2.2 The Base Case

The Base Case consists of the thirteen locations shown in Figure 7.2, with sizes varying from 110 to 250 m^2, and a total size of 3340 m^2. In addition, all locations of the instance are allowed to increase the area with 1% to provide flexibility of placing functions. Each node has a capacity of housing parts of two to three functions and the centers of one to two functions depending on their size. All functions can be spread across one to three nodes each, also depending on the size of the function. Perspective 1 on proximity values is given weight 100%, so the Base Case is completely based on this perspective and the corresponding proximity values between pairs of functions. No functions are locked to any locations. The proximity values used in the Base Case and the other technical tests are shown in Table 7.4. The computation time of the Base Case is 20 minutes and 39 seconds,

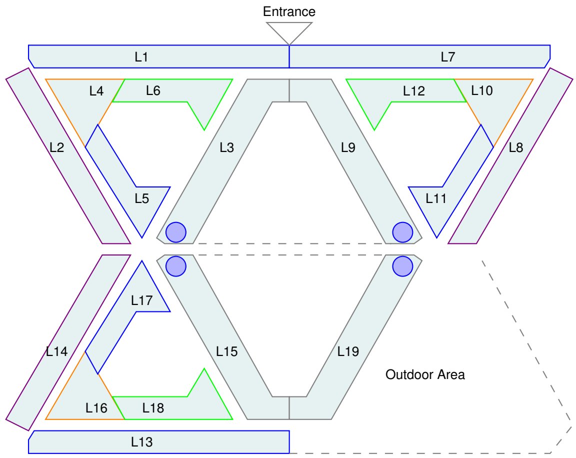


Figure 7.2: Footprint of Base Case with locations open for placement

and the objective value obtained is 872. The resulting optimal layout is shown in Figure 7.3 and illustrates the placements of the functions on the grid resulting from the given data. In the layout, each function is represented by a color, and the center of the function is indicated by an F and the function number. In the next sections, the layout of the Base Case is used as the reference.

f/f	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**
1				1					3	3	8		
2			10	9	8								
3				9	10				6				
4					8				9				
5							2						5
6		4					2			3			6
7	1	2	10										5
8			6	3		1			5				
9	7												
10				9									
11			8	3	9								10
12					1	6			6		9		10
13						5				2	8		

Table 7.4: Proximity values between functions of the technical study

(The upper right half represents Perspective 1 and the lower left half Perspective 2)

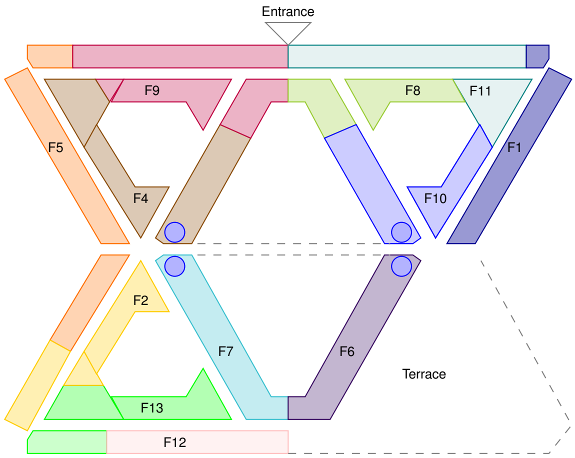
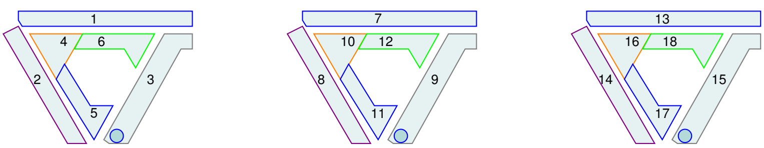


Figure 7.3: Result of Base Case

7.2.3 Varying Size of Footprint and Number of Floors

The footprint of modern hospital buildings varies greatly both in size and shape (Sykehusbygg HF, 2017a). Changing footprint and size of the footprint brings valuable insight into the functionality of the model on problems of different sizes. The nature of the shape of the building in the plans of the new hospital in Hammerfest gives four quadrants on each floor containing locations where functions can be placed. The Base Case has a footprint of one floor and three quadrants, plus an extra location connecting two of the quadrants. Increasing the footprint of the Base Case by one quadrant, or adding an additional floor, causes the computation time to increase significantly due to the quadratic nature of the model.



(a) Floor 1 **(b)** Floor 2 **(c)** Floor 3

Figure 7.4: Footprint of a 3-floor 1-quadrant building

Table 7.6 shows the results of the test instances. The name of the instances gives the number of floors and the number of quadrants on each floor, respectively, for example, instance *3F1Q* consists of three floors, with one quadrant on each floor, illustrated in Figure 7.4. As the size of the footprint increases, the number of functions to be placed is increased accordingly, to keep the area ratio between locations and functions somewhat consistent.

Table 7.6: Results of test instances with varying footprint

Instance	**Footprint** *Floor/Quad*	**Functions**	**Objective Value**	**Elapsed t** [hh:mm:ss]	**Rows/ Presolve**	**Columns/ Presolve**
1F1Q	1/1	6	356	00:00:0.32	487/406	324/306
1F2Q	1/2	10	668	00:00:45	3136/2852	2556/2496
Base Case	**1/3**	**13**	**872**	**00:20:39**	**8597/8226**	**7594/7543**
1F4Q	1/4	16	1312	12:26:16	17651/17110	16120/16008
2F1Q	2/1	10	699	00:00:12	2861/2606	2350/2292
3F1Q	3/1	13	869	00:01:40	6719/6388	5886/5814
4F1Q	4/1	16	1363	10:37:49	15343/14806	13912/13800

As the footprint is changed, the optimal solution changes due to the changed number of functions and changed possibilities for placing functions on different shapes and sizes of footprints. Increasing the number of floors has similar effects as adding quadrants to a single-floor building. The objective value increases with expanding the footprint and increasing number of functions, though not comparable due to increasing number of functions, and hence, varying number of proximity relations. Figure 7.5 shows the development of computation time for both increasing footprints on one floor horizontally and increase the number of floors. The effect is similar for both types of expanding the footprint, also because of the number of functions to be placed are increased accordingly.

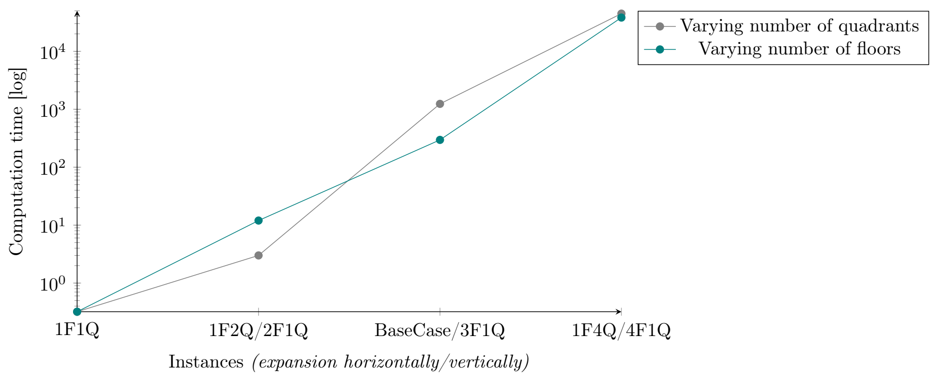


Figure 7.5: Computation time with increasing size of building

7.2.4 Locking Functions to Locations

The implementation of the model incorporates several ways of locking functions to locations. These include locking functions to a specific floor, locking functions to *not* be placed on floors, and locking functions to locations having a specific trait, for example, windows. Also, there are possibilities of forcing a function to be present in a specific location by w_{fi}, to cover at least a specified share of a location by t_{fi}, or to lock a function's center to a specified location by x_{fi}. Additionally, functionality for locking pairs of functions to the same floor is implemented, without necessarily specifying which floor they should be placed on.

In the test instances, the different methods for locking functions are tested consecutively, using one type of lock in each instance. The instances have a footprint with either 1 floor and 3 quadrants (as in Base Case), or 3 floors and 1 quadrant for the tests involving locking functions related to different floors. All instances with results are presented in Table 7.7. Which function, floor or location each lock considers is presented as follows. *LockSpec* locks functions to a location with a certain specification is indicated with *(Function) Specification.* In this instance, function 1 is locked to be located in any location that has a window. *Lock_x* and *Lock_w* are represented in the table as *(Function Location)*, where the function and the location it is set to be located in is stated in the parenthesis. *Lock_t* is given with *(Function Location) Fraction* where the fraction indicates the share of the designated location. *LockFloor* and *LockNotFloor* is indicated by *(Function Floor)*, and *LockSameFloor* gives the two functions that are to be placed on the same floor, by *(Function Function)*. As seen from Table 7.7, the objective value can only remain the same or be weakened as a consequence of locking functions. This follows naturally as the optimal solution may no longer be available when specific placements are imposed on the problem.

Table 7.7: Results of test instances with functions locked to locations

Instance	**Footprint** *F/Q*	**Locked functions**	**Objective Value**	**Elapsed t** [hh:mm:ss]	**Rows/ Presolve**	**Columns/ Presolve**
Base Case	1/3	**None**	**872**	**00:20:39**	**8597/8226**	**7594/7543**
LockSpec.	1/3	(1) Window	872	00:14:08	7733/7186	6694/6568
Lock_w	1/3	(1 1)	875	00:10:07	8597/7678	7594/7031
Lock_t	1/3	(1 1) 0.5	937	00:13:19	8597/7094	7594/6786
Lock_x	1/3	(1 1)	950	00:11:30	8597/6070	7594/5804
LockFloor	3/1	(1 1)	1178	00:08:21	6973/6523	6009/5934
LockNotF	3/1	(1 1)	1098	00:04:23	7381/6979	6435/6360
LockSameF	3/1	(1 2)	1098	00:06:34	7792/7435	6861/6786

An interesting result is a reduction in computation time when locking functions. This is a consequence of the reduced complexity of the problem when one function has limited choices of placement, and the other functions have to be placed in relation to this. For all lock instances except *Lock_x*, *Lock_w*, and *Lock_t*, the decision variables of other functions/locations are pre-processed with information on locked functions, resulting in a reduced number of rows and columns compared to the Base Case.

7.2.5 Varying Allowance in Spread and Capacity

For each function, a maximum allowed spread is defined, controlling how many locations the function can be spread across. In the same way, a maximum number of functions allowed in a location is defined as the maximum allowed capacity for each location. Both these parameters have the purpose of avoiding solutions where functions are scattered in many different locations, making them difficult to connect physically in the building physically. In addition, the number of centers of functions allowed in each location node is defined slightly stricter than the capacity to force a certain degree of spreading of functions centers, to avoid all functions being centered in the same location. The number of allowed centers in each node is maintained the same throughout the tests to avoid clustering of centers.

The maximum capacity of functions is based on the area of the location in question, and similarly, the maximum spread is defined based on the size of the function. Small functions/locations are assigned a lower number of spread/capacity, while larger functions/locations are assigned a higher number. The values given for spread/capacity of each function/location in the Base Case are shown in Table 7.3. The rest of the instances are given by decreasing and increasing spread/capacity compared to the Base Case and are summarized in Table 7.8. In the instance *Cap_4_5*, the maximum capacity is four functions for small locations and five for larger ones. Similarly, instance *Spread_3_4* allows small functions to be spread across three locations, and large ones on four.

Table 7.8: Results of test instances with varying values for spread/capacity

Instance	**Capacity**	**Spread**	**Objective Value**	**Elapsed t** [hh:mm:ss]	**Avg./max** functions per node	**Avg./max** nodes per function
Cap_4_5	4 or 5	3 or 4	842	00:14:51	1.52/3	2.23/3
Cap_3_4	3 or 4	2 or 3	847	00:16:07	1.52/3	2.23/3
Cap_2_3	2 or 3	2 or 3	847	00:15:03	1.52/3	2.23/3
Base Case	**1 or 2**	**2 or 3**	**872**	**00:20:39**	**1.42/2**	**2.07/3**
Spread_1_2	2 or 3	1 or 2	infeasible	–	–	–
Spread_3_4	2 or 3	3 or 4	848	00:06:19	1.42/2	2.07/4
Spread_4_5	2 or 3	4 or 5	848	00:08:16	1.42/2	2.07/4

All rows/presolved rows and columns/presolved columns have the same values as the Base Case (8597/8226 and 7594/7543) since the change in capacity/spread does not affect the decision variables or constraints created. The instance *Spread_1_2* tests a decreased allowed spread of functions and is infeasible when executed since

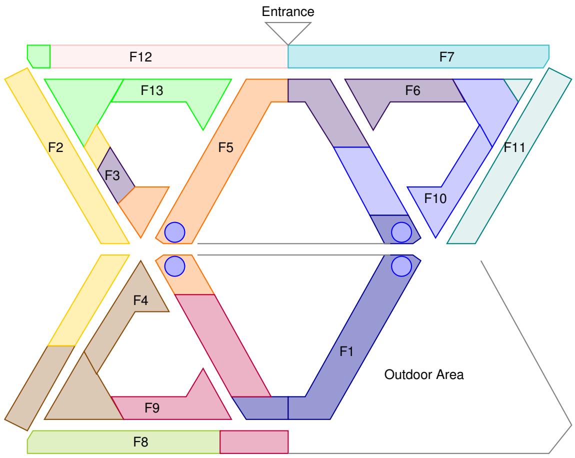


Figure 7.6: Footprint of instance *Cap_4_5*

some functions are too large to be spread across the number of locations that it is limited to. All solutions with increasing capacity or spread give lower objective values than the Base Case. From an optimization aspect, these solutions are preferable as the functions, in general, are placed closer to one another. However, the layout resulting from increased capacity/spread shows a scattered allocation of functions as a node can be divided into more parts. Figure 7.6 shows the resulting layout of the instance *Cap_4_5* with maximum capacity of four and five functions per location. Some of the locations hold parts of three different functions and certain functions are spread across four locations, neither allowed in the Base Case. In the *Cap_4_5* instance, in addition to capacity being increased to four or five functions, the allowed spread of functions are also increased by one per function to avoid this value from constraining the solution too much.

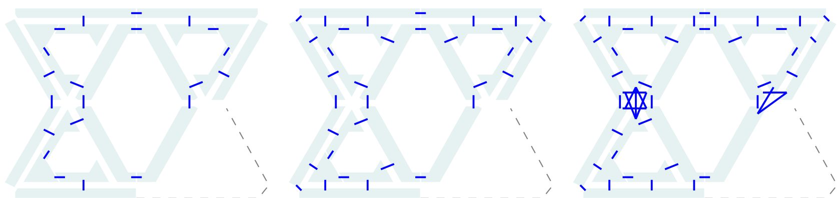
The effect of scattering of functions can be seen in the figure, by small parts of the functions being placed in locations not directly adjacent to the rest of the function. In scattered layouts, the possibility of physically connecting all parts of a function spread over different locations is weak. For a layout like this, linking all parts of a function is not always possible without moving functions' centers and hence change the objective value. This indicates that the solutions obtained are not as good as for the Base Case.

7.2.6 Varying Number of Neighbours

One aspect of how the locations of the building are modeled is the definition of neighboring locations for each location, only allowing functions only to be spread across locations that are neighbors. As neighbors are a choice of modeling, finding the right amount for each location is an important decision to be made. Giving each location the right number of neighbors will give results of convenient spreading of functions. Defining too few neighbors may lead to inefficient spreading or unsolvable problems while defining too many neighbors can cause an effect similar to allowing too much spreading of functions; frequent and inexpedient splitting of functions. For the tests, all parameters except the number of neighbors are kept constant. The neighbors defined for each location is illustrated in Figure 7.7

Table 7.9: Results of test instances with changed number of neighbours

Instance	Number of Neighbours	Objective Value	Elapsed t [hh:mm:ss]	Rows/ Presolve	Columns/ Presolve
Less Neighbours	Fewer (18)	1034	00:35:50	8597/6717	7594/6301
Base Case	**Normal (32)**	**872**	**00:20:39**	**8597/8226**	**7594/7543**
More Neighbours	More (39)	823	00:03:03	8597/8226	7594/7543



(a) Less Neighbours **(b)** Base Case **(c)** More neighbours

Figure 7.7: Footprint with neighbour relations for test instances

(blue lines indicate neighbouring locations)

From Table 7.9 it can be observed that the objective value is higher when few neighbor relations are defined, and lower (hence better) with more neighbors. With few neighbors, a more compressed allocation of each function follows, and the centers of the functions are further apart, while more flexibility is allowed in placements of functions when more neighbors are defined. The layout resulting from the instance with fewer neighbors is quite similar to the Base Case. In both cases, an inexpedient splitting of functions is avoided. The layout from the instance with more neighbors has some unwanted scattering of functions, caused by the wide definition of neighbors especially in the central parts of the building, near the elevators. For the instance having few defined neighbors, the solution software can reduce the number of rows and columns after pre-solving. This is due to the reduced possibilities of placements of functions in this instance; with fewer neighbors, some functions are too large to be placed in certain locations.

7.2.7 Varying Weights on Perspectives of Proximity Values

For all the technical test instances evaluated in the technical study so far, the proximity values are based solely on perspective 1. A study worth doing is looking at how the layout, computation time, and objective value turns out when including perspective 2 in the tests partly or fully. Five instances are tested and shown in Table 7.10, which are based on various weights of the two perspectives. Other parameters are maintained equal to the Base Case throughout the test instances. The proximity values of the two perspectives can be found in Table 7.4. Perspective 2 contains slightly fewer relations than perspective 1, and hence some fewer rows and columns are created as a proximity value of positive non-zero value is the criteria of some constraints.

Table 7.10: Results of test instances with varying weights on perspectives

Instance	Perspective	Objective Value	Elapsed time [hh:mm:ss]	Rows/ Columns	Prows/ Pcolumns
Base Case	**P1 = 1**	**872**	**00:20:39**	**8597/8226**	**7594/7543**
P_75/25	P1 = 0.75 P2 = 0.25	981.75	03:11:10	14389/14002	13066/13015
P_50/50	P1 = 0.5 P2 = 0.5	1008.5	20:40:46	14389/14002	13066/13015
P_25/75	P1 = 0.25 P2 = 0.75	928.75	10:06:52	14389/14002	13066/13015
P2	P2 = 1	786	01:07:13	8235/7865	7252/7201

An increase in computation time is observed for the instances combining values from both perspectives compared to the base case and the instance using only perspective 2. With a high number of non-zero proximity values, a correspondingly high number of relation variables (y_{figj}) are created from the placement variables (x_{fi}), and this is the main reason for the increased problem size and computation time. Another reason for this result could be the large number of proximity values that obtain averaged values from weighting the perspectives of different proximity values, rather than more extreme values when based on one single perspective. When many pairs of functions have similar requirements for proximity, they are harder to distinguish, and finding an optimal solution can be more challenging and time-consuming. For a real-world hospital, many, if not all functions will have relations, and consequently, they all need some degree of proximity. However, observing the difference in run time with a decreased number of proximity values points towards the advantages of not including relations between all functions of a hospital, but rather emphasize the most important ones.

In order to evaluate the resulting layouts of the instances, the layouts from each perspective are locked and tested with the weights of perspective 1 and perspective 2 used in the other instances. The objective values obtained when evaluating a layout with the other perspectives are shown in each column of Table 7.11, where the lowest value in each column is given the lightest color, and the highest value has the darkest. As observed from the table, the layouts corresponding to weights given solely to each of the two perspectives give the best objective values for each perspective, and the objective value increases when moving towards more significant shares of the other perspective(s). As expected, all the solution layouts perform best when evaluated on their original weighting of perspectives, and worse when put in the context of another perspective. This indicates that the obtained layouts are the best solutions for their respective perspectives.

Table 7.11: Objective value of technical instances with different sets of proximity values

	Objective value of				
Instance	**P1**	**0.75/0.25**	**0.5/0.5**	**0.25/0.75**	**P2**
Base Case	872	1 033.75	1 155.5	1 264.25	1 373
P_75/25	912	981.75	1 051.5	1 121.25	1191
P_50/50	1 090	1 062.25	1 008.5	954.75	901
P_25/75	1 354	1 228.75	1 078.5	928.75	779
P2	1 528	1 342.5	1 151	959.5	768

To analyze the model's ability to prioritize placing functions with high requirement of proximity values nearby each other, the average distances between functions having each proximity value (rounded to nearest integer for the weighted sums) are compared for all proximity values. Table 7.12 and Figure 7.9 show the averaged distances for the different proximity values. All instances show a decreasing trend from beginning to end. For the layout of perspective 1 (Base Case), the average distances are strictly decreasing by increasing proximity value, except for the relation having proximity value 1. For the other instances, there are more deviating values. In addition to the average distances, Table 7.12 also shows the number of relations that have a given proximity value. This value is to some extent a validity measure for the average distances calculated, as it shows the number of function pairs the calculations are based on. For the Base Case, the average distance for proximity 1 is divergent from the trend, but referring to the number of relations for proximity value 6 shows that this average is based only on one pair of functions. Most deviant from the decreasing trend of average distances is the instance of perspective 2. For this instance, there are in general fewer relations for each proximity value, and the most deviant values are based on few proximity values.

In conclusion, the relationship between the instances and the layouts are in general in accordance with the proximity values defined for each instance as higher proximity values, in general, give lower average distances between functions.

Table 7.12: Average distance between functions of equal proximity

Proximity Values		0	1	2	3	4	5	6	7	8	9	10
P1	# of relations	135	1	2	3	0	3	2	0	3	3	4
(Base Case)	Avg. dist.	13.7	16.0	9.6	9.5	-	8.3	7.5	-	6.0	6.0	10.5
P1=0.75/	# of relations	116	9	9	7	3	2	3	3	4	2.0	0
P2=0.25	Avg. dist.	12.8	12.7	10.2	12.4	7.3	6.0	6.3	7.0	9.0	1.5	–
P1=0.5/	# of relations	116	8	6	9	6	11	0	0	0	0	0
P2=0.5	Avg. dist.	11.9	11.4	8.2	9.1	9.5	8.7	–	–	–	–	–
P1=0.25/	# of relations	116	9	10	10	1	3	3	3	1	0	0
P2=0.75	Avg. dist.	12.8	13.4	10	11.2	6	9.7	8.7	4.3	9	–	–
P2	# of relations	137	3	2	2	1	1	3	1	2	3	1
	Avg. dist.	15.4	10.7	18.5	12	5	13	9	6	5.5	5.6	6

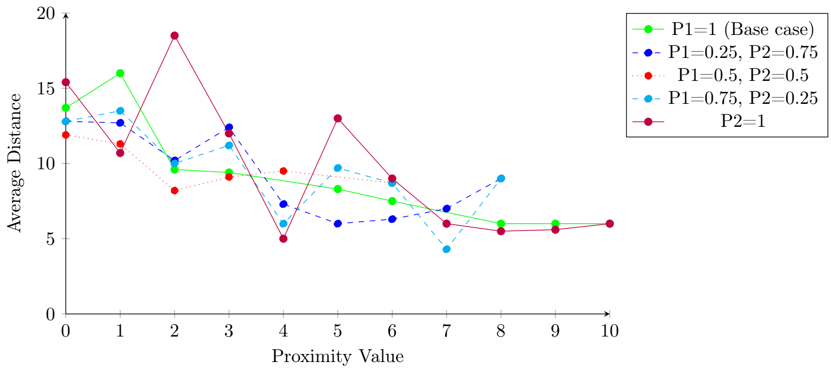


Figure 7.9: Average distance between functions of equal proximity values

7.2.8 Area Increase

In order to obtain results in accordance with the proximity values of the functions, in other words, that the functions deemed close to each other are in fact located near each other, it is important to ensure that the model is allowed to account for these values, without the placement being constrained by other factors. One factor that could interfere with the ability of the model to account for proximity values is the limitations enforced by the areas of the locations versus the area of the functions to be placed. If the areas represent a strict constraint, the functions will be located based on their required areas, diminishing the importance of their relations to other functions. The considerations of area increase are especially relevant when the model is used with multi-floor buildings. Inside one floor, any lack of available area for a function in a location could be obtained in one of the neighboring locations. As no neighbors are defined across floors, the sum of the areas of the functions allocated to a floor must fit inside the locations available on this floor. If not allowing any margin, the functions may be allocated to floors based solely on whether their areas fit together on each floor. On footprints of only one floor, the area increase gives certain flexibility increasing the probability of obtaining a solution within reasonable computation time.

Based on this, a certain margin should be given to the area of the functions, or equally on the area available in each location. The latter has been chosen, and four instances are tested with other choices of values for the area increase. Area increase values of 0%, 5%, 10% and 15% are tested. The results can be seen in Table 7.13.

Table 7.13: Results from test instances with varying area increase

Instance	Area increase	Objective Value	Elapsed time [hh:mm:ss]
Area +0%	0%	*995 (Best b.)*	*12:44:56*
Base Case	**1%**	**872**	**00:20:39**
Area +5%	5%	846	00:08:48
Area +10%	10%	805	00:06:28
Area +15%	15%	793	00:03:39

The Base Case has an allowed area increase of 1% and runs for about 21 minutes to the optimal solution with an objective value of 872. Increasing the allowed area improves the objective value, and lowers the computation time. Denying any additional area (area increase 0%), causes the model to run for about 13 hours and then run out of memory with no feasible solution found, but with a best bound of 995. Considering real hospital buildings, some extra area is deemed necessary and allowed, and is included in the planning of hospital layouts as of today. An increase in the area of 1% is within the margin of any of the studied suggested layouts (Sykehusbygg HF, 2018b), and will not significantly affect the result considering the final functionality of the hospital. An area increase of 5% gives a result similar to the result of the Base Case, with slightly higher tendencies of clustering of functions, in addition to some of the locations being covered to max (1.05 of the area) and some having available space. This is, in fact, the general tendency when expanding the area increase, but considering the way Sykehusbygg plans hospitals, and as long as the area increase values are kept reasonable, spreading these functions across the whole footprint by shifting the functions towards empty locations, is a simple adjustment to perform after obtaining the layouts.

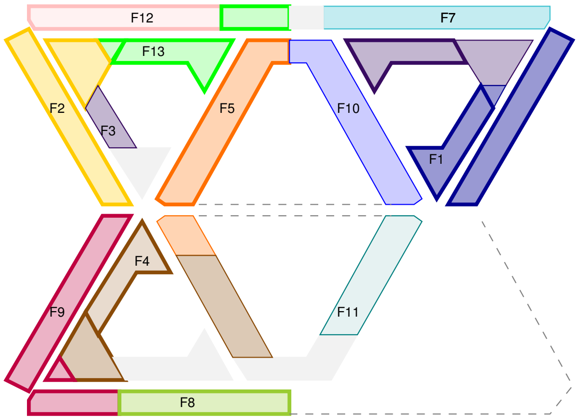


Figure 7.10: Footprint of a test instance with 15% area increase

(grey locations indicate no placement of functions)

An example of inexpedient clustering is shown in Figure 7.10 with 15% area increase. The locations with bold lines indicate that their area has been expanded to maximum, in this case to 1.15 times their real area. The grey locations on the figure indicate free space of locations. Even though adjustments could be performed after retrieving this result, too much clustering, as seen in the figure, is not desirable. A satisfying layout considering operational factors is in this case not achievable without changing placements of centers of functions and hence change the objective value.

7.2.9 Number of Solution Stages

A central aspect of this thesis is the choice of solution method for the model. As the size of the problems increases, the need for a two-stage approach rises. As the division in stages by separating the allocation of functions to each floor is, in fact, a simplification that will affect the result, a discussion on the quality of the solutions obtained by the two-stage method is appropriate.

Two instances of different size and footprint are created an tested with three different approaches to solution method. The first instance has a building of a one quadrant-footprint on three floors with 18 functions to be placed. A larger building with more functions is tested as well, as the solution of a larger problem is expected to show clearer variance from different solution methods than the smaller one. The larger instance consists of three floors, the two first floors having 2 quadrant-footprint and floor 3 having 1. There are 24 functions to be placed in the larger building.

For each of the two building structures, instance *One-Stage* solves the three floors simultaneously, with all functions free to be placed on any floor. In the two other instances, an initial stage 1 is added, allocating functions to floors before performing the internal allocation of the floors in the second stage. The *Two-Stage Simultaneous* instance then solves the three floors simultaneously, having the functions locked to their designated floor. Instance *Two-Stage Iterative* solves the floors iteratively from floor 1 to floor 3, based on the distribution of functions from the first stage, each floor accounting for the floors already solved directly, and the floors not yet solved as values from the functions in a total floor-node, similar to the method described in Section 6. The results from the instances are shown in Table 7.14.

Table 7.14: Results from test instances with solution methods of one and two stages

Instance	**Solution Approach**	**Objective Value**	**Elapsed Time [s]**			
			Floor 1	Floor 2	Floor 3	**Total**
3 Floors, 1 Quadrant footprint, 18 functions			*Elapsed time Stage 1: 0.10s*			
One-Stage	All floors solved simultaneously	1098	–	–	–	296
Two-Stages Simultaneous	Locked to floors in stage 2 and solved simultaneously	1295	–	–	–	7
2 Stages Iterative	Stage 2 solved iteratively F1 → $F2 \rightarrow F3$	1295	0.25	0.10	0.05	0.5
3 Floors, 2 + 2 + 1 Quadrants footprint, 24 functions			*Elapsed time Stage 1: 0.20s*			
One-Stage	All floors solved simultaneously	*1508*	–	–	–	*41503* *76% gap*
Two-Stage Simultaneous	Locked to floors in stage 2 and solved simultaneously	1630	–	–	–	80
Two-Stage Iterative	Stage 2 solved iteratively F1 → $F2 \rightarrow F3$	1678	0.30	0.15	0.15	0.8

Considering further use of the model on larger cases than the instances described in this technical study, the computation time elapsed for the different solution method is the most interesting result from these tests, due to the previously described exponentially increase in this measure for an increase in problem size. For the relatively small case with one quadrant on each of the three floors, the total allocation process can easily be done in one stage. However, solving the problem with two stages, either simultaneously or iteratively, represent a decrease in computation time. This decrease is even more comprehensive when solving the instances with the larger building in two stages. The software was not able to find any solution for the larger instances by the one-stage approach and ran out of memory after about 12 hours.

The objective value is, as expected, weakened (increased) when using more stages, as the initial division on different floors removes some of the flexibility in the internal allocation of stage 2. In addition, the objective value of solving the second stage simultaneously for all floors has a better expectancy than what is obtained when solving the floors iteratively, again caused by the higher degree of flexibility in allocating functions in relations across floors. In the smallest test instances, the objective values of both two-stage approaches turn out equal, but when solving for the larger building, the difference becomes visible. Solving stage 2 iteratively results in a slightly weaker objective value than what is obtained when solving all the floors in stage 2 simultaneously.

7.3 Discussion

Several aspects of the implementation of the model are tested separately, and the observations indicate different impacts on both objective value, layout and computation time from changing the different features of the model. When increasing the size of the footprint, the floors of the building and the number of functions of the test instances increases the complexity of the problem, and consequently, the computation time increases as expected. As a hospital will usually be an extensive building consisting of several floors, being aware of the impact on computation time may lead to making efficient choices in other aspects of the modeling; as the footprint and the total size rarely is a factor available for change or decreasing in size. However, independent of the actual size of the building, the footprint could be discretized to a higher or lower degree, and the functions could be divided into smaller or larger parts, depending on the desired degree of simplification of the problem, again depending on the case application. Since parts of the complexity of the problem come from the number of functions and number of areas, decreasing this number by defining larger locations and functions can reduce the computation time, but will also lead to a less detailed layout.

As observed, an efficient strategy for decreasing the computation time to find the optimal solution is to apply some form of locking of functions. For the instances with one floor and locking functions independent of floors, locking the center of a function is the strategy that has the largest effect on computation time. In this context, it is important to note that which functions that are locked to different locations may impact the performance of the instances compared to each other differently, as they are somewhat arbitrarily chosen. From a hospital point of view, forcing functions to *cover* locations may be desirable instead of locking the center of the function, providing more flexibility in where to place the rest of the function. This type of locking is available to the method demonstrated in $Lock_w$ and $Lock_t$. However, locking functions can cause convenient solutions to be lost. In hospitals, vital functions that need to be located at certain places, for instance near entrances, can be locked to be present at the relevant location. The locking of functions related to floor has the same time-reducing effect as the above-mentioned alternatives, and locking specifications and functions are in many real cases preferable and necessary, as certain functions have specific needs for placements in the building. For instance, the emergency department is unlikely to be placed on floor 5 of a hospital, and bed wards are not favorable to have on floor 1.

Both choosing the right capacity and spread of locations and functions, and defining the most appropriate amount of neighbor locations dependent on the layout are factors affecting the outcome of the final layouts. The Base Case is defined with each location having the capacity of one or two functions, and each function has a maximum spread of two or three locations. These values seem appropriate, as the layout produced by the Base Case both has good objective value and good functionality when considering placements of functions. Even though resulting in better objective value, instances with higher spread or capacity show tendencies of unwanted scattering of functions. In realistic cases, the locations and functions might be larger and of varying sizes. This raises an important question on whether to allow more variation in capacity/spread, both due to efficiency and a wish to obtain a good solution and to ensure feasible solutions.

Choosing the right amount and composition of neighbors gives good results in the allocation of functions concerning an even, yet small distribution without scattering. The instance tested with more neighbors leads to an inexpedient splitting of functions, while too few can lead to unsolvable problems or too compact grouping of functions with poor objective values. When defining neighbors in a real hospital, the choice of neighbors needs to be logic dependent on which locations are adjacent and likely to suit as neighbors for a function.

In a hospital, it is likely that the majority of the functions have some relation to each other. The results of the tests show an increased computation time for the instances that weights different perspectives, compared to the two instances based solely on one perspective. As discussed, the weighting causes more proximity values to be averaged. These results suggest a certain caution when defining proximity values. It is desirable to ensure that important values are emphasized, as the needs for proximity may be less clear to the model if many values are defined similar and averaged. Few proximity relations give emphasize to the values that are included and may lead to lower computation time.

In the test instances, the total area of the functions to be placed equals the total area of the locations available. To provide some form of flexibility, an allowed area increase percentage is added to each location. The technical study showed that both 1% and 5% are reasonable values, possibly with a larger requirement for such margin when considering multi-floor buildings. In hospital layout planning, these margins are used by both planners and architects to fit the right functions in the right locations, and using some extra area is, therefore, both realistic and allowed.

The technical study reveals a significant decrease in computation time when introducing an initial stage allocating functions to floors before solving the internal layout of each floor. When the model is to be used on a real hospital of many floors, the division in two stages is a convenient solution method that still is able to handle relations between floors while solving each floor separately. The difference in computation time is more significant in the larger case, where the one-stage approach is unable to find a solution. In the tests of this technical study, satisfactory solutions are obtained in reasonable time for the two-stage simultaneous method, solving all floors simultaneously after the functions being assigned a floor in stage 1. However, a real case could easily be double the size of the largest instance tested in this technical study, and would hence often be too extensive for using this method on all floors. However, as the simultaneous method gives results better or equal to the iterative, using this method on groups of floors between the iterations of the first method where possible could improve the solution.

When solving the floors iteratively on larger instances, the order in which the floors should be solved may not be obvious. Starting from floor 1 and iterating upwards, as is done in these tests may be a logical approach, but other solution approaches can also give good results. The effect of this choice is hardly traceable in instances as small as the ones in this technical study, but might be applicable for larger cases. Because the solution approach is of importance to the results and should be decided individually from each case, possible solution methods for a real case are further inspected in Section 8.3 of the case study.

As previously stated, the purpose of the model of this thesis is producing layouts to supplement discussions when planning internal hospital layouts. In many cases, a solution close to, but not optimal, will be sufficient for such use. If the optimal solution (using only one stage) is inaccessible due to the necessary computation capacity, a solution obtained by two stages might serve the same purpose, especially if an effort has been made to adjust the order of the iterations or organizing the methods within the two stages used. The test instances used in the technical study are of small scale compared to what a whole hospital building would be. The test instances show that despite the relatively small size, the problem has a high level of complexity, and clever choices of implementing the instances must be made. In the next chapter, a case study is presented, tested and discussed to look at the characteristics of the problem in a more realistic setting.

Chapter 8

Case Study

The overall purpose of this master's thesis is to illustrate how mathematical optimization can be used to assist the process of developing hospital layouts. With proximity values as the primary driver for placement, and other input parameters extending the specifications of needs, layouts that can be discussed by hospital planners are obtained. Because of the mentioned purpose of the model, it is of high interest to test the usefulness of the model on instances that are of the size and relevance to a real hospital, and further on instances based on actual data from a real hospital.

The case study is performed on the hospital that is to be built in Hammerfest, where Sykehusbygg is highly involved in the process. Choices of modeling are made based on the discussions of relevant features presented in the technical study, in addition to pertinent considerations made by Sykehusbygg. Because of the large size and complexity of the problem of a real hospital instance, the two-stage solution method presented in Section 6 is used to solve the case instances.

In Section 8.1 and Section 8.2, a detailed description of the case and the data used in the case is given. Section 8.3 discusses possible approaches to the solution method of stage 2 for the case. Afterward, Section 8.4 applies the method deemed appropriate on six sets of input parameters to analyze the impact of various input parameters and the quality of the solution. An analysis of the layout based on quantitative aspects is performed for all instances, while a qualitative discussion follows the result of the instance based on real considerations of the hospital.

8.1 Case Description

The case instances exemplify how the mathematical model can be used in the hospital layout planning process. As the model up until now has been tested on instances significantly smaller than a real-world case, an important aspect of this case study is testing the model's ability to solve larger instances to optimality using the chosen solution method. A real-world hospital has a large footprint, several floors, and a high number of functions with various sizes, which in turn leads to a problem of higher complexity than what is seen in the technical study. A case built on a set of input parameters from the concept report of the new hospital in Hammerfest (Sykehusbygg HF, 2018b) have been developed and are tested with the model.

The footprint and the internal layout of the building are directly reflected by the floor plans in the Concept Report for the building of Hammerfest hospital (Sykehusbygg HF, 2018b). The shape, which is similar to a bow-shape, is based on considerations regarding logistics for patients, employees and goods, distances and need for closeness, flexibility for the future, and technical infrastructure. Also, the shape of the building and the atria included ensures that there are windows and daylight available in large parts of the building. Five or six floors are planned for the building, including a basement. For the case study, all floors apart from the basement are incorporated in the footprint of the case. The choice of whether or not to include a basement is currently being discussed, as the geography and soil conditions make a basement an extensive that may be hard to defend. Hallways are a part of the building structure, so the gross area calculated for the functions does not include this area.

All functions of the hospital are included in the case study. Hammerfest Hospital is a relatively small hospital, but with a purpose of serving a wide range of functions. Therefore, the hospital has a large variety of different functions. In addition to having bed wards, operation departments and obstetrics, the hospital also includes a wide range of polyclinics for day treatment and should also accommodate functions for mental health for both children and adults. Many of the functions defined to be placed in the hospital are accompanied by support functions and special assigned clinical offices needed to be close to, but not necessarily adjacent to, the function in question. These support functions can be placed on other floors or where it is deemed appropriate, as patients do not use them. However, this still implies that short distances between a function and its support function and/or clinical office are desirable considering walking distances for employees.

The areas of the functions are defined as a gross area, comprising toilets, kitchenettes (small, decentralized kitchens in over-night departments) and electrical installations of the building, among other things. A central kitchen responsible for the cooking of all meals, an IT-service department, and central storage and bed-/sheet-handling are defined as separate functions.

A set of proximity values between pairs of functions is developed in collaboration with Sykehusbygg, emphasizing high values between functions with strong relations. The values reflect the proximity diagrams from Sykehusbygg, exemplified in Section 2. For the case study, only one perspective on proximity is weighted in all tests. The reason for this is that in contrast to using the model with input from different actors, Sykehusbygg has the role of incorporating the different aspects of different stakeholder into one, and therefore, the proximity values included in the plans for the hospital in Hammerfest already comprises different perspectives on proximity values. In addition to the proximity values of the real case, two additional random sets of proximity values are tested for comparison in Section 8.4.

Locking functions in various ways aim to imitate realistic requirements for placement of functions. Locking functions to be at the same floor, to be at a specific floor, to cover or have center in a certain position, or to lie in areas that have certain specifications (e.g., windows) are highly relevant in hospital layout planning. A set of locks for the case is developed and is presented in Section 8.2. On each floor, locations have a defined set of neighbors. These are the locations adjacent to the one in question, and hence the other locations that functions may acquire parts of if centered in the first mentioned location. The number and density of neighbors are defined based on the results obtained in the technical study, and each floor of this case has neighbors defined similarly to the base case of the technical study.

Distances between the nodes are calculated via the corridor nodes and elevators with an all-to-all shortest path algorithm. As earlier mentioned, the footprint of the building is in this model defined by locations, their size and the distances between the centralized nodes in all of them. The question of the distance measure when traveling by elevator one floor or more is as mentioned an important discussion. In this case, the cost of waiting for an elevator is defined based on consideration of how the distance units are defined in a horizontal direction on each floor and an aim to make the comparison of vertical distances between floors and the horizontal distances internally in each floor as realistic as possible. Preliminary studies showed no differences in the resulting optimal layout when changing this measure marginally. Although the case is a considerably larger instance, the chosen elevator cost is considered a fair guess in representing the average waiting time for the elevators, and time spent traveling each floor.

8.2 Data and Input from Sykehusbygg

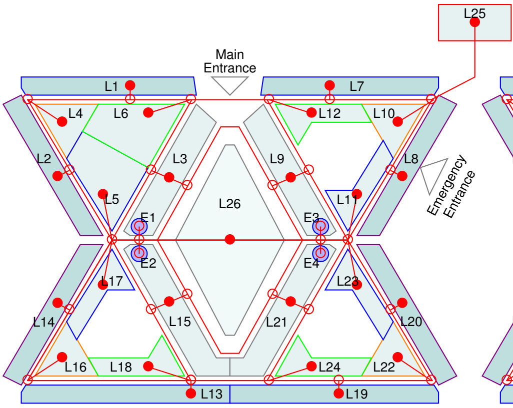
The functions to be placed in the hospital, the layout of the building, the functions locked to locations, and specifications are kept constant throughout the case. The key data for the case is shown in Table 8.1.

Table 8.1: Key figures of the case instances

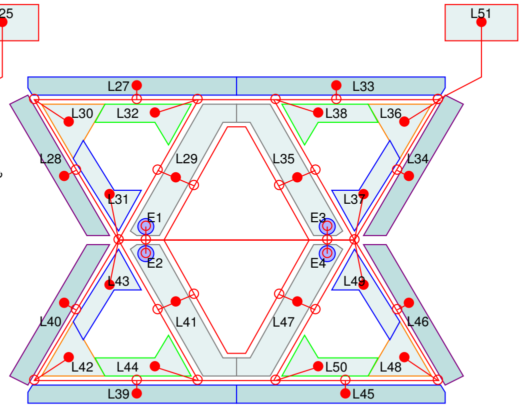
Data	Specifications	
Footprint	Specifications	5 Floors, 1-4 Quadrants
	Area per floor [m^2]	4820 4370 3370 2720 1000
	Total area [m^2]	16280
	Waiting-cost elevator [distance units]	2
Locations	Number of nodes	88
	Areas [m^2]	110 - 375
	Max Capacity (per node)	2
	Max Centers capacity of functions	$< 200m^2$: 1
	(per location area)	$\geq 200m^2$: 2
	Area increase (extra area available)	5%
Functions	Number	57
	Areas [m^2]	81 - 883
	Total Area [m^2]	16 280
	Max Spread over nr. of locations	$< 200m^2$: 2
	(per function area)	200 - $800m^2$: 3
		$> 800m^2$: 4
Distance measure	Waiting time elevators	2 units
	Travelling one floor by elevator	1 unit
Proximity Values	Number of function pairs of each proximity value *(proximity: nr.)*	**0:**1425 **1:**0 **2:**1 **3:**0 **4:**24 **5:**52 **6:**17 **7:**14 **8:**47 **9:**9 **10:**9

The building of the hospital in Hammerfest has five floors, defined by *floor 1* (Ground Floor) to *floor 5*. The footprint of the case with nodes, hallways, elevators, and locations corresponding to each floor are shown in Figure 8.1. Outdoor areas on rooftops and atria are indicated in the figure, and areas that have windows are marked with a darker color. Red dots represent the nodes with corresponding locations available for placement of functions and denoted by "L" and the corresponding location number. As seen from the figure, the floors vary in size and shape. The rooftop terraces and atria are unavailable for functions to be located in. The building consists of 88 locations available for functions, with areas ranging from 110 to 375 m^2. For all the locations there are restrictions on the number of function centers and functions present in the location. All locations in the case can contain fractions of a maximum of two functions. Locations larger than or equal to 200 m^2 can have two function centers, while all locations under 200 m^2

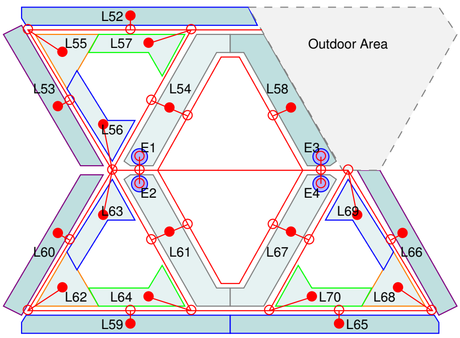
can have one function center.



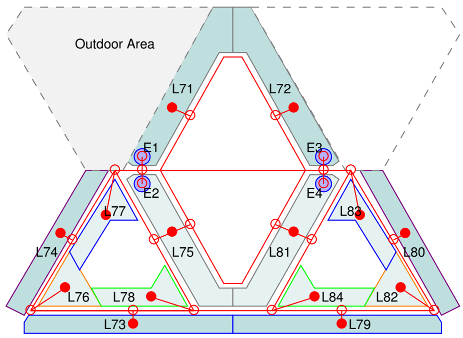
(a) Floor 1



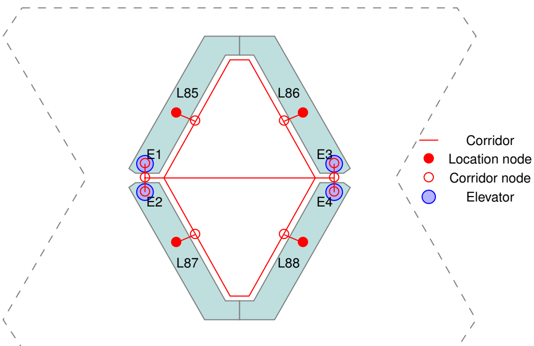
(b) Floor 2



(c) Floor 3



(d) Footprint of floor 4



(e) Floor 5

Figure 8.1: Footprints of each floor of the case instance

From earlier discussions, and as the functions of the case and the areas connected to each location vary in sizes, an area increase of 5% is added to each location.

Caused by the limited locations for placement on each floor, the problem is often not feasible if this area increase is not included, due to the impossible problem caused by the functions to be placed not being equal to the various locations. The result of this adjustment is that functions placed in each location possibly have larger areas in total than what is defined available at that location of the building. However, Sykehusbygg operates with a similar method of making areas of functions somewhat flexible regarding where they are placed, and if functions are compressed at one part of the building, functions could also be shifted manually. The gross area of functions may to some degree be flexible for the architects to later move to wherever there is room.

The case of the hospital of Hammerfest consists of 57 functions to be placed, ranging in area from 81 to 883 m^2. The list of functions used in the case represents all of the departments and other units that will be placed in the hospital. The list of functions included in the case is presented in Table 8.2. Based on the information on which functions will be a part of the new hospital from the Concept Report of Sykehusbygg HF (2018b), some adjustments, and further division are performed to obtain a suitable division of the functions for the model. Functions like for instance office work spaces are divided into different functions connected to their operational field. This is done to decrease the size of the functions, and because the nature of the functions can be divided into separate parts that could be convenient to split into different locations. For some functions like for example bed wards, the size in combination with the required specifications (windows) enforces a division of the function in two. Based on the area of each function, a value that represents the maximum number of locations it can be spread across is defined. Functions with an area below 200 m^2 can be spread across two nodes, functions between 200 m^2 and 800 m^2 can be spread across three nodes, and functions above 800 m^2 can be spread across four nodes.

Proximity values between pairs of functions of the case are shown in Table A.1 in Appendix A. Proximity values between one and ten are given to functions with a special requirement for closeness. The proximity values between all pairs of functions are symmetric. Even though most functions in a hospital have some relation, most of the pairs of functions are assigned a proximity value of zero. This is done to accentuate relations that are especially important and therefore, it does not mean that functions with proximity value equal to zero have zero relatedness. Functions that are split into separate functions have a high proximity value between them if they are highly related, and each of the parts of the functions has the same relation to other functions as the original function has relations too.

As the model locates the functions considering the specified inputs, the quality of

Table 8.2: Record of functions included in the case study

Function	Name	Area $[m^2]$	Function	Name	Area $[m^2]$
1	Emergency Department (Akutt)	641	30	Post Operative	252
2	Observation	311	31	Sterile Central	175
3	Emergency Department (Legevakt)	190	32	Intensive Care (1)	458
4	Imaging (1) - Treatment	316	33	Intensive Care (2) - Support	96
5	Imaging (2) - Support	494	34	Polyclinic - Endoscopy	143
6	Patient information / Reception	144	35	Polyclinic - Expedition	229
7	Centre of Learning and Mastery	102	36	Polyclinic - General Somatic	385
8	Cafeteria	243	37	Polyclinic - Ophthalmology	104
9	Hospital Ministry	101	38	Polyclinic - Gynecology	81
10	Personnel service / accommodation	435	39	Polyclinic - Lung/Cardio	96
11	Laboratories	380	40	Polyclinic - Mental Health and Substance Abuse Treatment	197
12	Central storage, Central kitchen	558			
13	Waste, Cleaning, Garment handling	242	41	Polyclinic - Injury	133
14	Bed handling	114	42	Polyclinic - Support	505
15	IT Services, Medical Tech, MOMD	221	43	Polyclinic - Otorhinolaryngology	126
16	Ambulance (1) floor 1	250	44	Oncological day unit (1)	188
17	Ambulance (2) floor 2	250	45	Oncological day unit (2) - Support	188
18	Day Surgery	270	46	Education	455
19	Office workspace (1)	264	47	Research	135
20	Office workspace (2)	264	48	Bed ward - Medical (1)	883
21	Office workspace (3)	264	49	Bed ward - Medical (2)	883
22	Meeting- and Support rooms	130	50	Bed ward - Orthopedics/Surgery	676
23	Clinical Office - Polyclinic	197	51	Bed ward - Gynecology	222
24	Clinical Office - ER	197	52	Patient hotel	401
25	Clinical Office - Imaging	197	53	Bed ward - Children	398
26	Clinical Office - Bed wards	197	54	Child Habilitation	127
27	Clinical Office - Intensive Care	197	55	Obstetrics (Delivery)	247
28	Operation	706	56	Physio, Occupational therapy, Nutrition (1)	266
29	Pre Operative	191	57	Physio, Occupational therapy, Nutrition (2) - Support	164

the layout comes from the detailing and considerations made to the input data. The input data is a result of a few meetings with Sykehusbygg, discussing special needs regarding relatedness between functions, and specifications of individual functions. This information is used to determine the proximity values and locking of functions. In addition to proximity values, other aspects also impact the solution. To ensure that the hospital fulfills certain demands, a variety of functions are locked in different ways. This will also reduce the complexity of the problem, as shown in the technical study. Table 8.3 displays the locked functions defined in the case study.

Table 8.3: Record of specifications of functions in the case study

Specification	Location/Floor	Functions
Functions locked to specification locations	Windows	44, 45, 48, 49, 50, 51, 52, 53
Functions locked to floor	1	1, 2, 3, 12, 13, 14, 15
Functions not in floor	1-2	46, 47, 48, 49, 50, 51, 52, 53
	1-3	19, 20, 21, 22
	3-5	4, 5, 8
Center of functions locked in location	25	16
	51	17
Functions locked to cover location	8	1
	26	6, 7, 9
Functions locked to the same floor		4&5 18&28 29&30, 32&33, 44&45, 52&53, 56&57

8.3 Solution Methods for the Case

This section inspects the different strategies for solving stage 2 after stage 1 is settled. Stage 1 is solved for the input data developed for the Hammerfest hospital case, and the functions assigned to each of the five floors of the hospital are shown in Table 8.4. This creates an equal basis for the different solution methods of stage 2 that are tested and discussed in this section. It is important to keep in mind that the results of stage 1, for example the number of functions placed and the number of relations on the various floors, impact the computation times and the objective value, and that applying the methods on another stage 1-solution may give different performance.

Firstly, an attempt is made to solve the whole hospital in one stage. This is not successful with the computation capacity available. The instance runs out of memory after 49 hours, without finding any feasible solutions. Consequently, this proves that to use the model on a case instance as large as the case of Hammerfest Hospital without large extensions of computational capacity, an alternative solution method is required. If separating the solution method in two stages, as discussed in Section 6, satisfactory solutions can be obtained. Stage 1 is quite straight-forward while solving stage 2 can be done using several different approaches. To better understand how choices of approach and the order in which floors are solved affect different measures in stage 2, this section presents several ways of sorting the floors and incorporating different levels of information transferred between the iterations when solving stage 2.

Table 8.5 summarizes the different ways in which stage 2 is executed, including key results. The various methods are compared in relation to computation time

Table 8.4: Functions assigned to floor 1-5 in stage 1 with proximity set 1 and locks of functions to floors (Case Instance 1)

Floor 1			
1	Emergency Department (Akutt avdeling)	**10**	Personnel service / accommodation
2	Observation	**12**	Central storage, Central kitchen
3	Emergency Department (Legevakt)	**13**	Waste, Cleaning, Garment handling
4	Imaging (1) - Treatment	**14**	Bed handling
5	Imaging (2) - Support	**15**	IT Services, Medical Tech, MOMD
6	Patient information/Reception	**16**	Ambulance (1) floor 1
7	Centre of Learning and Mastery	**31**	Sterile Central
8	Cafeteria	**44**	Oncological day unit (1)
9	Hospital Ministry	**45**	Oncological day unit (2) - Support
Floor 2			
11	Laboratories	**33**	Intensive Care (2) - Support
17	Ambulance (2) floor 2	**35**	Polyclinic - Expedition
18	Day Surgery	**36**	Polyclinic - General Somatic
24	Clinical Office - ER	**37**	Polyclinic - Ophthalmology
25	Clinical Office - Imaging	**38**	Polyclinic - Gyneacology
28	Operation	**41**	Polyclinic - Injury
29	Pre Operative	**43**	Polyclinic - Otorhinolaryngology
30	Post Operative	**55**	Obstetrics (Delivery)
32	Intensive Care (1)		
Floor 3			
23	Clinical Office - Polyclinic	**47**	Research
27	Clinical Office - Intensive Care	**51**	Bed ward - Gynecology
34	Polyclinic - Endoscopy	**52**	Patient hotel
39	Polyclinic - Lung/Cardio	**53**	Bed ward - Children
40	Polyclinic - Mental Health/ Substance Abuse Treatment	**54**	Child Habilitation
42	Polyclinic - Support	**56**	Physio, Occupational therapy, Nutrition (1)
46	Education	**57**	Physio, Occupational therapy, Nutrition (2) - Support
Floor 4			
19	Office workspace (1)	**26**	Clinical Office - Bed wards
20	Office workspace (2)	**48**	Bed ward - Medical (1)
21	Office workspace (3)	**50**	Bed ward - Orthopedics/Surgery
Floor 5			
22	Meeting rooms, Employee areas, Support rooms	**49**	Bed ward - Medical (2)

and objective value, and discussed on their appropriateness and usefulness. Some instances follow an iterative structure, while other methods are executed simultaneously for some or all of the floors. For the iterative methods, caused by the floors having different area, structure and number of functions and relations, the order in which the floors are solved have to be decided from considerations of different measures. As the hospital of the case study constitutes an extensive problem, solving the floors in steps with still ensuring information transfer between the floors is a favorable approach.

Table 8.5: Key results from tests of different approaches of stage 2

Nr.	Solution Approach	Objective Value	Computation Time [s] elapsed time for stage 1 (4.18s) is added to total						Section
			Fl. 1	**Fl. 2**	**Fl. 3**	**Fl. 4**	**Fl. 5**	**Total**	
(1)	All floors solved simultaneously (locked floor)	Best b. 8872	-	-	-	-	-	112368	8.3.1
(2)	Each floor separately	14329	254	2100	561	0.73	0.21	3104	8.3.2
(3)	F1 → $F5$	14088	254	5285	561	0.81	0.28	6107	8.3.3
	+ F1 re-run	13880	+223					6330	
	+ F2 re-run	13862		+927				7257	
	+ F3 re-run	13862			+299			7557	
	+ F4 re-run	13862				+0.52		7557	
	+ F5 re-run	13862					+0.20	7558	
(4)	F5 → $F1$	13815	312	1894	848	0.55	0.22	3060	8.3.4
(5)	F2 → $F3$ → $F1$ → $F4$ → $F5$	13880	203	2100	481	0.69	0.21	2790	8.3.5
(6)	F3 → $F2$ → $F1$ → $F4$ → $F5$	13600	8990 (2% gap)	699	561	2.49	0.20	10257	8.3.6
(7)	F1 → $F2$ → F3+F4+F5 locked floor	14088	254	5285	9838			15383	8.3.7
(8)	F1 → $F2$ → F3+F4+F5 not locked floor	13448	254	5285	23179 (1% gap)			28523	8.3.7

Following is a brief description of the eight instances, before each instance is further evaluated in sections 8.3.1 to 8.3.7. Solution approach (1) attempts to solve the whole hospital simultaneously, only based on information in which floor each function is placed on from stage 1. Solution approach (2) solves each floor separately with the information on which floors the functions on other floors are locked to. Since the floors of solution approach (2) are not solved iteratively, but independent of each other, they can be solved at the same time and the total computation time of this approach consists of the elapsed time of the floor using the longest time in addition to the computation time of stage 1. Solution approaches (3)-(8) are solved with partly or fully iterative approaches. In solution approach (3), the floors are solved iteratively, starting with floor 1, and solving each floor from the bottom to the top of the building. For each iteration, the functions of previously solved floors are locked with center and placement to the assigned locations, while functions of unsolved floors are locked to the floor assigned in stage 1, and hence relations to functions on other floors are accounted for as previously described in Section 6.

In order to examine the effect of more iterations, re-runs of the floors use the locked placements of functions in the other floors are performed. Solution approach (4) is solved in a similar way as approach (3), but starts with floor 5 and iterates down to floor 1, without re-runs of the floors. The same goes for solution approach (5) that is solved starting with the floor assigned the most functions (except the functions already locked with the center to that floor), and ends with the floor with the least functions to be placed, with no re-runs. The order of this solution approach is consequently floor 2, floor 3, floor 1, floor 4, and floor 5. Based on the results of the above-mentioned tests, as is discussed further in the sections below, examining an iterative approach starting with floor 3 is interesting. Therefore, solution approach (6) begins with floor 3, thereafter floor 2, floor 1, floor 4, and floor 5. Solution approach (7) is composed of both an iterative part, and one part where some floors are solved together. First, floor 1 and floor 2 are solved iteratively, and thereafter floor 3, 4 and 5 are solved simultaneously with the floor 1 and floor 2 locked to locations, and functions of floor 3, 4 and 5 locked to their assigned floors. Solution approach (8) is similar to solution approach (7), but without locking the functions assigned to floor 3, 4, and 5 to their respective floor. As observed from Table 8.5, the solving of floor 1 of solution approach (6) and floor 3 of solution approach (8) have not obtained an optimal solution, due to loss of memory-capacity, and the runs of the floors are therefore given by the optimality gap obtained.

8.3.1 Solving all floors simultaneously (1)

An attempt is made to solve stage 2 in one iteration, by locking the functions to their respective floors from stage 1 and solving all the floors at the same time. As shown in Table 8.5, the software available does not manage to solve this extensive problem. The best bound found is 8872, and no feasible solution is obtained. Hence, this approach is deemed inappropriate for the instance, and the following methods of solving stage 2 involves a higher degree of simplifications than only locking to floors according to stage 1.

8.3.2 Solving each floor of stage 2 separately (2)

The method with the least interweaving between the floors is solving each floor individually, only considering which functions are assigned to each floor from stage 1. Therefore, functions of each floor are allocated without information on which locations functions of other floors assigned, only on which floor each function is located. This means that the model will only to a certain degree be able to locate functions having high proximity with functions in other floors near elevators. As seen from Table 8.5, solving each floor separately gives the weakest objective value of the instances that manage to obtain a solution. This is because of the lack of consideration of relations to specific function-placements on the other floors when solving a floor.

8.3.3 Solving stage 2 iteratively from floor 1 to 5 (3)

A natural approach to allocating functions to locations on different floors is starting at floor 1, where some functions are locked to locations based on the design of the building, especially considering the importance of entrances. For the building of the case study, there are two different entrances. The main entrance and emergency entrance are located as shown in Figure 8.1. The emergency department is locked to be present in location L8, and the patient information/reception, the Centre of learning and mastery, and the hospital ministry are locked to be present in main atrium (L26). From considerations of these locked placements, the layout of floor 1 have an impact on the rest of the floors of the building and therefore optimizing the building with floor 1 as the starting point seems promising. The computation time of floor 2 stands out as the longest, which is reasonable considering the size of the floor and the number of functions to be placed on this floor.

When aiming to obtain a solution as close as possible to the optimal solution (obtained by theoretically solving the whole hospital in one stage), iterating through the floors of the hospital multiple times to handle relations between locations across floors is an option. Each floor, starting from floor 1, is re-solved with all other floors locked to the solution layout from the previous iteration. Solving floor 1 and 2 for the second time results in a solution with better objective value than the first iteration but it does not improve when re-solving floor 3 to floor 5, indicating that all floors at this point have obtained the best layout considering solution method and iteration approach of stage 2.

8.3.4 Solving stage 2 iteratively from floor 5 to 1 (4)

In addition to solving stage 2 iteratively from bottom to top, a test of solving the floors from floor 5 to floor 1 is performed to compare results. This method is expected to give a weaker result than when starting with floor 1, reasoned with floor 5 being a small and symmetric floor, and hence it seems inexpedient to use this floor as the basis for solving the other floors. Despite the expectations, this solution approach results in an objective value better than the iterative solution moving from floor 1 to 5, in shorter computation time. The reason for this is unclear; it could be a coincidence caused by the functions defined and the specifications described for the building, as well as it could be caused by the fact that the different floors have a different degree of symmetry, making some orders of floors preferable. This is further discussed in Section 8.3.6.

8.3.5 Solving stage 2 iteratively; highest to least number of functions (5)

An option of first solving the floors with the most functions to be placed (the functions that are left when not accounting for the ones locked to locations) and end with the floor with the least number of "free" functions is inspected. By placing a large number of functions in the first iteration, the foundation for solving the subsequent floors is stronger by the many already solved placements, and therefore the solution time is predicted to be shorter for the rest of the floors. This solution gives an objective value equal to approach (3) with re-solving of floor 1, in less computation time, and hence proves to be a better approach for this instance. This is also the approach with the shortest computation time of all instances tested in this section.

8.3.6 Solving stage 2 starting with floor 3 (6)

Seen from the results of approaches (3) to (5), starting to iterate at the top of the building gives the best result in the shortest computation time. However, floor 4 and 5 have fewer functions to place on a smaller number of locations than floor 1-3, making it reasonable to assume that solving these two floors is not the main reason for the good result. This leaves floor 3 as the possible candidate. Based on this, a test is performed to see if solving floor 3 before floor 1 and 2 gives a better solution for stage 2. The test does, in fact, show better results, pointing to some of the assumptions being right. However, when solving floor 1 after having solved floor

3 and 2, there is not enough computation capacity to reach optimality. Anyhow, with an optimality gap of 3%, a solution used in further iterations (floor 4 and 5) is found in reasonable time, and even though an optimal solution is desirable, it is not necessarily crucial. Solving floor 3 before floor 2 gives a significant shorter computation time of solving floor 2 than for the other approaches.

Another interesting observation is the difference in solution time of solving floor 1 for approach (5) and (6). The only difference between the two approaches is the order of floor 2 and 3. Solving floor 2 first results in computation time for floor 1 of 203 seconds, while starting with floor 3 results in computation time for floor 1 of 8990 seconds with 2% optimality gap. The symmetry of the floors could impact these findings. Floor 3 is asymmetric, while floor 2 is symmetric, and with an asymmetric foundation from floor 3 before solving floor 2, a solution on floor 1 is harder to obtain. The way the functions are locked in floor 2 and 3 in the two cases causes a significant difference in computation time when solving floor 1. Exactly why this happens is hard to provide a clear answer to, and depends on many factors like the exact composition of functions on each floor, in addition to the selection of functions on each floor and their proximity values.

8.3.7 Solving floor 1 and floor 2 iteratively, and floor 3, 4, and 5 jointly (7&8)

As a hybrid between solving stage 2 iteratively and solving the whole case in one iteration, floor 1 and floor 2 are solved as the first and second iteration, respectively, followed by solving floor 3, 4 and 5 simultaneously. Firstly, in approach (7), by locking the functions to their designated floor, and in approach (8), without locking the functions to the floors they are assigned to in stage 1, giving room for shifting functions between floors.

Solution approach (7) with functions locked to floors, generates an objective value equal to the iterative method of approach (3) (floor 1 to floor 5) without re-runs of the floors, again pointing to the previously discussed assumption that the three last floors have less impact on the result than the first two, larger floors. Solution approach (8) solves the three last floors in one iteration and obtains the best objective value seen among all the approaches, most likely caused by the flexibility given in allocating functions between floor 3, 4 and 5 after knowing the locations of the functions in floor 1 and 2. The elapsed time of both test (7) and (8) is longer than the previously tested approaches and proves that solving the three last floors in one iteration requires extensive computational capacity.

8.3.8 Discussion

Solving the whole case instance in one stage is not possible due to the complexity of the problem. Neither is solving stage 2 for all floors simultaneously if including information on which floor each function is placed on. This proves a need for a solution method with an iterative structure. Even though Hammerfest Hospital is a small hospital, the solution method of solving the problem in two stages is necessary if choosing to solve the problem with exact MILP methods. From the solution methods presented of stage 2, solving each floor separately gives the least favorable objective value, which proves the value of considering relations to functions that are placed in other floors. The advantage of solving the floors with some degree of interweaving is shown from the objective value of approaches (3)-(7) in comparison to approach (2).

When choosing the preferred approach to iterations, the objective value in relation to the computation time is considered. From Table 8.5, the objective value does not vary tremendously for the different approaches. However, for this case instance and the result of stage 1, solving floor 1 and 2 of the problem iteratively, and after that floor 3, 4, and 5 simultaneously (approach 8), gives the best result, but also requires the longest computation time.

As seen from the tests, and especially with approach (5) and (6), the order of the iterations, and hence which functions are placed in the various locations, has a great impact on the solution and computation time. Approach (4), (5) and (6) solves floor 2 and 3 before floor 1, and provide good results, while the best is obtained with approach (8) where floor 1 is solved before floor 2. This also indicates how various solutions of stage 1 regarding number and composition of functions on each floor, which functions are locked, the footprint of each floor (with or without symmetry) and various relations between the functions impact the solution method and the quality of the result of stage 2. In conclusion, the tests show that for the specified input, the solution approaches give similar objective values, being quite collateral when using them on different input values and that there is a trade-off between computation time and improvement in objective value.

8.4 Results of Case study

After observing a minimal difference in objective value between the choices of solution approaches in Section 8.3, and based on the computation times, the approach of solving the floors iteratively from floor 1 to floor 5 with re-runs of floor 1 and floor 2 is chosen as the solution approach for the rest of the case study. This is based on considerations of objective value and computation time, and as some of the results from different approaches were somewhat surprising and hence is likely to be caused by some degree of randomness both in data and iteration method, the choice was also built on a qualitative discussion of what seemed reasonable considering the aspects of the case.

In the following section, three different sets of proximity values are used to compare and analyze the functionality of the model. The case instance that is developed based on data and information from Sykehusbygg *(case instance 1)*, is in Section 8.4.2 evaluated regarding the operational aspects of the resulting layout.

8.4.1 Numerical Analysis

To analyze the performance of the model and solution approach, six case instances are created. All instances are solved in two stages, having the Emergency Department (1) locked to the emergency entrance, and the Patient Information/Reception (6), Centre of Learning and Mastery (7) and the Hospital Ministry (9) locked to the main entrance/atria. The parameters varied between the instances are proximity values and the inclusion of locking an additional set of functions to reasonable placements and floors. Table 8.6 presents an overview of which parameters are included in each case instance. Three sets of proximity values are used in the test. Proximity set 1 is created based on information retrieved from Sykehusbygg, while proximity set 2 and proximity set 3 are developed based on random values. The values of proximity set 2 are equal to the ones of proximity set 1 regarding number and range of values, but in proximity set 2 the functions are sorted randomly so that the composition of proximity values for individual functions are similar. Proximity set 3 is based on equal values as proximity set 2, but only around half the number of non-zero values.

Table 8.6: Key data of case instances

Instance	Proximity Set	Locks to floors and specifications (As in Table 8.3)
Case Instance 1	1	Yes
Case Instance 2	1	No
Case Instance 3	2	Yes
Case Instance 4	2	No
Case Instance 5	3	Yes
Case Instance 6	3	No

Even though the values of proximity set 2 and proximity set 3 are insignificant to Sykehusbygg and for evaluation on operational factors, solving the case based on different proximity values is useful when evaluating the accuracy and functionality of the model in general. All proximity sets are tested on both the locking of functions to floors described in Table 8.3, and without any locking (the mentioned locked Emergency Department, Patient Information/Reception, etc. are retained). The locking of functions to floors is developed based on what seems reasonable for the case of Hammerfest Hospital. Proximity set 1, proximity set 2, and proximity set 3 are shown, respectively, in Appendix A, proximity set 1 being the one previously used in the study of solution approaches.

All case instances are solved by first solving stage 1, and then using the solution approach of iterating from floor 1 to floor 5, and re-solving floor 1 and 2. Some of the functions defined to require window are allowed to be partly placed in locations without windows, as parts of these functions such as storage, toilets and workstations do in reality not need windows. Also, some adjustments on area increase (1-2% compared to what is presented in the case data) have been made to make the instances solvable within reasonable computation times. Increasing the area by 1-2% will not have any significant impact on the result layouts.

Evaluating the objective values of the two stages

For each case instance, stage 1 is executed to allocate functions on the different floors of the building. The resulting distribution of functions for all case instances is shown in the tables of Appendix B. As can be seen from the tables, the functions that are locked to floors have been assigned to the same floors in Case Instance 1, 3, and 5, and hence, the results of stage 1 have some similarities for these case instances. Besides, Case Instances 1 and 2, 3 and 4, and 5 and 6, respectively, have similarities because each pair is based on the same sets of proximity values.

Based on results from the technical study, for each proximity set, the expectancy is that the objective value when not including locked functions is better than when functions are locked to floors and specifications. Not locking functions to floors is expected to free the model to group high relations on the same floor and hence obtain better results when iterating over the different floors. As explained, the choices of locked floors are equal for instances 1, 3 and 5, and based on discussions with Sykehusbygg; the locked functions are in many ways in accordance with set 1 of proximity values. The objective values of stage 1 and stage 2 for the six instances are listed in Table 8.7. In stage 1, the expectancy of the instances not having locked functions achieving the best results have been met, and for each proximity set, locked functions give weaker objective value. When not being forced to place certain functions on designated floors, the model can prioritize clustering functions of high relation on equal floors, prioritizing to minimize vertical distances.

Table 8.7: Objective values in stage 1 and 2 for case instances

	Objective Value	
Case Instance	Stage 1	Stage 2
(1) Proximity 1 *With locks*	1 110	13 862
(2) Proximity 1 *No Locks*	1 089	14 572
(3) Proximity 2 *With locks*	1 365	16 175
(4) Proximity 2 *No locks*	1 085	15 619
(5) Proximity 3 *With locks*	490	6 508
(6) Proximity 3 *No locks*	435	5 691

(lighter colors indicate better objective value)

The results of stage 2 meet the expectation for the instances based on random sets of proximity values (proximity set 2 and 3). The choice of which functions defined locked are determined independent of the proximity relations in these instances, and hence all locking of functions is likely to counteract with the model's aim of locating functions based on proximity values. When solving instance 1 and 2, based on the realistic data from Sykehusbygg in proximity set 1, the opposite happens. Even though the result of stage 1 pointed to a better objective value for the instance without locked functions (Instance 2), in stage 2, instance 1, with included locked functions, performs better than instance 2 with no locked functions.

The reason for this is twofold. As the choices of which functions to lock and the proximity values are based on the same considerations (the ones from Sykehusbygg and the hospital in Hammerfest), the locked functions can comply with the proximity values in a way that is favorable for the overall result. Also, the explanation lies in the chosen solution method. Stage 1 is solved with the aim of minimizing the proximity relations between different floors. As was explained in Section 8.3, stage 1 accounts for locations as each floor being a collected location, having a certain distance to each of the other floors. This means that stage 1 will prioritize to cluster groups of high proximity relations on each floor; as the internal distances of each floor are not accounted for in this stage.

The graph in Figure 8.2 shows whether or not functions of each proximity value is placed on the same floor or not in instance 1 and instance 2. With the defined locked functions, it seems as though instance 1 has been able to collect a larger share of the function pairs having high proximity relations on the same floors, whereas the relations are slightly more evenly distributed in Instance 2. Stage 1 aims to distribute functions in a way that minimized the impact of proximity values

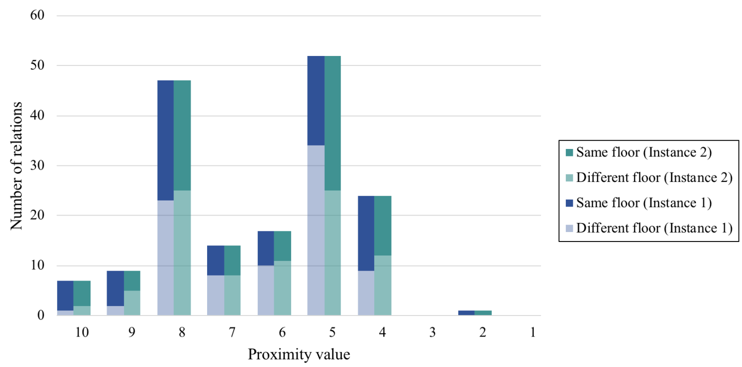


Figure 8.2: Relationships on same floor/different floor

between floors, and the total effect of inter floor-relations is smaller for Instance 2 than for the Instance 2 with locked functions. However, the distribution between floors in Instance 1 proves better after solving Stage 2.

With the solution method used for this case study, stage 2 has no opportunity to shift functions between floors after receiving inputs from stage 1. This means that even though stage 1 has performed well in distributing functions of non-zero proximity value on the same floors, stage 2 is forced to impose a certain distance between these functions, as they have to be distributed over the area of the floor they are assigned to. A certain marginalization of horizontal distances, by the natural emphasis on vertical distances between floors in stage 1, is unveiled in stage 2. This effect shows that some locations on different floors are closer together when considering distances including both vertical and horizontal part, than certain locations on the same floor. Long horizontal distances are obtained when forced to spread functions on remote locations inside one floor. However, stage 2 is designed to account for relations across floors as far as what is possible without shifting functions between floors, so the model is prioritizing placement of functions with relations on other floors near elevators.

Even though raising a discussion of vertical versus horizontal distances and the emphasis of these in the different stages, the results from instance 1 and 2 show good correlations regarding the data they are based on. It seems as the choices of specified locked functions have enhanced the overall distribution of functions considering the proximity values defined in collaboration with Sykehusbygg, even though this was not visible in stage 1.

Evaluating the performance with respect to proximity values

The layouts obtained from the instances are tested with the proximity sets of the other case instances. This is done to prove that each proximity set has created the solution layout with the lowest objective value of the case instances. Each of the layouts is locked with all centers of the solution, and the proximity values are replaced by the two other proximity sets to calculate the objective value. The results are shown in Table 8.8.

Table 8.8: Objective value of instances with different proximity values

	Objective Value of		
Case Instance	Proximity set 1	Proximity set 2	Proximity set 3
(1) Proximity 1 *With locks*	13 862	21 785	9 212
(2) Proximity 1 *No Locks*	14 572	21 717	9 649
(3) Proximity 2 *With locks*	18 497	16 175	6 663
(4) Proximity 2 *No locks*	20 351	15 619	6 189
(5) Proximity 3 *With locks*	19 005	17 857	6 508
(6) Proximity 3 *No locks*	14 626	20 251	5 691

(lighter colors indicate better objective value)

As already seen, the instance based on proximity set 1 is the best solution considering this set. Case Instance 4 has no connection to proximity values of set 1, and obtains, as expected, a poor result, similar to what instance 1 and 2 does when tested for proximity set 2. The third set of proximity values consists of fewer entries than set 1 and 2, and the objective values obtained with this set is accordingly smaller. The instances based on proximity set 2 performs well on the values of set 3 and likewise the other way around, which is a natural result of the two proximity sets being based on the same, but a different number of proximity values.

As was done in the technical study in Section 7.2.7, an analysis of the average distances between functions having different proximity values is performed. This gives insight into the ability of the model to account for the need for closeness defined between functions. For the case instances, the ones including functions locked to floors and specifications are expected, to a certain degree, to decrease the overall effect of the proximity values. This is because the model is forced to account for locked functions, which may be contradicting the proximity values. Table 8.9 shows, for each case instance, the average distance between a pair of functions having each proximity value. Also, for each proximity value, the number of function pairs that have that relation is listed.

Table 8.9: Average distance between pairs of functions of equal proximity value

	Average Distances										
Proximity values	0	1	2	3	4	5	6	7	8	9	10
# of relations	1454	0	1	0	24	52	17	14	47	9	7
(1) Proximity 1 *With locks*	19.9	–	17	–	13.4	14.6	11.6	10.4	13	13.2	8.4
(2) Proximity 1 *No locks*	20	–	17	–	15.8	15.2	15.8	16.0	11.9	11.1	7.7
(3) Proximity 2 *With locks*	19.3	–	16	–	19.2	13.9	15.8	17.6	13.3	19.5	11.1
(4) Proximity 2 *No locks*	19.3	–	17.5	–	16.5	15.8	15.9	15.5	13.6	12.7	8.7
# of relations	1558	0	0	0	12	16	4	6	22	5	2
(5) Proximity 3 *With locks*	20.1	–	–	–	13.9	19.5	11	20.7	13.4	12.0	15.5
(6) Proximity 3 *No locks*	19.4	–	–	–	20	21.9	20	25.2	16.2	18.2	17

As can be seen from the table, case instances 1-4 have a larger number of relations for each non-zero proximity value than case instances 5 and 6. This is naturally caused by proximity set 1 and 2 having a larger amount of non-zero values. The instances using proximity set 3 (Case Instance 5 and 6) does not show a trend as clear as for the instances using proximity set 1 and 2. The averaged distance of case instance 5 and 6 are based on fewer values, and therefore the performance of the model becomes less clear.

Proximity set 1 and 2 are similar in the way of having an equal amount of proximity values in total, in addition to the values being distributed in a similar pattern, possibly creating groups of functions that are naturally placed together in the building. The relationship between average distances and proximity value in the case instances 1 to 4 are illustrated in Figure 8.3.

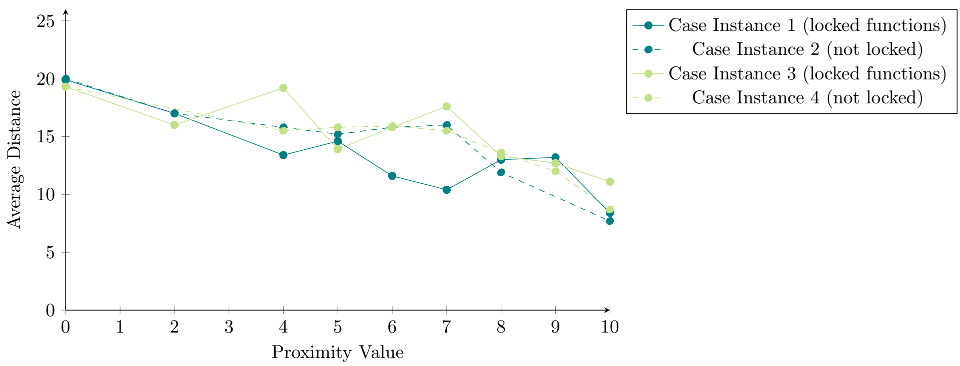


Figure 8.3: Average distances between functions of equal proximity values

The graph shows a decreasing trend for the average distances with increasing proximity value for all instances illustrated. The locked functions that Case Instance 1 and 3 are based on are developed with similar intentions as what proximity set 1 is based on. This is because both proximity set 1 and the locks to floors and specifications are part of the actual case data. An example of this is a high proximity value between Imaging (4) and Imaging Support (5), which are also locked to be either on floor 1 or floor 2. This means that in Case Instance 1, the locked functions will act in accordance with the proximity values, whereas in Case Instance 3, accounting for proximity values of set 2, the locked functions may counteract with the defined proximity values. This tendency can be seen in the graph of Figure 8.3, where the graph of Case Instance 3 is the most deviant from the decreasing trend.

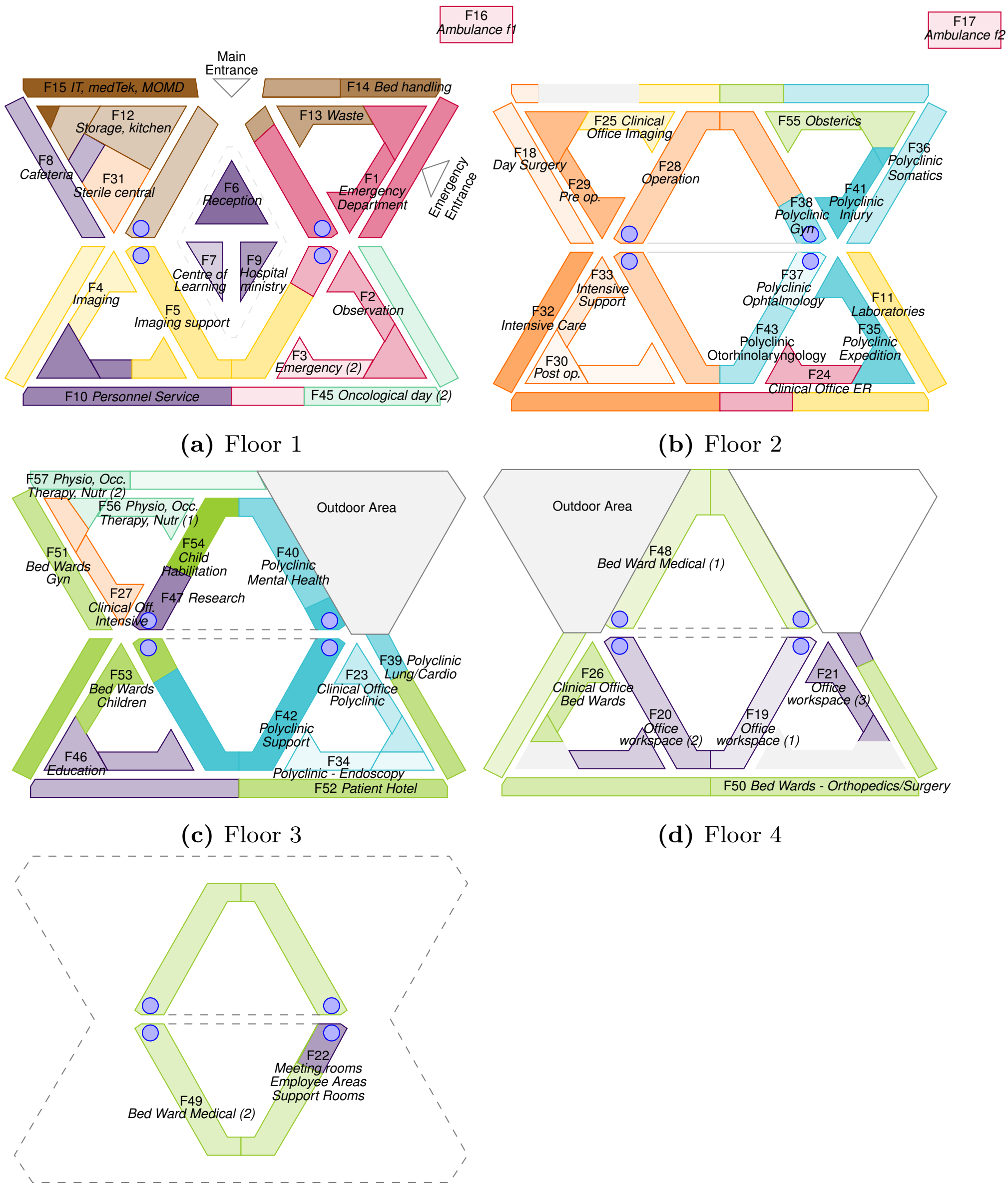
The case instances without locked functions (instances 2 and 4) show a similar graph, which is also smoother than the graphs of case instance 1 and 3, caused by the instances only being based on proximity values. Even though case instance 1 have corresponding locked functions and proximity values, the values occasionally contradict, making the graph less even than the one of instance 2.

8.4.2 Operational aspects

In addition to observing the mathematical relationship between the input parameters and the resulting layout, observations based on operational aspects of the hospital are of great value. These aspects include the internal distances between functions and the relative placements of functions deemed important to each other throughout the floors and on each floor. After using the model on the case data from Hammerfest Hospital, the suggested layout have been presented to Sykehusbygg, and their comments on functionality and suggestions for improvement are discussed below. Only the layout of the real case data is displayed in this section because this layout is the only one of hospital relevance. Appendix C show the center placements of each function on their respective floors for each of the six case instances.

The solution method used is the iterative approach that iterates from bottom to top, with re-solving floor 1 and 2. In the following, the obtained resulting layout for the case from Hammerfest shown in Figure 8.4 (the same figures displayed in larger sizes are shown in Appendix D) is discussed on operational aspects, both observing the effect of proximity values and locks, and from the comments made by Sykehusbygg during a meeting held to present the results (Sykehusbygg HF, 2018a).

Floor 1 (ground floor) of the hospital building houses many vital functions such as the two emergency functions (1) and (3), Imaging (4) and (5), and the Ob-



(a) Floor 1

(b) Floor 2

(c) Floor 3

(d) Floor 4

(e) Floor 5

Figure 8.4: Result Layouts of each floor for proximity set 1

(grey locations indicate no placement of functions)

servation (2). Several of these functions are required to be placed at floor 1 due to the emergency- and ambulance entrance and other functions, such as the Patient Information/Reception (6) needs to be in the immediate vicinity of the main entrance. The locations of floor 1 are in many ways the most valuable and accessible locations in the hospital, and there is a wish to place many functions on this floor. In the result, the Cafeteria (8) is placed in a nice location with windows, and the Personnel Service/Accommodation (10) is in proximity to the emergency functions, allowing employees to be available if an urgent situation should arise while resting between their shifts.

The Oncological Day function (44) with support function (45) are placed in a somewhat shielded corner of the building with windows, a nice placement considering that many patients spend hours at a time, a few time a week receiving treatment in this department. A group of functions that occupies valuable space in this first floor is the supporting functions Central Storage (12), Waste and Cleaning (13), Bed Handling (14) and IT Services (15). Ideally, according to Sykehusbygg, the building should have a basement where these functions could have been placed, but as a basement is considered too expensive, and as these functions require access to delivery of goods entrance, they have to be placed at the ground floor. This gives these non-medicine functions locations near the Emergency Department (1), which is not desirable, but these functions could easily be shifted towards the locations on the upper left corner in the figure, and be replaced with more relevant functions like for example the Sterile Central (31).

Sykehusbygg mentioned that the Imaging (4) and Imaging Support (5) could have switched places in order to give better access between the emergency functions and imaging. As of now, the proximity values from the emergency departments are equal to both Imaging and Imaging support, even though there might be a greater need for proximity from other functions to the Imaging than to the support function.

Sykehusbygg found the layout of floor 2 of the hospital convenient and well-designed. Especially the fact that many of the different Polyclinics (35-38), (41) and (43) are centered around the Polyclinic Expedition (35) near the elevator, giving patients arriving through the main entrance easy access to the polyclinics during daytime. In addition to this, the surgical departments (Operation (28), Pre- and Post Operative (29 and 30) and Intensive Care (32 and 33)) are grouped and located at this floor, giving easy access to emergency departments on the floor below. Sykehusbygg commented that when Day Surgery (18) is located at the same floor as the polyclinics, the expedition and rooms of the polyclinics could be used collaboratively, by serving as rooms for pre-operational conversations and preparations of patients having day surgery, an effect not incorporated in existing plans, but suggested after seeing the layout of the floor.

On floor 3, the rest of the polyclinics (34),(39) and (40) are located, near the Polyclinics support (42), and the Polyclinics Clinical Offices (23). All these functions are located at the same half of the building as the polyclinics located in the floor below, meaning that patients can easily access these different functions by elevator. Sykehusbygg approves this solution, and the collection of polyclinics in the same areas is also what has been desirable when planning the layout of Hammerfest Hospital. The Bed Ward Children (53) is located on this floor, same as the Patient Hotel (52) and the Bed Ward Gynecology (51). All of these functions require windows, forcing them to be further apart from each other than what was the original intention with the proximity values. When discussing the layout of this floor, one could easily argue that the Patient Hotel could switch places with the Student Area (46), but as the input into the model is that the Patient Hotel needs windows, the locations where the Student Areas are placed are not available for placement of the Patient Hotel. This is one of the effects of modeling with strict mathematical constraints and requirements. However, as is seen with most of these adjustments, they are easily found when studying the layouts in hindsight, which is the purpose of the model.

The two upper floors of the hospital consist mainly of Bed Wards (48-50) and Office Spaces (19-22) due to a large number of locations with windows, these floors are well suited for bed wards, and combined with offices occupying the locations facing inward towards the atria, the layouts of these floors are expedient and well distributed according to Sykehusbygg. The only comment or suggestion for improvement of these floors is to switch places between the Bed Ward Medical 2 (49) located in floor 5 with the Bed Ward Orthopedics/Surgery (50) in floor 4; locating the two medical bed wards adjacent, and still keeping the bed ward of orthopedics/surgery close to elevators with easy access to the surgical departments of the lower floors. The placement of these functions in the layout is an example of the areas of functions and locations being evaluated strictly (with the limits allowed), preventing the medical bed wards to be located together, as there are not enough total area. The area of Bed Wards Medical 2 (49) and Bed Wards Orthopedics/Surgery (50) could go through some adjustments to allow the medical bed wards to be on the same floor (the functions may even require less area when placed together because of overlapping equipment/spaces), and some parts of the functions (workstations for the nurses and doctors) can also be placed in locations without windows.

In discussing the layouts, Sykehusbygg finds the resulting layouts of the model to be a valuable contribution to discussions when planning hospitals, and is positive to the insight layouts based on actual input data can give in addition to the manual methods already used. Different approaches on how to use the outputs have been discussed, among them using the model as a tool for generating different layouts emphasizing different views on proximity and presenting them to the representatives of the employees and the other stakeholders. The goal is, as previously mentioned, to use the output of this model (layouts) to create a decision-making tool for creating discussion and open up for new possibilities in hospital planning.

Looking at specific layouts that the model created in the meeting with Sykehusbygg caused the meeting quickly evolve into discussions around the placement of specific functions and how to better utilize locations. The discussion in the meeting quickly developed into a discussion on placements of functions in the building and in relation to each other, a good signal considering the purpose of the model, and is therefore likely to cause the same effect for the stakeholders on hospitals to be built. There are many human aspects of hospital planning that can not be replaced by a mathematical model. The model as a tool can have potential to be used as inputs and suggestions open for discussion. The model can (relatively) quickly produce possible layouts based on different inputs, and could, therefore, be a valuable contribution in the room plans used by Sykehusbygg to evaluate and decide the areas, number and composition of rooms for each department, and functions of the hospital. The purpose of using the model is to create the layouts and afterward make alterations that consider other aspects that not already included, and then, together with the architects, adjust the functions of the layout to discover new solutions. Also, a comment from Sykehusbygg is that the model can be used as an evaluation of existing hospital buildings and suggest ideas for improvement.

8.4.3 Discussion

Six case instances are created based on three different sets of proximity values, and two variations of whether or not functions are locked to floors and specifications. For stage 1, the case instances with no locks of functions to floors give better objective value than the corresponding instance having some functions locked. Stage 1 does not consider any distribution of the functions inside each floor, and hence the functions allocated to the same floor have no distance between them. When placing functions of high relatedness on the same floor in stage 1, the functions have no contribution to the objective value. Instances with certain functions locked to floors can only obtain the same, or a higher objective value in stage 1 than the instances without these locked functions.

If all the instances had been solved in one single iteration, considering all floors, locations, vertical- and horizontal distances and functions at once, an instance with functions locked to floors could not have obtained a better objective value than an instance based on the same proximity values without locked functions. As this solution method is not an opportunity due to the size of the case, the solution has to be done in iterations. This forces a split in consideration of different aspects, stage 1 focusing on horizontal distances, causing stage 2 to have limited possibilities in emphasizing these distances other than locating functions near elevators.

Stage 2 allocate functions to locations on each of the floors. For the case instances based on proximity set 2 and 3, the objective value of the instances without functions locked to floors give the best objective value. However, after solving case instances 1 and 2, using proximity set 1, instance 2 with certain functions locked to floors gives a better objective value. When analyzing the result of stage 1 compared to the considerations made with locked functions and proximity values, even if a bit random, this seems reasonable. For proximity 1, the values defined in collaboration with Sykehusbygg, the instance that included the locked functions performed best. Which functions to lock is also decided in collaboration with Sykehusbygg, and hence it seems as the two aspects enhanced each other, leading to better results than what is obtained when the functions are freer to be located. Locking functions lead to an expedient distribution between the floors, enabling more relations to be emphasized through the extended use of closeness between different floors.

When testing the layouts of each instance on the proximity values of the other instances, the analysis shows that for each set of proximity values, the layouts initially based on these parameters give the best results. In general, the instances without locked functions show the best accordance to the proximity values and hence obtain the best objective values, with the exception being case instance 1, performing better than instance 2.

From the analysis of testing the performance of the model in regards to placing functions having a high need for relation nearby each other, the model performs very well. There are apparent correlations between high proximity values and shorter distances between functions. This is most clear for the case instances that are based on the most proximity values (Case Instances 1-4). Also, the instances that are based on no locking to floors have the smoothest graphs, with the instances with locks being more deviating as a result of possibly contradicting or pointwise enhancing locks and proximity values.

The conducted case study aims to show the functionality of the model of this thesis on instances with close to realistic input, in size of building, number of locations and functions and in relations between the functions. In general, the layout obtained is evaluated as a good fit considering operational aspects. The feedback from Sykehusbygg on the obtained layouts shows a good ability of the model to account for important operational aspects. In a planning process, all layouts are supposed to be evaluated and restructured by Sykehusbygg, and the layouts are considered a good basis for discussions.

Certain traits of the model and their impact on operational aspects were noted. Some of the functions are divided into the main part, and a supporting part comprising storage, meeting rooms, toilets, and other support rooms. In deciding the proximity values, the two parts have been given equal proximity relations to all other functions that they are related to, in addition to the high proximity between them. The result of this is occasionally the support function being given a more favorable location than the main function, caused by area and the locations available. In general, the main function is the one that should be given the most convenient location of these two; hence the proximity value of this function should be emphasized. However, the location of these functions can easily be corrected in hindsight and does therefore not constitute any considerable trouble.

The bed wards of floor 4 and 5, could have benefited from switching places. The placement of these functions in the layout obtained from the model is an example of the areas being evaluated strictly, as there are not enough available area in floor 4 for both of the medical bed wards. In reality, this could easily have been adjusted, according to Sykehusbygg, but the effect is a result of having to decide on a certain limit for areas in maintaining the right accuracy of placement of functions overall. This leads back to the discussion of the defined area increase, where the value chosen is the one deemed appropriate in obtaining good layouts.

On floor 3, a seemingly noticeable improvement could be for the Patient Hotel to switch places with the Student Area, giving both the functions more closeness to favorable other functions. However, as the model is told that the Patient Hotel needs windows, the location of the student areas is not suited as it does not have windows over the entire location. By a quantitative evaluation, the patient hotel could easily work with some of the area having only windows out to the atrium, but as the window requirements strictly constrain the model, this is not an obtainable solution directly. Again, these adjustments are also easily corrected when discussing the layout afterward. An alternative approach would have been for the requirements for windows could have been given specifications on the share of the function that needs to fulfill this condition.

The process of collaboration with Sykehusbygg has been done in a certain amount of iterations (meetings). The possibilities mentioned above for improvement could have been subject to changes if more iterations of discussions and improvements had been executed. However, as mentioned, the corrections were easily observed on the result layouts, and are therefore easily adjustable manually. Overall the layout of the hospital performs well based on the input parameters, which proves the usefulness of applying the model on real-world cases.

Chapter 9

Concluding remarks

The aim is for the model of this thesis to work as decision support, producing hospital layouts that can contribute in discussions and suggest solutions based on different input. Different layouts can be evaluated numerically as well as on an operational basis, giving new insight into existing suggestions and ranking the layouts produced.

This master's thesis has illustrated the use of mathematical optimization in the layout planning of hospitals. The model is able to generate suggestions of appropriate internal layouts of a hospital building for use in discussions and as direct input in the planning of the final layout. Proximity values that handle the requirement for relatedness between functions are used as the measure for minimizing distances between functions that have important and frequent interactions. The model has features of the Quadratic Assignment Problem (QAP) and handles the relations between pairs of functions that are placed in different locations of the hospital building. The nature of the QAP creates complex combinatorial problems that escalate in size with small increases in parameters of the problem. Instances tested with the model of this thesis are proved to be hard to solve to optimality with the available software when reaching a certain size. Therefore, a decomposition of the solution method of solving the allocation process of functions in two stages has been proposed as an appropriate solution method.

Stage 1 of the solution method allocates functions to floors concerning the proximity values and specifications on whether the functions need to be placed at certain floors or locations of the building. After the initial allocation, each floor is solved in an iterative or partly simultaneous manner, handling relations to other floors directly for the functions already placed and as relations to whole floors for the functions only assigned a floor. Through the calculation of distances, the decision of which elevator to use is included in the shortest-path algorithm, making the model able to handle several elevators. Different approaches to solving stage 2 of the problem are tested and compared and shows that including some interweaving between the floors when solving each floor creates a better objective value than when handling each floor separately. The two-stage approach suggested satisfies the desire of solving the layouts with sufficient performance in reasonable time.

The planning of a hospital is an extensive process. The need for immediate results from the mathematical model are not deemed prominent, and producing possible layouts within a reasonable time, such as within a day, is thought to be sufficient regarding computational time. However, if the planning and evaluation of layouts are to be executed with several iterations to include as detailed input as possible efficient methods for retrieving results are favorable. The choice of detailing in both the locations and functions profoundly impact the solution time. Evaluating the complexity of the problem of this thesis, the method of solving the problem in two stages seems appropriate considering the case instance and the level of detail chosen, as the problem is by this solvable in a reasonable time with the available software.

With the model of this thesis, layouts can be developed based on different considerations and inspected by the hospital planners. When planning hospitals today, possible layouts are developed manually without quantification of the performance of the layouts, and the discussions are majorly based on qualitative considerations. By summarizing needs for interactions in prioritized values of proximity between pairs of functions, the model generates possible layouts based on mathematical optimization. The definition of values of proximity as a summary of needed relations and current interactions between functions are thought to be a conceivable measure and an easy way of prioritizing locations of functions when lacking accurate flow data.

The tests performed throughout this thesis show a clear connection between values of proximity and the resulting optimal solution layouts, both illustrated by comparing the objective values across various solutions and analyzing each layout by the average distances between functions of certain proximity values. Also, the layout created based on real input data from Sykehusbygg shows good results related to operational aspects.

The model of this thesis accounts for varying footprints independent of geometry and size and handles the allocation of functions of different sizes and requirements. The modeling choices regarding the footprint consist of locations that can hold more than one function by different fractions, and functions that can be spread across different locations. To enable this sensibly, a set of adjacent locations are defined for each location, over which functions can be spread and remain compact and collected. As far as the literature study of this thesis is concerned, this approach is unique in the field of Hospital Layout Planning and Facility Layout Planning.

The traits of the model make it well suited for modeling a wide variety of buildings, accounting for different requirements and needs of specific functions. The model as formulated in this thesis does not depend on a quadratic division of either the footprint or the functions as seen in the majority of the literature and is independent of the geometry of the locations matching the configuration of functions to be placed.

Additional iterations of discussions with Sykehusbygg and following improvements and adjustments of the model could make the results even more accurate considering the case in question, but many of the modifications suggested could also easily be a part of the manual revision of the obtained layouts. As mentioned, the goal is not to design a tool that replaces the way hospitals are planned today. The model works according to the considerations in the form of input data from Sykehusbygg. Quantified results cannot give a full picture including all human aspects of deliberation. Even though the generated layouts will not be the finished plans for the hospital, Sykehusbygg HF can use the model as a decision support tool. As of now, considering the planning of Hammerfest hospital, the obtained results demonstrated good abilities as a contribution to discussions, and hence fulfill the purpose proposed in collaboration with Sykehusbygg.

Chapter 10

Future research

In this master's thesis, a facility layout problem has been applied to the hospital environment. The goal of this thesis has been to formulate and implement a model that can produce internal layouts able to facilitate the planning process and work as quantitative input to decision making. Considering these purposes, the model meets the expectations. However, there are still possible extensions to the implementation and solution method open for exploration, where some are listed below.

The formulation of the area of the building and the allocation of functions have been done in a way that combines binary and continuous decisions, with the placement of a function in a location being binary, while the share of the functions present in each location is continuous. The locations are connected through the nodes deemed their centers, and the distances to all other parts of the building are calculated from this node. This implies that even though the locations may be spread over a certain area, the distances to other locations are calculated from only the center of the location. A distance calculation with a more detailed approach could be developed.

In standard Facility Layout Problems (FLP), a placement cost is usually included, especially when considering large industrial installations and facilities. A similar placement costs could easily be relevant in hospital layout planning. In this thesis, the focus has been directed towards the relatedness measure, and all placement costs are assumed equal, and hence neglected in the objective function. This placement cost could be a valuable contribution in considering economic aspects of such problems to a more significant extent.

A general issue with quadratic assignment problems is the immense size obtained when formulating realistic problems with a significant amount of input data. A solution method decomposing the problem into stages is proved efficient in this thesis. Further development of this method could include extensive evaluation of the iterations, the order in which subproblems (i.g. floors) are solved and the use of combinations of simultaneous and iterative approaches. Besides this, other heuristic methods are considered worth exploring. This is based partly on the high number of models in literature solved by heuristic approaches, and on the belief that based on the nature of the problem, such methods can efficiently produce good solutions in a short amount of time. Different types of genetic algorithms are deemed relevant as parts of solving the problems and thought to allocate functions to floors and throughout floors in a targeted and systematic way. Including some form of switching of functions between floors after the functions are assigned to floors have the potential of obtaining solutions closer to optimal than solving the problem in two separate stages has. This can be seen by the solution approach in Section 8.3 that solved three of the floors in question simultaneously and without boundaries on the floors.

As for operational considerations, for the layout to better fit the hospital in question, including a more detailed set of considerations in the data, more iterations with Sykehusbygg and possibly other actors involved in the construction project is desirable. Layouts and input data could be more thoroughly inspected and processed, which in turn will lead to layouts even more accurate for Sykehusbygg to use in their discussions. Also, if a better database of data considering flows of patients, nurses, doctors, and material could be established, proximity values could be substituted with more detailed data, and give an even better picture of which functions are in requirement of closeness. However, the placement of functions can only change to a certain degree, and there is a limit to the improvement possible to obtain, as an improvement in one direction may impair other aspects of the solution.

When planning a new hospital, flexibility of locations may be desirable. Functions of a hospital are often able to align and cooperate on the use of certain facilities, such as some examination rooms, waiting areas, or meeting rooms. Such flexible locations are valuable when planning for various future changes that create new demands for the hospital, as these locations can easily be switched between functions that increase or decrease in size or change in demand for locations. In the hospital layout problem of this thesis, the need for flexibility can be included by defining the data of the functions and the locations conveniently; creating the mentioned flexible locations shared by two or several functions. To do this, additional discussions and insight into functions and their specific needs are required. Such changes would, in any case, have to be considered in relation to a convenient degree of detailing of functions.

The model accounts for special requirements of functions, such as the need for windows or need for placement on, or avoidance of, certain floors. Revealed by the evaluation of the layouts from the case based on Hammerfest hospital, the constraints induced on the model by these requirements are occasionally too strict considering the realistic operational aspects. Even though a function is defined to require windows, this is rarely the case for the entire area of the function. Modifications of constraints could be implemented as a way of nuance the functionality of the model even more, for instance by demanding only a certain fraction of the function to be placed in window locations.

Another way of including a more diverse form of locking functions is to include a higher degree of detailing of both functions and locations in the data set. By splitting the bed wards into several different parts, for instance in patient rooms, workstations, storage rooms, kitchenettes and toilets, some of the functions can have demands of windows, while other functions have other specifications. It is in this case important to ensure that all the parts of the functions are placed adjacently, by giving the functions high values of relations and/or locking the functions to be in the same parts of the building. The question of detailing of the problem is also something that should be considered in further research. The degree of division of functions needs to be determined depending on which level of planning the actors using the tool desires. Further division into more and smaller locations will give a more detailed distance calculation basis, but will also increase the size and complexity of the problem, and may require other solution methods.

Bibliography

Afrazeh, A., Keivani, A., and Farahani, L. N. (2010). A new model for dynamic multi floor facility layout problem. *Advanced Modeling and Optimization*, 12(2):249–256.

Ahmadi, A. and Jokar, M. R. A. (2016). An efficient multiple-stage mathematical programming method for advanced single and multi-floor facility layout problems. *Applied Mathematical Modelling*, 40(9):5605–5620.

Ahmadi, A., Pishvaee, M. S., and Jokar, M. R. A. (2017). A survey on multi-floor facility layout problems. *Computers & Industrial Engineering*, 107:158–170.

Arnolds, I. and Nickel, S. (2015). *Layout Planning problems in health care, Applications of Location Analysis*, pages 109–152.

Balakrishnan, J., Cheng, C. H., Conway, D. G., and Lau, C. M. (2003). A hybrid genetic algorithm for the dynamic plant layout problem. *International Journal of Production Economics*, 86(2):107–120.

Bazaraa, M. S. (1975). The quadratic set covering (assignment) problem: Applications and computation. *Unpublished manuscript.*

Bernardi, S. and Anjos, M. F. (2013). A two-stage mathematical-programming method for the multi-floor facility layout problem. *Journal of the Operational Research Society*, 64(3):352–364.

Braglia, M., Zanoni, S., and Zavanella, L. (2003). Layout design in dynamic environments: Strategies and quantitative indices. *International Journal of Production Research*, 41(5):995–1016.

Braut, G. S. (2015). Helseforetak (English: Health Institutions). *Store Medisinske Leksikon: sml.snl.no/helseforetak (2017-12-17).*

Burkard, R. E. (1984). Quadratic assignment problems. *European Journal of Operational Research*, 15(3):283–289.

Che, A., Zhang, Y., and Feng, J. (2017). Bi-objective optimization for multi-floor facility layout problem with fixed inner configuration and room adjacency constraints. *Computers & Industrial Engineering*, 105:265–276.

Christensen, H. I., Khan, A., Pokutta, S., and Tetali, P. Multidimensional bin packing and other related problems: A survey.

Drira, A., Pierreval, H., and Hajri-Gabouj, S. (2007). Facility layout problems: A survey. *Annual Reviews in Control*, 31(2):255–267.

Elshafei, A. N. (1977). Hospital layout as a quadratic assignment problem. *Journal of the Operational Research Society*, 28(1):167–179.

Farahani, R. Z., Asgari, N., Heidari, N., Hosseininia, M., and Goh, M. (2012). Covering problems in facility location: A review. *Computers & Industrial Engineering*, 62(1):368–407.

Helber, S., Böhme, D., Oucherif, F., Lagershausen, S., and Kasper, S. (2016). A hierarchical facility layout planning approach for large and complex hospitals. *Flexible Services and Manufacturing Journal*, 28(1-2):5–29.

Helse- og omsorgsdepartementet (2017). Oversikt over landets helseforetak (English: Health Institutions in Norway). *Regjeringen, www.regjeringen.no/no/tema/helse-og-omsorg/sykehus/innsikt/oversikt-over-landets-helseforetak/ (2017-12-17).*

Kaku, B. K., Thompson, G. L., and Baybars, I. (1988). A heuristic method for the multi-story layout problem. *European Journal of Operational Research*, 37(3):384–397.

Koopmans, T. C. and Beckmann, M. (1957). Assignment problems and the location of economic activities. *Econometrica: Journal of the Econometric Society*, pages 53–76.

Kouvelis, P., Kurawarwala, A. A., and Gutierrez, G. J. (1992). Algorithms for robust single and multiple period layout planning for manufacturing systems. *European Journal of Operational Research*, 63(2):287–303.

Kusiak, A. and Heragu, S. S. (1987). The facility layout problem. *European Journal of operational research*, 29(3):229–251.

Kvillum and Vigerust (2017). A One-Stage Approach to the Facility Layout Problem in Hospital Planning. *Norwegian University of Science and Technology, Institute of Industrial Economics and Technology Management.*

Lodi, A., Martello, S., and Monaci, M. (2002). Two-dimensional packing problems: A survey. *European Journal of Operational Research*, 141(2):241–252.

Meller, R. D., Narayanan, V., and Vance, P. H. (1998). Optimal facility layout design. *Operations Research Letters*, 23(3):117–127.

Moseley, L. (1963). A rational design theory for planning buildings based on the analysis and solution of circulation problems. *Architects' Journal*, 138:525–37.

Murray, A. T., Tong, D., and Kim, K. (2010). Enhancing classic coverage location models. *International Regional Science Review*, 33(2):115–133.

Rambøll Norge AS (2018). Correspondance with Rambøll Norge AS v/Siri Rørholt, Specialist Manager.

Shayan, E. and Chittilappilly, A. (2004). Genetic algorithm for facilities layout problems based on slicing tree structure. *International Journal of Production Research*, 42(19):4055–4067.

Singh, S. P. and Sharma, R. R. (2006). A review of different approaches to the facility layout problems. *The International Journal of Advanced Manufacturing Technology*, 30(5-6):425–433.

Statistisk Sentralbyrå (2017). Nøkkeltall helse (English: Key figures for health). *Statistisk Sentralbyrå, www.ssb.no/en/helse/nokkeltall/health–245784 (2017-12-17).*

Sykehusbygg HF (2017a). Correspondance with Sykehusbygg HF Autumn 2017.

Sykehusbygg HF (2017b). Nye Hammerfest Sykehus, Idéfaseapport. (English: The new hospital of hammerfest, idea phase report). *Idéfasestudie.*

Sykehusbygg HF (2017c). Sykehusbygg HF Purpose. *http://sykehusbygg.no/om/formal/ (2017-11-20).*

Sykehusbygg HF (2018a). Correspondance with Sykehusbygg HF Spring 2018.

Sykehusbygg HF (2018b). Nye Hammerfest Sykehus, Konseptrapport. (English: The new hospital of hammerfest, concept report). *Konseptstudie.*

Sykehusbygg HF (2018c). Sykehusbygg HF Prosjekter (English: Sykehusbygg HF, Projects). *http://sykehusbygg.no/prosjekter/ (2018-05-29).*

Tari, F. G. and Neghabi, H. (2015). A new linear adjacency approach for facility layout problem with unequal area departments. *Journal of Manufacturing Systems*, 37:93–103.

Tompkins, J. A., White, J. A., Bozer, Y. A., and Tanchoco, J. M. A. (2010). *Facilities planning*, pages 18–24.

Vos, L., Groothuis, S., and van Merode, G. G. (2007). Evaluating hospital design from an operations management perspective. *Health care Management Science*, 10(4):357–364.

Appendices

Appendix A

Proximity Values for the Case Study

Below, the proximity values are presented as values between pairs of functions. No value indicates zero. Table A.1 dispays the proximity values of proximity set 1, while Table A.2 displays both proximity set 2 and proximity set 3. Proximity set 3 are the values in bold and with colored column.

Table A.1: Proximity values for Proximity set 1 of the case study

f/f	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57
1		10	10	8	8					5						8								8				5	5			5	5								8														5		
2			8	8	8											8								8				5	5			5	5								8														5		
3				9	9						7																																														
4					10													8							8			8	6	8	8	8									8																
5																		8							8			8	6	8	8	8									8																
6							5	4																											6									4	4												
7																																																									
8												6																																													
9																																																4	4						4		
10																													4																										4		
11																																			4	4			4		4																
12													4	4	4																																										
13														9	4																																										
14															4																																										
15																																																									
16																																																									
17																																																									
18																												8	9	8	8	8	8																								
19																																																									
20																																																									
21																																																									
22																																																									
23																																		6	6	6	6	6	6	6	6	6	6														
24																																																									
25																																																									
26																																																9	9	9	7		7				
27																																7	7																								
28																													8	8	8	8	8																								
29																														8	8	8	8																								
30																															8	8	8																								
31																																8	8																								
32																																	10																								
33																																																									
34																																			5	5	5	5	5	5	5	5	5														
35																																				7	7	7	7	7	7	7	7														
36																																					5	5	5	5	5	5	5														
37																																						5	5	5	5	5	5														
38																																							5	5	5	5	5												7		
39																																								5	5	5	5														
40																																									5	5	5											4		6	6
41																																										5	5														
42																																											5														
43																																																									
44																																													10												
45																																																									
46																																															2										
47																																																									
48																																																	10	8	5						
49																																																		8	5						
50																																																			5						
51																																																							6		
52																																																					9		9		
53																																																						8			
54																																																									
55																																																									
56																																																									10
57																																																									

Table A.2: Proximity values for Proximity set 2 and 3 of the case study

f/f	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57
1																10																**9**									5	**4**													8		
2											8			8			8	**10**			5				8						8						5	5		5				5					8								
3											**8**										8	**8**			**8**									**8**				**8**		8				**8**													
4															2																																										
5																			10																																			**6**			
6										6							6																			6									6		6			6	6	6	6	6			
7																		**5**																			4	4																			
8																										**4**	**4**		**4**																												
9																					7																			7																	
10																	5																		**4**	5									5		**5**			5	5	**7**	5	5			
11																	8	**8**			8	8			**10**						**9**			8				6						8												8	
12																																					9									9											
13																				4										4													4														
14																		**8**																																							
15																																																									
16																																**9**									5	**4**													8		
17																		**8**							**8**										**4**	**5**									**5**		**5**			5	**5**	**7**	5	**5**			
18																					5				8						10						5	5		5				5					8								
19																																																						**6**			
20																														4													4														
21																						**8**			**8**									**8**				**8**		10				**8**				4									
22																									**8**									**8**				**8**		8				**8**													
23																																																									
24																																																**4**									10
25																															**9**			8				6						8												8	
26																											**4**		**9**																												
27																												**6**	4																												
28																																																**4**									
29																																																									
30																																											4														
31																																			7																						
32																																									7					7									9		
33																																																**5**									
34																																						9		8				8													
35																																																			4	4					
36																																					7								**5**		**5**			5	5	**7**	5	5			
37																																									**6**	**4**															
38																																								8				**8**		8											
39																																														**8**								**4**			
40																																												**8**													
41																																																							**5**		
42																																																									
43																																																									
44																																																									
45																																															**5**			5	5	**5**	5	5			
46																																																									
47																																																		5	5	5	5	5			
48																																																				6					4
49																																																									
50																																																			**5**	**7**	**5**	**5**			
51																																																				**7**	5	5			
52																																																					7	7			
53																																																						**5**			
54																																																									
55																																																									
56																																																									
57																																																									

Proximity set 2 includes all the values in the table, Proximity set 3 includes values in bold (and marked with color)

Appendix B

Results of Stage 1 of the Case Study

Following, the distribution of the functions on floors for stage 1 of case instances 1-6 are presented. These distributions are used as input in stage 2 for each instance, respectively.

Table B.1: Functions assigned to floor 1-5 in Case Instance 1

(Proximity 1, locked to floors/specifications)

Floor 1			
1	Emergency Department (Akutt avdeling)	**10**	Personnel service / accommodation
2	Observation	**12**	Central storage, Central kitchen
3	Emergency Department (Legevakt)	**13**	Waste, Cleaning, Garment handling
4	Imaging (1) - Treatment	**14**	Bed handling
5	Imaging (2) - Support	**15**	IT Services, Medical Tech, MOMD
6	Patient information/Reception	**16**	Ambulance (1) floor 1
7	Centre of Learning and Mastery	**31**	Sterile Central
8	Cafeteria	**44**	Oncological day unit (1)
9	Hospital Ministry	**45**	Oncological day unit (2) - Support
Floor 2			
11	Laboratories	**33**	Intensive Care (2) - Support
17	Ambulance (2) floor 2	**35**	Polyclinic - Expedition
18	Day Surgery	**36**	Polyclinic - General somatic
24	Clinical Office - ER	**37**	Polyclinic - Ophthalmology
25	Clinical Office - Imaging	**38**	Polyclinic - Gyneacology
28	Operation	**41**	Polyclinic - Injury
29	Pre Operative	**43**	Polyclinic - Otorhinolaryngology
30	Post Operative	**55**	Obstetrics (Delivery)
32	Intensive Care (1)		
Floor 3			
23	Clinical Office - Polyclinic	**47**	Research
27	Clinical Office - Intensive Care	**51**	Bed ward - Gynecology
34	Polyclinic - Endoscopy	**52**	Patient hotel
39	Polyclinic - Lung/Cardio	**53**	Bed ward - Children
40	Polyclinic - Mental Health/ Substance Abuse Treatment	**54**	Child Habilitation
42	Polyclinic - Support	**56**	Physio, Occupational therapy, Nutrition (1)
46	Education	**57**	Physio, Occupational therapy, Nutrition (2) - Support
Floor 4			
19	Office workspace (1)	**26**	Clinical Office - Bed wards
20	Office workspace (2)	**48**	Bed ward - Medical (1)
21	Office workspace (3)	**50**	Bed ward - Orthopedics/Surgery
Floor 5			
22	Meeting rooms, Employee areas, Support rooms	**49**	Bed ward - Medical (2)

Table B.2: Functions assigned to floor 1-5 in Case Instance 2

(Proximity 1, not locked to floors/specifications)

Floor 1			
1	Emergency Department (Akutt avdeling)	**18**	Day Surgery
2	Observation	**24**	Clinical Office - ER
4	Imaging (1) - Treatment	**25**	Clinical Office - Imaging
5	Imaging (2) - Support	**27**	Clinical Office - Intensive Care
6	Patient information/Reception	**28**	Operation
7	Centre of Learning and Mastery	**29**	Pre Operative
9	Hospital Ministry	**33**	Intensive Care (2) - Support
10	Personnel Service/Accommodation	**44**	Oncological day unit (1)
16	Ambulance (1) floor 1	**45**	Oncological day unit (2) - Support
Floor 2			
3	Emergency Department (Legevakt)	**41**	Polyclinic - Injury
17	Ambulance (2) floor 2	**42**	Polyclinic - Support
23	Clinical Office - Polyclinic	**43**	Polyclinic - Otorhinolaryngology
30	Post Operative	**48**	Bed Ward - Medical (1)
31	Sterile Central	**49**	Bed Ward - Medical (2)
32	Intensive Care (1)	**55**	Obstetrics
35	Polyclinic - Expedition		
Floor 3			
8	Cafeteria	**39**	Polyclinic - Lung/Cardio
11	Laboratories	**47**	Polyclinic - Mental Health/ Substance Abuse Treatment
26	Clinical Office - Bed Wards		
34	Polyclinic - Endoscopy	**50**	Orthopedics/Surgery
36	Polyclinic - General Somatic	**51**	Bed ward - Gynecology
37	Polyclinic - Ophtalmology	**52**	Patient Hotel
38	Polyclinic - Gynecology	**53**	Bed ward - Children
Floor 4			
12	Central storage, Central kitchen	**46**	Education
13	Waste, Cleaning, Garment handling	**47**	Research
14	Bed handling	**54**	Child Habilitation
15	IT Services, Medical Tech, MOMD	**56**	Physio, Occupational therapy, Nutrition (1)
20	Office Workspace (2)	**57**	Physio, Occupational therapy, Nutrition (2)
22	Meeting rooms, Employee areas, Support rooms		
Floor 5			
19	Office workspace (1)	**21**	Office workspace (3)

Table B.3: Functions assigned to floor 1-5 in Case Instance 3

(Proximity 2, locked to floors/specifications)

Floor 1			
1	Emergency Department (Akutt avdeling)	**13**	Waste, Cleaning, Garment handling
2	Observation	**14**	Bed handling
3	Emergency Department (Legevakt)	**15**	IT Services, Medical Tech, MOMD
4	Imaging (1) - Treatment	**16**	Ambulance (1) floor 1
5	Imaging (2) - Support	**26**	Clinical Office - Bed wards
6	Patient information/Reception	**27**	Clinical Office - Intensive Care
7	Centre of Learning and Mastery	**41**	Polyclinic - Injury
9	Hospital Ministry	**42**	Polyclinic - Support
12	Central storage, Central kitchen	**55**	Obstetrics (Delivery)
Floor 2			
8	Cafeteria	**35**	Polyclinic - Expedition
10	Personnel service/accommodation	**36**	Polyclinic - General somatic
17	Ambulance (2) floor 2	**37**	Polyclinic - Ophthalmology
18	Day Surgery	**38**	Polyclinic - Gynecology
28	Operation	**40**	Polyclinic - Mental Health
29	Pre Operative		/Substance Abuse Treatment
30	Post Operative	**44**	Oncological day unit (1)
31	Sterile Central	**45**	Oncological day unit (2) - Support
32	Intensive Care (1)	**54**	Child Habilitation
33	Intensive Care (2) - Support		
Floor 3			
11	Laboratories	**46**	Education
23	Clinical Office - Polyclinic	**47**	Research
25	Clinical Office - Imaging	**50**	Bed ward - Orthopedics/Surgery
34	Polyclinic - Endoscopy	**51**	Bed ward - Gynecology
39	Polyclinic - Lung/Cardio	**52**	Patient hotel
43	Polyclinic - Otorhinolaryngology	**53**	Bed ward - Children
Floor 4			
19	Office workspace (1)	**24**	Clinical Office - ER
20	Office workspace (2)	**49**	Bed ward - Medical (2)
21	Office workspace (3)	**56**	Physio, Occupational therapy, Nutrition (1)
22	Meeting rooms, Employee areas, Support rooms	**57**	Physio, Occupational therapy, Nutrition (2)
Floor 5			
48	Bed ward - Medical (1)		

Table B.4: Functions assigned to floor 1-5 in Case Instance 4

(Proximity 2, not locked to floors/specifications)

Floor 1			
1	Emergency Department (Akutt avdeling)	**36**	Polyclinic - General somatic
5	Imaging (2) - Support	**42**	Polyclinic - Support
6	Patient information/Reception	**47**	Research
7	Centre of Learning and Mastery	**50**	Bed ward - Orthopedics/Surgery
9	Hospital Ministry	**51**	Bed ward - Gynecology
10	Patient information/Reception	**53**	Bed ward - Children
16	Centre of Learning and Mastery	**54**	Child Habilitation
19	Hospital Ministry		
Floor 2			
2	Observation	**39**	Polyclinic - Lung/Cardio
17	Ambulance (2) floor 2	**40**	Polyclinic - Mental Health /Substance Abuse Treatment
18	Day Surgery	**41**	Polyclinic - Injury
21	Office workspace (3)	**45**	Oncological day unit (2) - Support
32	Intensive Care (1)	**46**	Education
35	Polyclinic - Expedition	**49**	Bed ward - Medical (2)
37	Polyclinic - Ophthalmology	**52**	Patient hotel
38	Polyclinic - Gynecology	**55**	Obstetrics (Delivery)
Floor 3			
3	Emergency Department (Legevakt)	**31**	Sterile Central
11	Laboratories	**33**	Intensive Care (2) - Support
12	Central storage, Central kitchen	**34**	Polyclinic - Endoscopy
14	Bed handling	**44**	Oncological day unit (1)
22	Meeting rooms, Employee areas, Support rooms	**48**	Bed ward - Medical (1)
24	Clinical Office - ER	**57**	Physiotherapy, Occupational therapy, Nutrition (2)
25	Clinical Office - Imaging		
Floor 4			
4	Imaging (1) - Treatment	**28**	Operation
13	Waste, Cleaning, Garment handling	**30**	Post Operative
20	Office workspace (2)	**43**	Polyclinic - Otorhinolaryngology
23	Clinical Office - Polyclinic	**56**	Physio, Occupational therapy, Nutrition (1)
Floor 5			
8	Cafeteria	**27**	Clinical Office - Intensive Care
15	IT Services, Medical Tech, MOMD	**29**	Pre Operative
26	Clinical Office - Bed wards		

Table B.5: Functions assigned to floor 1-5 in Case Instance 5

(Proximity 3, locked to floors/specifications)

Floor 1			
1	Emergency Department (Akutt avdeling)	**15**	IT Services, Medical Tech, MOMD
2	Observation	**16**	Ambulance (1) floor 1
3	Emergency Department (Legevakt)	**23**	Clinical Office - Polyclinic
6	Patient information/Reception	**26**	Clinical Office - Bed wards
7	Centre of Learning and Mastery	**37**	Polyclinic - Ophthalmology
8	Cafeteria	**41**	Polyclinic - Injury
9	Hospital Ministry	**42**	Polyclinic - Support
12	Central storage, Central kitchen	**43**	Polyclinic - Otorhinolaryngology
13	Waste, Cleaning, Garment handling	**55**	Obstetrics (Delivery)
14	Bed handling		
Floor 2			
4	Imaging (1) - Treatment	**29**	Pre Operative
5	Imaging (2) - Support	**30**	Post Operative
11	Laboratories	**31**	Sterile Central
17	Ambulance (2) floor 2	**32**	Intensive Care (1)
18	Day Surgery	**33**	Intensive Care (2) - Support
24	Clinical Office - ER	**35**	Polyclinic - Expedition
27	Clinical Office - Intensive Care	**39**	Polyclinic - Lung/Cardio
28	Operation	**40**	Polyclinic - Mental Health /Substance Abuse Treatment
Floor 3			
10	Personnel service/accommodation	**45**	Oncological day unit (2) - Support
25	Clinical Office - Imaging	**46**	Education
34	Polyclinic - Endoscopy	**47**	Research
36	Polyclinic - General somatic	**48**	Bed ward - Medical (1)
38	Polyclinic - Gynecology	**51**	Bed ward - Gynecology
44	Oncological day unit (1)	**54**	Child Habilitation
Floor 4			
19	Office workspace (1)	**52**	Patient hotel
20	Office workspace (2)	**53**	Bed ward - Children
21	Office workspace (3)	**56**	Physio, Occupational therapy, Nutrition (1)
22	Meeting rooms, Employee areas, Support rooms	**57**	Physio, Occupational therapy, Nutrition (2)
50	Bed ward - Orthopedics/Surgery		
Floor 5			
49	Bed ward - Medical (2)		

Table B.6: Functions assigned to floor 1-5 in Case Instance 6

(Proximity 3, not locked to floors/specifications)

Floor 1			
1	Emergency Department (Akutt avdeling)	**22**	Meeting rooms, Employee areas, Support rooms
3	Emergency Department (Legevakt)	**24**	Clinical Office - ER
6	Patient information/Reception	**26**	Clinical Office - Bed wards
7	Centre of Learning and Mastery	**27**	Clinical Office - Intensive Care
8	Cafeteria Hospital Ministry	**28**	Operation
9	Hospital Ministry	**29**	Pre Operative
11	Laboratories	**34**	Polyclinic - Endoscopy
16	Ambulance (1) floor 1	**37**	Polyclinic - Ophthalmology
20	Office workspace (2)	**38**	Polyclinic - Gynecology
21	Office workspace (3)	**42**	Polyclinic - Support
Floor 2			
4	Imaging (1) - Treatment	**39**	Polyclinic - Lung/Cardio
14	Bed handling	**40**	Polyclinic - Mental Health /Substance Abuse Treatment
15	IT Services, Medical Tech, MOMD	**41**	Polyclinic - Injury
17	Ambulance (2) floor 2	**44**	Oncological day unit (1)
18	Day Surgery	**45**	Oncological day unit (2) - Support
19	Office workspace (1)	**46**	Education
25	Clinical Office - Imaging	**48**	Bed ward - Medical (1)
31	Sterile Central		
32	Intensive Care (1)		
Floor 3			
2	Observation	**51**	Bed ward - Gynecology
33	Intensive Care (2) - Support	**52**	Patient hotel
35	Polyclinic - Expedition	**53**	Bed ward - Children
36	Polyclinic - General somatic	**54**	Child Habilitation
47	Research	**55**	Obstetrics (Delivery)
50	Bed ward - Orthopedics/Surgery		
Floor 4			
5	Imaging (2) - Support	**23**	Clinical Office - Polyclinic
10	Personnel service / accommodation	**49**	Bed ward - Medical (2)
12	Central storage, Central kitchen	**57**	Physio, Occupational therapy, Nutrition (2)
Floor 5			
13	Waste, Cleaning, Garment handling	**43**	Polyclinic - Otorhinolaryngology
30	Post Operative	**56**	Physio, Occupational therapy, Nutrition (1)

Appendix C

Center Placements of Case Instances

In this appendix, the center placements for all of the case instances in Section 8 is listed, to get an indication of the distribution of functions throughout each floor.

Floor 1	Floor 2	Floor 3	Floor 4	Floor 5
(1) 11	(11) 46	(23) 69	(19) 81	(22) 88
(2) 23	(17) 51	(27) 56	(20) 75	(49) 87
(3) 24	(18) 28	(34) 70	(21) 83	
(4) 17	(24) 50	(39) 66	(26) 77	
(5) 15	(25) 32	(40) 58	(48) 71	
(6) 26	(28) 29	(42) 67	(50) 79	
(7) 26	(29) 31	(46) 62		
(8) 2	(30) 42	(47) 54		
(9) 26	(32) 40	(51) 53		
(10) 13	(33) 43	(52) 65		
(12) 6	(35) 49	(53) 63		
(13) 12	(36) 34	(54) 54		
(14) 7	(37) 47	(56) 57		
(15) 1	(38) 35	(57) 52		
(16) 25	(41) 37			
(31) 5	(43) 47			
(44) 20	(55) 38			
(45) 19				

Table C.1: Center Placements of Case Instance 1

Indicated by *(Function) Location*

Floor 1	Floor 2	Floor 3	Floor 4	Floor 5
(1) 8	(3) 44	(8) 57	(12) 75	(19) 85
(2) 23	(17) 51	(11) 65	(13) 73	(21) 87
(4) 11	(23) 36	(26) 54	(14) 78	
(5) 21	(30) 43	(34) 58	(15) 84	
(6) 26	(31) 31	(36) 67	(20) 74	
(7) 26	(32) 40	(37) 66	(22) 77	
(9) 26	(35) 49	(38) 58		
(10) 4	(41) 37	(39) 66		
(16) 25	(42) 46	(40) 69		
(18) 17	(43) 34	(50) 53		
(24) 12	(48) 27	(51) 56		
(25) 24	(49) 47	(52) 64		
(27) 18	(55) 32	(53) 63		
(28) 5				
(29) 14				
(33) 2				
(44) 19				
(45) 13				

Table C.3: Center Placements of Case Instance 2

Indicated by *(Function) Location*

Floor 1	Floor 2	Floor 3	Floor 4	Floor 5
(1) 8	(8) 50	(11) 54	(19) 81	(48) 87
(2) 5	(10) 35	(23) 70	(20) 83	
(3) 17	(17) 51	(25) 56	(21) 77	
(4) 19	(18) 43	(34) 63	(22) 75	
(5) 14	(29) 45	(39) 64	(24) 84	
(6) 26	(30) 30	(43) 61	(49) 71	
(7) 26	(31) 41	(46) 59	(56) 76	
(9) 26	(32) 44	(47) 69	(57) 78	
(12) 9	(33) 40	(50) 53		
(13) 24	(35) 36	(51) 58		
(14) 2	(36) 49	(52) 66		
(15) 18	(37) 47	(53) 67		
(16) 25	(38) 41			
(26) 3	(40) 31			
(27) 15	(44) 28			
(41) 11	(45) 34			
(42) 6	(54) 37			
(55) 23	(28) 27			

Table C.5: Center Placements of Case Instance 3

Indicated by *(Function) Location*

Floor 1	Floor 2	Floor 3	Floor 4	Floor 5
(1) 10	(2) 37	(3) 63	(4) 72	(8) 85
(5) 15	(17) 51	(11) 58	(13) 75	(15) 88
(6) 26	(18) 49	(12) 53	(20) 84	(26) 86
(7) 26	(21) 43	(14) 67	(23) 71	(27) 86
(9) 26	(32) 44	(22) 60	(28) 83	(29) 88
(10) 23	(35) 38	(24) 64	(30) 78	
(16) 25	(37) 35	(25) 56	(43) 73	
(19) 9	(38) 29	(31) 69	(56) 77	
(36) 5	(39) 40	(33) 57		
(42) 6	(40) 31	(34) 54		
(47) 11	(41) 32	(44) 61		
(50) 14	(45) 34	(48) 65		
(51) 20	(46) 41	(57) 70		
(53) 24	(49) 27			
(54) 17	(52) 46			
	(55) 50			

Table C.7: Center Placements of Case Instance 4

Indicated by *(Function) Location*

Floor 1	Floor 2	Floor 3	Floor 4	Floor 5
(1) 11	(4) 45	(10) 56	(19) 74	(49) 85
(2) 3	(5) 40	(25) 69	(20) 78	
(3) 23	(11) 49	(34) 67	(21) 83	
(6) 26	(17) 51	(36) 70	(22) 81	
(7) 26	(18) 34	(38) 67	(50) 72	
(8) 5	(24) 30	(44) 58	(52) 80	
(9) 26	(27) 32	(45) 60	(53) 77	
(12) 1	(28) 27	(46) 52	(56) 84	
(13) 13	(29) 43	(47) 62	(57) 73	
(14) 15	(30) 44	(48) 61		
(15) 7	(31) 37	(51) 66		
(16) 25	(32) 47	(54) 63		
(23) 14	(33) 41			
(26) 17	(35) 36			
(37) 19	(39) 31			
(41) 19	(40) 35			
(42) 21				
(43) 18				
(55) 22				

Table C.9: Center Placements of Case Instance 5

Indicated by *(Function) Location*

Floor 1	Floor 2	Floor 3	Floor 4	Floor 5
(1) 8	(4) 45	(2) 54	(5) 77	(13) 87
(3) 11	(14) 41	(33) 61	(10) 83	(30) 88
(6) 26	(15) 38	(35) 64	(12) 71	(43) 86
(7) 26	(17) 51	(36) 67	(23) 78	(56) 85
(8) 2	(18) 31	(47) 69	(49) 80	
(9) 26	(19) 40	(50) 53	(57) 73	
(11) 21	(25) 37	(51) 58		
(16) 25	(31) 49	(52) 66		
(20) 18	(32) 35	(53) 63		
(21) 23	(39) 41	(54) 56		
(22) 9	(40) 48	(55) 57		
(24) 4	(41) 43			
(26) 17	(44) 46			
(27) 15	(45) 34			
(28) 13	(46) 44			
(29) 5	(48) 27			
(34) 12				
(37) 3				
(38) 9				
(42) 6				

Table C.11: Center Placements of Case Instance 6

Indicated by *(Function) Location*

Appendix D

Resulting layouts of Hospital Case

In Figure D.1-D.5, larger images of the resulting layouts from the hospital case instance presented in Section 8.4.2 are shown.

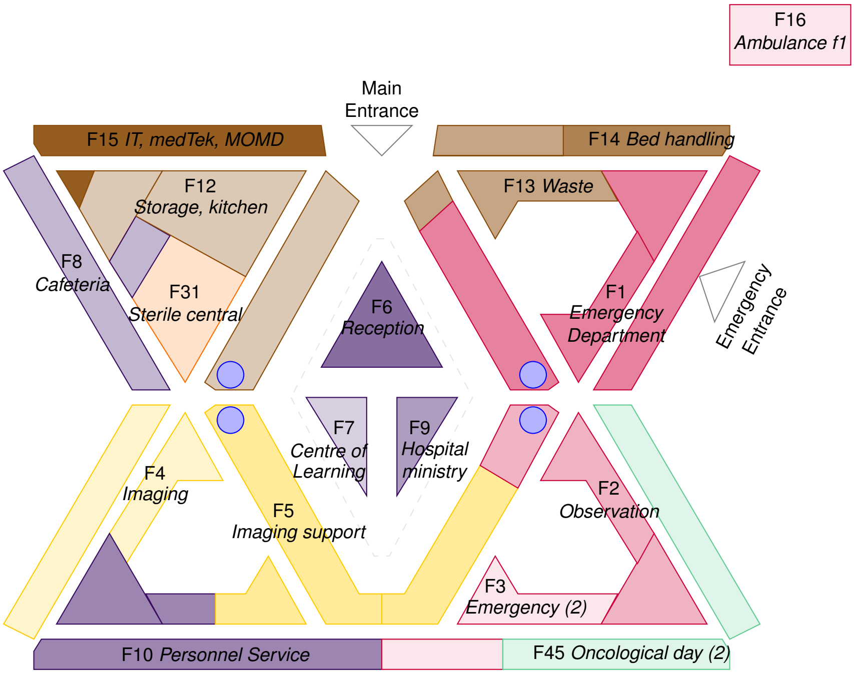


Figure D.1: Final Layout Floor 1

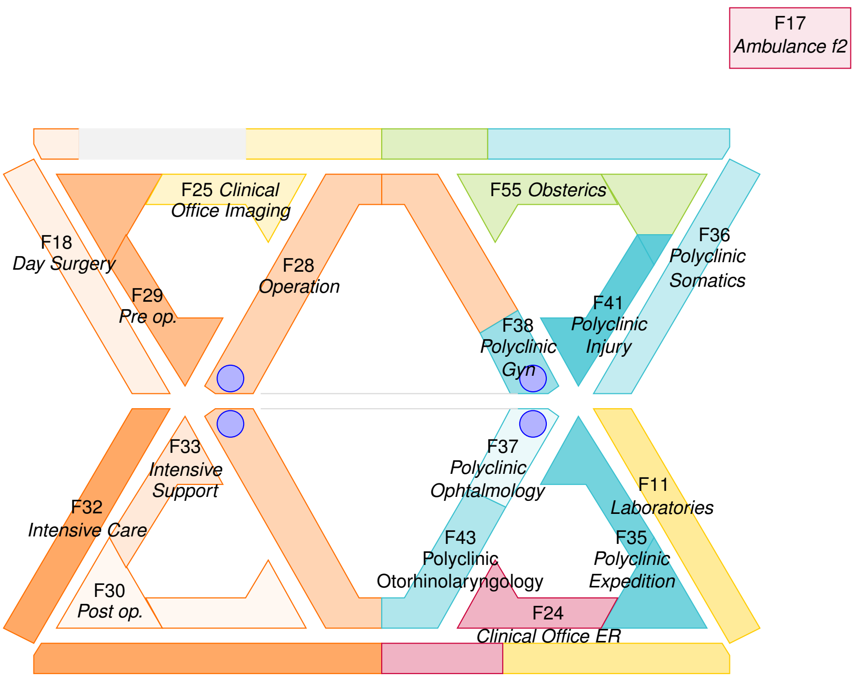


Figure D.2: Final Layout Floor 2

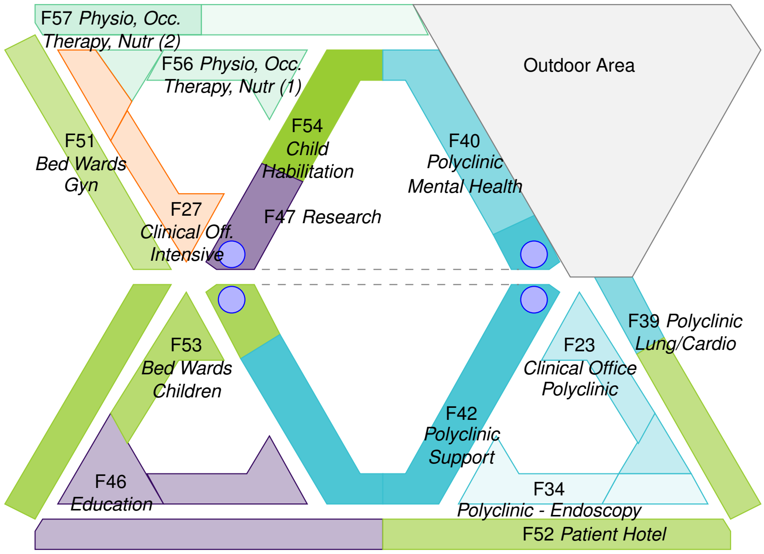


Figure D.3: Final Layout Floor 3

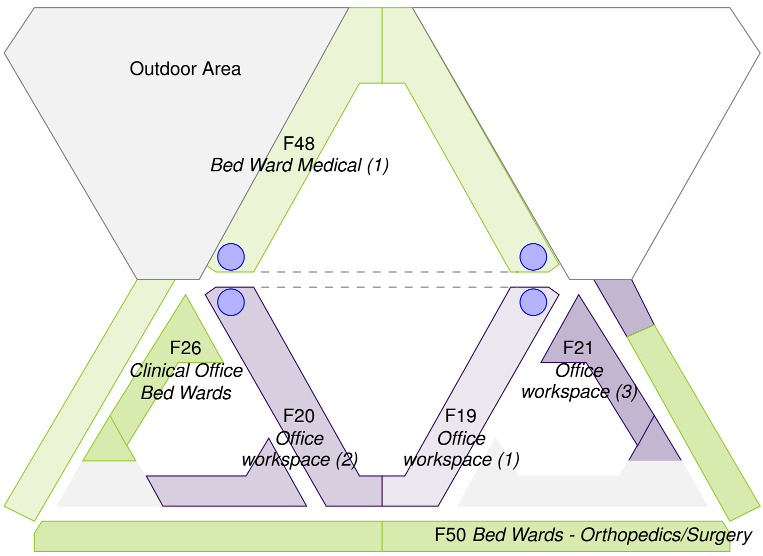


Figure D.4: Final Layout Floor 4

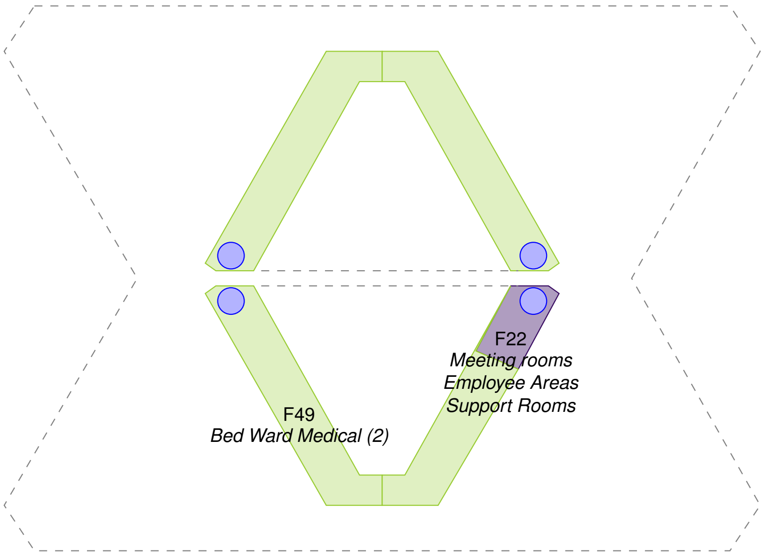


Figure D.5: Final Layout Floor 5