Patient-Centric Facility Design for Physical Rehabilitation Hospitals

by

Alejandro Teran-Somohano

A dissertation submitted to the Graduate Faculty of
Auburn University
in partial fulfillment of the
requirements for the Degree of
Doctor of Philosophy

Auburn, Alabama
December 14th, 2019

Keywords: facility layout problem, optimization, hospital design, healthcare

Approved by

Alice E. Smith, Chair, Joe W. Forehand/Accenture Distinguished Professor of Industrial and Systems Engineering, Auburn University
Jeffrey S. Smith, Joe W. Forehand Jr. Professor of Industrial and Systems Engineering, Auburn University
Daniel Silva, Assistant Professor of Industrial and Systems Engineering, Auburn University
Kevin R. Gue, Professor of Industrial Engineering, University of Louisville

Abstract

This dissertation presents two novel models for solving the facility layout problem in a physical rehabilitation hospital. The first model solves the block layout problem, where the relative location and size of the departments in the facility are determined. The model is based on Space Syntax. Space Syntax offers a series of tools that can be used to analyze and quantify spatial relations which are useful when modeling block layouts. Two Space Syntax-based metrics are introduced to model proximity and ease of access in layout designs. A tabu search algorithm is used to find the block layouts. A set of test problems show the ability of this approach to handle healthcare-specific design requirements better than existing metrics. Results show that the Space Syntax approach provides powerful, but easy to use, modeling capabilities, and that the block layouts resulting are more realistic. The second model is a mixed integer program for constructing corridor networks on a block layout that minimize travel distance, number of intersections and maximum traffic on a turn. Both models are configurable so that facility designers can generate different designs based on their goals by changing the model parameters. An evaluation of designs resulting from different parameter combinations is performed and some guidelines for facility designers are provided.

Acknowledgments

I would like to thank my advisor, Dr. Alice Smith for her guidance and support throughout my doctoral studies at Auburn. I would also like to thank my committee members for their challenging questions, especially during and after the proposal defense. Special thanks to Dr. Silva for being willing to join the committee last minute.

I want to thank my wife, Joanna, for her unwavering support during these last few years. I also want to thank my two children, Miguel and Marysia, for being one of the reasons my studies took so long, but also for keeping me smiling and motivated this whole time.

Special thanks to my parents for everything.

Deo gratias.

Table of Contents

List of Tables

List of Illustrations

List of Abbreviations

ACC	Total accessibility depth
ADL	Adaptation to Daily Living
CTN	Corridor Terminating Node
FBS	Flexible Bay Structure
FLP	Facility Layout Problem
GA	Genetic algorithm
IE	Industrial Engineering
I/O	Input/Output
MFI	Maximum Flow on an Intersection
MIP	Mixed Integer Program
NFT	Near Feasibility Threshold
QAP	Quadratic Assignment Problem
TCTN	Total Number of Corridor Terminating Nodes
TD	Total Travel Distance
TF	Total Flow

TI	Total Number of Intersections
TS	Tabu Search
TT	Total Traffic Weighted by Turns
UAFLP	Unequal Area Facility Layout Problem
XML	Extensible Markup Language

I. Introduction

Facility layout research originated in manufacturing. Attempts to apply facility design techniques to healthcare facilities have been significantly limited by the absence of domain-specific objectives and constraints. The available facility layout models were not developed with the considerations suitable for healthcare facilities.

Healthcare facility design is important because there is significant evidence that the physical environment has an impact on the quality of patient care, and that well-designed facilities can help improve the service provided (Ulrich et al., 2008). Some of the concerns related to the physical layout of the facility include: patient waiting times, staff and patient travel distances, time spent by staff at bedside, patient visibility, ease of movement through facility, traffic along corridors, patient privacy, infection control, noise management, access to natural light, access restrictions, and availability of sufficient space for medical equipment. Current approaches found in the facility layout literature are incapable of addressing these concerns.

This dissertation is focused on the design of physical rehabilitation hospitals. Rehabilitation hospitals are specialized clinics focused on helping functionally limited patients recover strength and functions so that they may go back to leading a normal, independent life. Rehabilitation exists for patients that have been discharged from a general hospital but are not yet ready to return home. The purpose of their therapy is to prepare them for this return.

A patient might need rehabilitation for a variety of reasons so that each patient requires different treatment. Even patients with a similar condition might need different treatment, depending on the severity of their condition, age, sex, etc. Therefore, rehabilitation patient treatment is highly personalized and is imparted by a team of nurses, doctors, and therapists. In general, physical rehabilitation consists of three types of therapy: physical therapy for increased strength and mobility; occupation therapy for improved everyday living skills; and speech therapy for recovery of communication skills. Additional forms of therapy that are not related to physical therapy may be required by specific patients (hemodialysis, for instance). Some rehabilitation hospitals are equipped to provide these services, but they are often outsourced to an external healthcare provider.

Figure 1 presents a sample layout of a rehabilitation hospital. Some design patterns that are found across floorplans include patient rooms aligned along the edge of the facility and next to each other, nurse stations located in the center of the patient room area, clustering of administrative offices, among others.

Figure 1: Sample floor-plan for a 34-bed rehabilitation hospital

The main department types are: administrative and maintenance rooms, patient rooms (normal and bariatric rooms, which accommodate patients with severe obesity), cafeteria and kitchen, activity rooms, speech therapy rooms, physical therapy gyms (indoor and outdoor), occupational therapy rooms (called ADL – Adaptation to Daily Living), pharmacy and pharmacy stations, nurse stations, exam rooms, case manager offices, and storage rooms. Additional departments may include a vestibule, visitor lobby, meeting rooms, etc.

Besides the daily movement of patients and staff throughout the facility, the hospital also needs to be able to handle material movement. This material includes supplies such as linens and other items for patient rooms, medicines for the pharmacy (also delivered to the patients), food for the cafeteria and kitchen, etc. The aisle network, therefore, must accommodate the needs of both patients and staff, while allowing the free flow of material and people.

In terms of the shape of the facility, it varies from hospital to hospital. Rehabilitation providers purpose build many of their hospitals, but it is not uncommon to acquire and restore others. One story buildings are preferred because two stories make it complicated to move patients around, while waiting for elevators increases traffic and non-value added waiting times.

Rehabilitation hospitals offer both inpatient and outpatient rehabilitation programs. Inpatient programs require the patient to remain in the hospital. The patient will usually remain in the hospital from two to four weeks. Outpatient programs allow the patient to come to the hospital for therapy and then head back home once they are done with the day's sessions.

Since rehabilitation hospitals are specialized hospitals, they deal with a pre-defined number of patient conditions. The specific conditions treated by rehabilitation hospitals can be arranged into the categories listed in Table 1.

Table 1: Patient Condition Classification (Centers for Medicare and Medicaid Services, 2012)

Medicaid Code	Condition
01 Stroke	Stroke
02 TBI	Traumatic Brain Injury
03 NTBI	Non-traumatic Brain Injury
04 TSCI	Traumatic Spinal Cord Injury
05 NTSCI	Non-traumatic Spinal Cord injury
06 Neuro	Neurological
07 FracLE	Fracture of Lower Extremity
08 ReplLe	Replacement of Lower Extremity
09 Ortho	Other Orthopedic
10 AMPLE	Amputation of Lower Extremity
11 AMP-NLE	Amputation, Other
12 OsteoA	Osteoarthritis
13 RheumA	Rheumatoid, Other
14 Cardiac	Cardiac
15 Pulmonary	Pulmonary
16 Pain	Pain Syndrome
17 MMT-NBSCI	Multi-Trauma no Brain or Spinal Cord Injury
18 MMT-BSCI	Multi-Trauma with Brain or Spinal Cord Injury
19 GB	Guillain-Barré Syndrome
20 Misc	Miscellaneous
21 Burns	Burns

The number of patients per each category varies widely. The distribution of patients per category (the clinical mix) for 2012 at three different levels, local (Dothan, AL hospital), regional and national are presented in Table 2.

Table 2: Patient Clinical Mix 2012 (UB Foundation Activities, Inc., 2013)

Category	Local %	Region %	National %
01 Stroke	16.0	20.7	22.1
02 TBI	1.6	3.5	4.0
03 NTBI	1.3	5.0	5.8
04 TSCI	0.4	1.5	1.4
05 NTSCI	2.0	3.7	4.4
06 Neuro	45.4	14.0	11.0
07 FracLE	8.5	12.3	10.5
08 ReplLe	6.8	8.3	9.9
09 Ortho	6.1	6.8	6.6
10 AMPLE	3.6	3.3	2.8
11 AMP-NLE	0.0	0.1	0.1
12 OsteoA	0.3	0.3	0.3

13 RheumA	0.4	0.5	0.5
14 Cardiac	1.6	4.1	3.9
15 Pulmonary	0.5	1.5	1.3
16 Pain	0.4	0.7	0.8
17 MMT-NBSCI	2.3	3.0	2.7
18 MMT-BSCI	0.5	1.3	1.4
19 GB	0.2	0.3	0.4
20 Misc	2.4	9.1	9.9
21 Burns	0.0	0.1	0.1

Additional patient population data such as age group distribution and gender mix is presented in Table 3 and Table 4.

Table 3: Patient Distribution by Age (UB Foundation Activities, Inc., 2013)

Age group	Local %	Region %	National %
0-44	3.6	6.8	7.4
45-64	27.7	25.4	26.1
65-74	26.9	24.9	24.5
75-140	41.8	42.9	42.0

Table 4: Patient Distribution by Gender (UB Foundation Activities, Inc., 2013)

Gender	Local %	Region %	National %
Male	41.5	44.9	45.9
Female	58.5	55.1	54.1

The patient mix reveals some important design requirements. The type of conditions treated, as well as the age distribution, make it necessary to move some patients around in wheelchairs. Furthermore, some patients have special infection control needs, either because they are infectious and can transmit diseases to other patients, or because they are more susceptible to an infection. Patients with severe burns are especially susceptible to infection.

Each individual patient has a specific treatment plan that is delineated by the patient's treatment team. Based on that treatment plan, a daily schedule is produced. Though each patient has an individualized schedule, they all have a similar structure:

a) Morning: nurse visit for any needed treatments, nurse or therapy staff aid patient in using the restroom, getting clean and dressed, breakfast, and two to three therapy sessions before lunch.

b) Mid-Day: lunch.

c) Afternoon: therapy sessions with rest provided in between sessions, nurses check on patient throughout the day.

d) Evening: dinner and time for family and friend visits, nurses help the patient groom, put on bedclothes, and get ready for bed.

e) Late Evening: nurse checks in on patient to provide any required treatment.

Throughout the day the patient receives visits from the assigned physician and case manager. During the time with no scheduled treatment, patients rest and relax either in their room or in the activity areas available.

A sample patient schedule is shown in Table 5.

Table 5: Sample Patient Schedule

Room: 205-A Case Manager: LF Diagnosis: NEU Date of Birth: 12/28/1935 Physical Therapist: JM Occupational Therapist: MG Speech Therapist: PC Technician: BH			
7:00 AM	Wake up	10:30 AM	Speech Therapy
7:15-7:30 AM	Personal grooming	10:45 AM	Speech Therapy
7:45 AM	Breakfast	11:00 AM-12:00 PM	Rest break
8:00 AM	Breakfast	12:00 PM	Lunch
8:15-8:30 AM	Personal grooming	12:15-1:00 PM	Personal grooming
8:30 AM	Physical Therapy	1:00 PM	Occupational Therapy
8:45 AM	Physical Therapy	1:15 PM	Occupational Therapy
9:00 AM	Physical Therapy	1:30 PM	Physical Therapy
9:15 AM	Physical Therapy	1:45 PM	Physical Therapy

9:30 AM	Occupational Therapy	2:00-2:30 PM	Rest break
9:45 AM	Occupational Therapy	2:30 PM	Speech Therapy
10:00 AM	Occupational Therapy	2:45 PM	Speech Therapy
10:15 AM	Occupational Therapy	2:45-7:00 PM	Free time

This research develops a modeling framework based on Space Syntax (Hillier, 1998; Hillier & Hanson, 1993) for healthcare facility layout and includes two models and solution methods for the physical rehabilitation hospital layout problem. The first model is used for designing the hospital's block layout. It makes use of Space Syntax techniques and metrics to model the spatial characteristics that help meet the healthcare-specific requirements demanded of a good hospital design. The model considers total travel distance, total number of turns required, as well as the overall facility's accessibility as its objectives, subject to constraints on the size and shape of the departments, on proximity, accessibility, and contact with the outside. The second model generates corridor networks on a given block layout. This model considers the trade-off between intersections on the network (which permit more routes between departments, but also make navigation more complex), the total travel distance, and the total traffic at turns. The first model is the solved using a tabu search metaheuristic while the second is modeled and solved as a mixed-integer program.

1.1 Research Objectives

The research objectives of this study are:

- Develop a modeling framework for healthcare facility layout design which takes into consideration sector-specific requirements,
- Develop models for generating block and detailed hospital layouts,
- Design and implement a solution procedure for designing physical rehabilitation hospital layouts.

1.2 Major Contributions

The research in this dissertation provides important contributions to both the facility layout literature in general and to healthcare specifically. The main contribution to the literature in general is the development of a modeling framework that can quantify the spatial characteristics of a facility, thus enabling the explicit linking of these to desired operational outcomes. By using the theory and techniques provided by Space Syntax, facility models can be developed that increase understanding of how spatial organization and operations relate and can help support the design of these facilities. This contribution is important because it can make the problem of designing facilities an interesting problem in and of itself, as opposed to it being a benchmark problem that is used to test optimization algorithms with little, if any, practical applications. An additional contribution to the general literature is the nested-bay facility representation used, which allows the representation of non-rectangular facilities.

The main contribution to the healthcare facility literature is that this research is the first effort (as far as we know) to construct hospital layouts considering healthcare-specific requirements. That is, the models proposed are not simply analogs of those found in the manufacturing literature. New objectives and constraints are developed to incorporate the concerns of healthcare facility designers. These include access to natural light, proximity (to address noise reduction, privacy, and infection control), and ease of access. The corridor generation model also considers the number of intersections (related to ease of navigation) as well as traffic and congestion. Traffic control is important in physical rehabilitation hospitals where patients are often transported by wheelchairs, which can lead to congestion and delays.

1.3 Organization of the Dissertation

This dissertation is organized as follows. Chapter II provides an overview of the facility layout literature from an engineering perspective, in both manufacturing and healthcare, and it then introduces Space Syntax and its use in healthcare applications. A brief review of tabu search is also included. Chapter III provides a detailed description of the specific design requirements encountered in planning a physical rehabilitation hospital. Chapter IV presents the Space Syntax-based block layout model, together with the tabu search algorithm used to solve it. The performance of the algorithm and the resulting designs are presented and analyzed. Chapter V describes the mixed integer programming model for corridor network generation. The block layouts obtained in Chapter IV are re-evaluated in light of the corridor networks that are imposed on them. Chapter VI offers the conclusions of this research, together with future areas of research.

II. Literature Review

The facility layout problem (FLP) is a well-known and well-studied problem in industrial engineering (IE). The FLP can be defined as the problem of finding the "best" (or close to best) arrangement of departments, the aisle structure, and the entrances and exits of each department within a facility with respect to one or more objectives.

The FLP first arose in the factory. It was from the factory floor that it took its terminology—facilities, departments, input/output points—and its concerns. However, as with IE in general, it has outgrown and moved beyond manufacturing. With this growth, the concerns of those working in the area have also expanded, so that the FLP has to be adapted and stretched to fit its new areas of application. In particular, the present work is concerned with the application of the FLP to healthcare facilities.

This literature review is divided into three parts: the first section provides an overview of the FLP literature from the engineering perspective, and discusses the models and methods developed by engineers over the last few decades for manufacturing layout and for healthcare. The second section introduces a spatial theory developed by architects called Space Syntax, which will form the basis of the modeling approach proposed in this dissertation, together with its use in healthcare facilities. The third section provides a brief introduction to tabu search, the metaheuristic algorithm used in the present work to solve the FLP for hospitals. The chapter closes with a discussion on the gaps found in the literature.

2.1 The FLP from the IE Perspective

Though the design of facilities is something that has been done from ancient times, it was not until the 1950s that analytical models were developed for addressing the FLP (Heragu, 2008). Most of the time, a facility is designed following a sequential process. First, the facility is represented in an abstract form called a block layout, where the blocks indicate the overall position and size of the different departments. Then, a set of alternative layouts are generated, examined and compared, and one is selected depending on a series of criteria or objectives. A variety of optimization methods are used for this step. Lastly, aisles/corridors and input and output points are added to the block layout, resulting in a detailed layout. In recent years, integrated approaches have been proposed, where some of the detailed layout aspects are incorporated into the design of the block layout. However, the sequential approach remains the most common one.

2.1.1 Representing the Block Layout

The block layout model represents the facility as a series of (usually) rectangular shaped departments, in order to indicate the relative location and size of the actual departments. It is, by far, the most commonly used approach in the literature. The block layout model can be further divided depending on whether the departments are of equal or unequal size. In the case of departments of equal size and shape, the most common formulation is known as the quadratic assignment problem (QAP), introduced by Koopmans and Beckman (1957). When departments are not of equal sizes, the problem is referred to as the Unequal Area FLP (UAFLP). A graphic representation of both these formulations can be found in Figure 2.

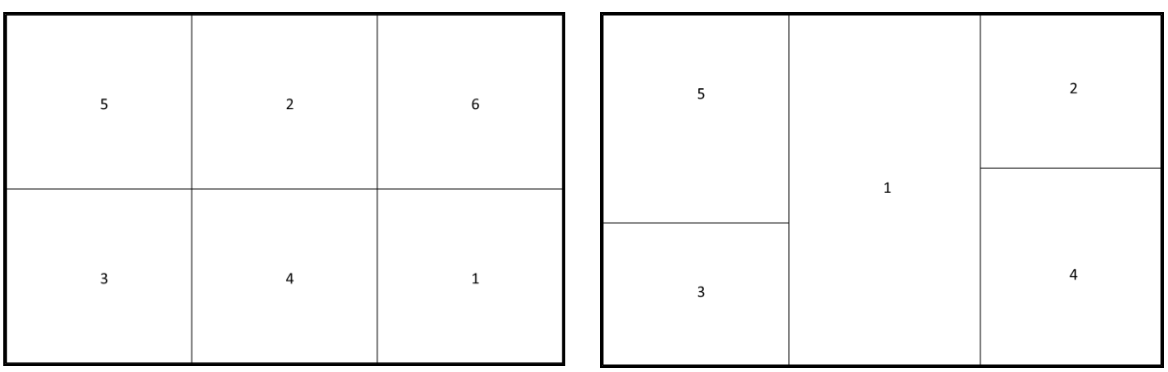


Figure 2: Block Layout Formulations (a) QAP, (b) UAFLP

To work with any of these formulations, it is necessary to represent the relative positions of the departments in some simplified manner. In all papers surveyed in which the FLP was modeled as a QAP, the department arrangement is represented as a permutation of departments (Burkard & Bönniger, 1983; Kochhar, Foster, & Heragu, 1998; Peer & Sharma, 2008; Rosenblatt & Lee, 1987; Singh & Sharma, 2008; Tate & Smith, 1995a; Urban, 1987; Wilhelm & Ward, 1987). With the UAFLP, on the other hand, a variety of representations have been used. These include department permutations (Bozer, Meller, & Erlebacher, 1994; Deb & Bhattacharyya, 2005), coordinates (Jankovits, Luo, Anjos, & Vannelli, 2011; Kim & Kim, 2000), flexible bays (Aiello, Enea, & Galante, 2002; Enea, Galante, & Panascia, 2005; Konak, Kulturel-Konak, Norman, & Smith, 2006; Kulturel-Konak & Konak, 2011; Lee & Lee, 2002; Norman, Arapoglu, & Smith, 2001; Norman & Smith, 2006; Ozdemir, Smith, & Norman, 2003; Raman, Nagalingam, & Gurd, 2009; Ulutas & Kulturel-Konak, 2012; Yapicioglu & Smith, 2012b, 2012a), sequence pairs (Kim & Goetschalckx, 2005; Liu & Meller, 2007), location matrices (Aiello, Scalia, & Enea, 2012; Kim & Kim, 1998) and slicing trees (Azadivar & Wang, 2000; Wong, 2010; Wu & Appleton, 2002). The most popular representations are the slicing trees and the flexible bay structures, as well as variations of them. We will discuss these two in greater detail.

2.1.1.1 Flexible Bays

The flexible bay representation was popularized by Tate and Smith (Tate & Smith, 1995b). The rectangular facility is divided into a flexible number of vertical bays of varying width which, in turn, are divided into rectangular departments. The layout structure is described in two parts: a permutation of departments and a set of breakpoints. (Meller & Gau, 1996b) The departments are laid out following the sequence from top to bottom and left to right (Tate & Smith, 1995b) though alternative layout sequences have been used as well (Ozdemir et al., 2003; Ulutas & Kulturel-Konak, 2012).

Consider the layout in Figure 3, where there are three bays and six departments. Using the flexible bay notation, this layout would be described by the following two sequences: 5, 3, 5, 6, 2, 1 and 2, 3, where the first list is the department sequence and the second one is the breakpoint sequence.

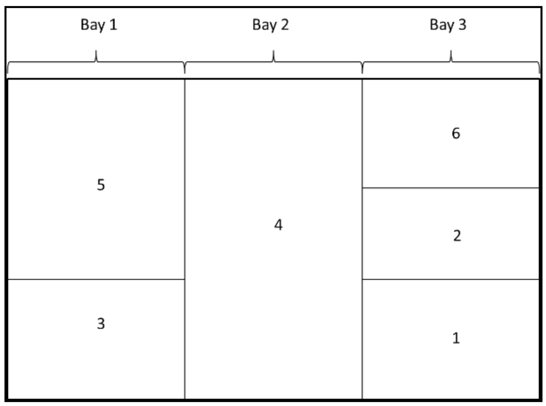


Figure 3: Flexible Bay Representation

The flexible bay representation has become one of the most popular ones in the literature of recent years. In the papers surveyed, it was the most commonly used. Its popularity is due in part to it being extremely easy to encode while consuming little resources and because it engenders a reasonable aisle structure.

A development of the flexible bay representation is the Relaxed Flexible Bays concept introduced by Kulturel-Konak & Konak (2011). The novelty of this approach is that departments within a bay can have different widths, thus allowing empty space within a bay. This leads to a larger number of potential layouts that can be evaluated and compared.

2.1.1.2 Slicing Trees

Another popular layout representation is the slicing tree. A detailed description of it can be found in Tam (1992). A slicing tree is a binary tree composed of internal nodes which indicate the direction in which a rectangular partition of the floor is cut, and external nodes representing the departments (Wu & Appleton, 2002). Internal nodes are labeled depending on the direction of the cut. Azadivar & Wang (2000) uses * and +, Wong, (2010) and Wu & Appleton (2002) use *h* and *v*, whereas Tam (1992) uses four labels: U, L, B, R. The tree can then be represented by a string called a *Reverse Polish Expression* (Azadivar & Wang, 2000), though alternative string representations exist, such as the ones presented in Wu & Appleton (2002).

A change to one of the internal nodes (the cut directions) will automatically lead to a different layout. However, Tam (1992) argues that the use of four operators, instead of two, allows for a simplified encoding because it can generate the same layouts as the two operator version without having to move the leaves of the tree. Figure 4 presents a slicing tree with its corresponding layout, using the operators from Tam (1992).

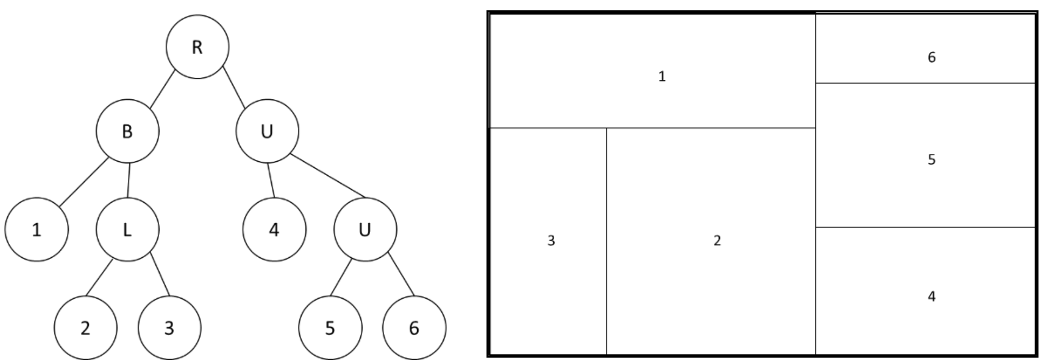


Figure 4: Slicing Tree Representation

2.1.1.3 Graph Theoretic Representations

In the graph model of a facility, the relationship among departments in terms of adjacency is the central focus. The shape and area of the departments is initially ignored. Each node in the graph represents a department and the arcs connecting them indicate adjacency between departments (Meller & Gau, 1996b). Each arc is then weighted by the amount of interaction between the departments it connects.

A very important concept that is used in graph models is that of a *planar graph*. A planar graph can be drawn in two dimensions without any arc crossing another arc. A maximal planar graph is such that it is already planar and adding an additional arc will make it non-planar (Heragu, 2008). After an initial planar graph has been developed, it is possible to modify it to generate alternative layouts. Once a desired layout has been found, the dual of the planar graph is built and from it a block layout can be constructed. A detailed description of this process can be found in Francis, McGinnis, & White (1992) and Heragu (2008). Figure 5 (a) shows the primal and dual graphs of the block layout shown in (b).

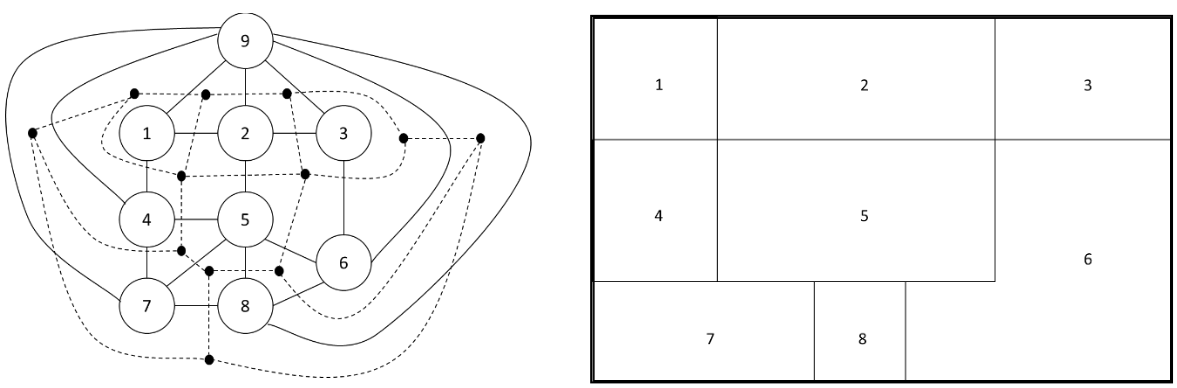


Figure 5: Graph Model (a) Primal (numbered nodes and solid arcs) and Dual (solid nodes with dotted arcs) Graphs (b) Block Layout. Node 9 represents space exterior to the facility.

The present work relies heavily on the use of graphs of the spatial relations between departments, but these are not limited to adjacency as in the graph representation. A limitation of the graph representation approach is that the area, dimensions, and shape of the departments still need to be determined after the desired configuration of the graph has been established.

2.1.2 Optimizing the Block Layout

The definition of the FLP proposed in the introduction to this chapter is good for describing the overall problem of designing a facility. However, as Francis et al. (1992) points out, the practical approach to this design problem often involves abstracting it and posing it as an optimization problem. That is, the facility's purpose is represented by a set of performance metrics and the desired spatial arrangement is found as the result of optimizing those metrics subject to a set of constraints. This section will review the different approaches to the FLP as an optimization problem.

2.1.2.1 Optimization Objectives

The "goodness" of a layout has traditionally been conceived in terms of material handling cost (Meller & Gau, 1996b) or in terms of transportation distance. However, with the advent of new applications of the FLP, additional criteria for evaluating a layout have been incorporated into the problem.

2.1.2.1.1 Material Handling Cost

The minimization of material handling cost is, by far, the most popular objective in the literature (Heragu, 2008). The objective function is given by:

$$Min \sum_{i} \sum_{j} c_{ij} f_{ij} d_{ij} , \tag{2.1}$$

where

c_{ij} = cost of moving a unit of material a unit distance between departments *i* and *j*,

f_{ij} = number of loads or trips required between departments *i* and *j*, and

d_{ij} = distance between departments *i* and *j*.

Usually, the cost and the flow are provided as parameters, whereas the distance is calculated as the problem is being solved. Several ways of measuring the distance between departments have been utilized and will be discussed in greater detail later.

2.1.2.1.2 Transportation or Travel Distance

The travel distance objective function is a simplified version of the material handling cost function. The cost of moving a unit, c_{ij}, is not considered in the function, thus reducing the objective function to be the sum of flow multiplied by the distance.

2.1.2.1.3 Closeness Ratings

Some researchers have focused instead on the idea of closeness between departments. It is assumed that the material handling cost is reduced when two departments with high levels of interaction are adjacent (Meller & Gau, 1996b). Based on an activity relationship analysis (descriptions of such a procedure are presented in Francis et al. (1992) and Heragu (2008)) a relationship chart is developed which indicates the desirability of two departments being adjacent. This desirability is encoded and assigned a numerical value that is used in maximizing the following objective function:

$$Max \sum_{i} \sum_{j} (r_{ij}) x_{ij}, \tag{2.2}$$

where

r_{ij} = closeness rating between departments *i* and *j*, and

x_{ij} = binary variable indicating whether department *i* and *j* are adjacent.

2.1.2.1.4 Other Manufacturing Specific Objectives

Other objectives encountered in the manufacturing layout design literature include: cycle time and productivity (Azadivar & Wang, 2000), WIP (Raman et al., 2009), robustness (Kulturel-

Konak, Smith, & Norman, 2004; Norman & Smith, 2006), amount of slack space (Lee & Lee, 2002), re-arrangement costs (Kulturel-Konak, Smith, & Norman, 2007; McKendall & Liu, 2012; Raman et al., 2009), department shape (Wang, Hu, & Ku, 2005; Wong & Chan, 2009; Yang & Hung, 2007), and deviation from budget (Raman et al., 2009; Wong & Chan, 2009)

2.1.2.1.5 Non-Manufacturing Objectives

The move from the industrial to the service sector has brought about a shift in the type of objective functions being considered. For layout design in retail, the objective functions include revenue (Ozgormus & Smith, 2018; Yapicioglu & Smith, 2012b, 2012a) and exposure of products (Mowrey, Parikh, & Gue, 2019); for the design of museums, theme parks and other attraction-based enterprises, the objective of uniformity of the distribution of attraction arises (Li & Smith, 2018), and in the layout design of construction sites, an objective seeking to minimize the risk posed by various hazards is considered (Abune'Meh et al., 2016).

Noticeably, the literature on healthcare facility layout has not yet provided any new objective functions.

2.1.2.1.6 Multiple Objectives

There are two ways of dealing with multiple objectives.

A first approach is to incorporate them into a single objective function. The simplest way of doing this is by using weighted objectives, where the weight is assigned according to some criterion, such as the importance of that objective in the decision-making process. Such a model was first developed and analyzed using a graphic method by Rosenblatt (1979). Fortenberry & Cox (1985) developed a model where the workflow is weighted by the closeness rating. In Urban (1987), a constant to weigh the importance of the closeness rating relative to the work flow is used.

Determining the value of the weights poses a new series of problems, including that of having different scales between objectives.

Seeking to answer what value of the weights is the most appropriate, (Meller & Gau, 1996a) studied the effect of various values of α in an objective function of the form:

$$\alpha \sum_i \sum_j c_{ij} f_{ij} d_{ij} - (1-\alpha) \sum_i \sum_j (r_{ij}) x_{ij}, \qquad (2.3)$$

where $\alpha \in [0,1]$.

A similar objective function was used in Peer & Sharma (2008), where the flow and the closeness ratings are normalized, so as to avoid the scaling conflict. On the other hand, Yapicioglu & Smith (2012b) developed an objective function that does not rely on weights.

The second way of addressing multiple objectives is by solving a multi-objective optimization problem. An advantage of this approach is that multi-objective optimization procedures generate a set of solutions for different values of each objective. The set of non-dominated solutions is called the Pareto front. The Pareto front shows the trade-off between conflicting objectives. Not much work has been done with multiple objectives, with the exception of Yapicioglu & Smith (2012a) and Kulturel-Konak et al. (2007).

2.1.2.2 Distance Metrics

Most objective functions rely on the distance between departments. Measuring this distance brings with it some difficulties, such as determining the beginning and ending points of the measurement, the realism of the measurement, and its ease of calculation.

As for the starting and ending points, distance is often measured between the centroids of the departments; however, there is a growing body of work that measures the distance from a department's input/output points (I/O). As to the specific metric used, it can depend on the material handling system used (Ozdemir et al., 2003). Some of those distance metrics are now described.

2.1.2.2.1 Euclidean Distance

The simplest and most unrealistic distance metric is the Euclidean distance. It measures the length of a straight-line between the centers of two departments. It can be calculated using the following formula, where d_{ij} is the distance, x_i and y_i the coordinates of department *i*'s centroid:

$$d_{ij} = \sqrt{\left(x_i - x_j\right)^2 + \left(y_i - y_j\right)^2}. \tag{2.4}$$

2.1.2.2.2 Rectilinear Distance

The rectilinear distance metric is the most widely used metric in the literature. It measures both the horizontal and vertical distance between department centroids and is calculated by:.

$$d_{ij} = \left|x_i - x_j\right| + \left|y_i - y_j\right|. \tag{2.5}$$

2.1.2.2.3 Contour Distance

The contour distance adds a layer of realism by only allowing travel that originates and ends at an I/O point and runs along the boundaries (or perimeter) of the departments (Arapoglu, Norman, & Smith, 2001; Kim & Goetschalckx, 2005; Norman et al., 2001). Once the candidate I/O points have been determined for every department, a material flow network is constructed where each node represents an I/O point and the arcs connecting them are the department edges (Norman, Smith, Yildirim, & Tharmmaphornphilas, 2001). In order to find the actual distance, a shortest path on a network problem is solved (Norman et al., 2001).

The distance between departments *j* and *k*, therefore, is defined as:

$$d_{j,k} = \mathrm{Min}\left\{d_{j,l,k,m} \colon l \in Loc_j, m \in Loc_k\right\}. \tag{2.6}$$

where Loc_j and Loc_k are sets of candidate I/O points for departments j and k, respectively, and $d_{j,l,k,m}$ is the distance between I/O point l in department j and I/O point m in department k (Norman et al., 2001).

2.1.2.2.4 Aisle or Corridor Distance

If the actual aisle structure and I/O points of the facility are known, then it is possible to measure the length of aisles and use that as the distance metric (Heragu, 2008). However, this is not very common.

2.1.2.3 Constraints

The most important constraints used in optimization models of facilities reflect structural limitations to the design, such as location and shape. The types of constraints most commonly found in the literature are shape constraints and area constraints. Other types of constraints, such as overlap constraints, can be addressed by the representation method used. For example, the procedure used for decoding a flexible bay representation automatically ensures that no departments will overlap.

2.1.2.3.1 Shape Constraints

These types of constraints seek to restrict the shape of the departments. In general, only rectangular departments are allowed, though some of the older grid-based discrete representations permit irregular department shapes. The purpose of these constraints is to keep the design from generating extremely thin and elongated departments, which are unrealistic. These constraints have been enforced in various ways. Montreuil's model, referenced in Meller & Gau (1996b), does this by setting bounds on the length and width of each department, in the following manner:

$$lb_i \leq 2w_i \leq ub_i, \tag{2.7}$$

where lb_i and ub_i are the lower and upper bound, respectively.

Alternatively, these shape requirements can be enforced using an aspect-ratio constraint, which ensures that the ratio between the largest and the smallest side of the rectangle does not exceed a certain value. Two different ways of calculating this aspect ratio have been used, the first one in Sadan Kulturel-Konak (2007) and Aiello et al. (2002):

$$\alpha_i \leq \alpha, \tag{2.8}$$

where $\alpha_i = \max\{l_i, w_i\} / \min\{l_i, w_i\}$ and l_i, w_i are the length and width of department i, respectively.

The second form, used in Bozer et al. (1994) and Yapicioglu & Smith (2012a), is:

$$\frac{P_i}{4\sqrt{A_i}} \leq \bar{\alpha}_i, \forall i, \tag{2.9}$$

where P_i is the perimeter of department i, A_i its area and $\bar{\alpha}_i$ the maximum aspect ratio for that department. It is based on the principle that "the perimeter of an object increases as it becomes more irregular in shape" (Bozer et al., 1994).

2.1.2.3.2 Area Constraints

Area constraints prevent departments from having abnormal sizes, such as being larger than the facility itself, or being smaller than a certain minimal area. When formulated in their most basic form, they are non-linear, which increases the difficulty of solving the problem. Some authors keep the non-linear constraints (Jankovits et al., 2011), but most others seek alternatives. In order to circumvent this issue, several approaches have been proposed, including using surrogate functions based on the department perimeter (Montreuil, 1990) and a polyhedral outer approximation method (Kim & Goetschalckx, 2005). Other methods simply avoid the problem, either by fixing the department shapes (Kim & Kim, 2000) or by using the flexible bay representation, which enforces this constraint using only linear equations (Konak et al., 2006). To avoid departments being placed outside of the facility, the width and length of the facility are often provided as a user-defined parameter.

2.1.3 The Detailed Layout

Once a good block layout has been found, the next step is to design the detailed elements of the layout, including the corridor network.

Many approaches do not actually construct a corridor network, but locate the input and output points, and use the contour of the departments as a proxy for the actual network. A variety of techniques are compared in Arapoglu et al. (2001), both exact and heuristic, for locating input/output points based on the contour distance metric. Kim & Kim (1999) use a branch and bound algorithm to determine the input/output locations for a given layout. Other approaches concurrently design the block layout and the approximated corridor structure. Deb & Bhattacharyya (2005) test different techniques for generating block layouts and locating input/output points simultaneously, while both Scholz , Jaehn, and Junker (2010) and Friedrich, Klausnitzer, and Lasch (2018) extend STaTs, a slicing tree, tabu search algorithm, to include the location of input/output points while using the contour distance metric. Norman et al. (2001) also uses a genetic algorithm to concurrently optimize department shape, location, and the number and location of input/output points. In Aiello et al. (2002), the layout and the aisle structure for a unidirectional AGV system are concurrently optimized using a hybrid genetic algorithm and an integer programming technique. A combination of linear programming, simulated annealing, and heuristic algorithms are used by Kim & Goetschalckx (2005) to concurrently optimize the block layout, the input/output point location and the network connecting all those points. By taking advantage of a slicing floor-plan encoding, Wu & Appleton (2002) simultaneously solve the block layout and the aisle structure problems using a genetic algorithm.

Alagoz, Norman, & Smith (2008) design an actual corridor network on a given block layout by using a multi-phase method. The authors assume the existence of a block layout with optimally located input/output points (at department intersections or at the facility exterior) and build the aisle structure that minimizes material handling cost and aisle cost.

A different set of approaches operate in the opposite direction, that is, given an aisle structure, the objective is to locate the departments and input/output points in the most effective way. For instance, Montreuil & Ratliff (1989) use cut trees to design a flow network and, based on it, generate a layout. Similarly, Tretheway & Foote (1994) develop an algorithm called FAST that fixes an aisle structure and places the departments around that structure. Building on this work, Benson & Foote (1997) developed DoorFAST, an algorithm that adds the optimal location of input and output points given an aisle structure. Likewise, Yapicioglu & Smith (2012a, 2012b) assume a race-track aisle structure and locate the departments in such a way that maximizes revenue for a retail facility.

More recent work on aisle networks has focused on the design of non-traditional aisles. Masel & Marinchek (2016) do this for a manufacturing facility, Mowrey et al. (2019) for retail, but by far the most common applications are in warehouse design (Gue, Ivanović, & Meller, 2012; Öztürkoğlu & Hoser, 2019).

2.1.4 The FLP in Healthcare

Research in the design of optimal layouts for hospitals dates to the 1970s. From the beginning it differed little from layout design for manufacturing, as Delon (1970) defined it: to "locate the departments of the hospital in such a way as to minimize the total cost of traffic among all departments." This was achieved using the CRAFT (Buffa, Armour, & Vollmann, 1964) and CORELAP (Lee & Moore, 1967) algorithms developed for manufacturing facilities.

The bulk of the available literature concerning optimal layouts of healthcare facilities is based on the QAP, beginning with Elshafei (1977) who sought to minimize total distance travelled. Building on that work, Butler, Karwan, & Sweigart (1992) present a modified QAP where the objective is the minimization of travel distance cost with constraints on the area of the departments, space allocation at each floor, and the number of beds available for each department or service (Butler, Karwan, Sweigart, & Reeves, 1992).

Another well-known hospital layout problem is the Krarup30 problem, a QAP with 30 departments that originated in 1972 when several architectural firms competed to design the Klinikum Regensburg, a university hospital in Germany. Several approaches for solving it are described in Hahn and Krarup (2001), including various heuristics (Burkard & Bönniger, 1983; Burkard & Stratmann, 1978), simulated annealing (Burkard & Rendl, 1984), and tabu search (Skorin-Kapov, 1990; Taillard, 1991).

Yeh (2006) proposed an instance of the QAP where 28 departments must be allocated in a four-story hospital building and used an annealed neural network to solve it. To solve the same problem, Liang and Chao (2008) present a tabu search algorithm and Cheng and Lien (2012) a hybrid swarm algorithm known as a particle bee algorithm.

Recent work by Arnolds & Gartner (2018) and Helber, Böhme, Oucherif, Lagershausen, & Kasper (2016) continues the trend of using the QAP for addressing the layout problem for entire hospitals, including hospitals composed of multiple buildings.

Other researchers have focused not on the entire hospital, but on some department within it. Some examples include emergency rooms (Ahmed & Alkhamis, 2009; Rismanchian & Lee, 2017; T.-K. Wang, Yang, Yang, & Chan, 2015; Zuo et al., 2019), operating theaters (Chraibi, Osman, & Kharraja, 2019), and radiology units (Benitez et al., 2018). These approaches do not rely on the QAP, but on the UAFLP.

2.2 The Architectural Perspective: Space Syntax

Space Syntax is defined as "a set of techniques for the representation, quantification, and interpretation of spatial configuration in buildings and settlements" (Hillier, Hanson, & Graham, 1987). It first developed as a descriptive theory of space. Two key concepts in Space Syntax are spatial configuration and spatial relation. A spatial relation exists when there is some relationship between two spatial units. A spatial configuration, on the other hand, is "a set of interdependent relations in which each is determined by its relation to all the others" (Hillier, 1998). That is, a spatial configuration is a system of spatial relations, and a configurational analysis allows quantification and comparison of different configurations. The interested reader is referred to Hillier and Hanson (1993) and Hillier (1998) for an in-depth description of the theory behind Space Syntax, as well as its most common techniques. A brief introduction to the Space Syntax methodology is provided in Chapter III.

The Space Syntax literature has focused on the description and comparison of already existing facilities. A wide range of spatial units (convex spaces, axial lines, visibility polygons), spatial relations (adjacency, permeability, visibility), and spatial metrics (depth, integration, control) have been used for studying the configuration of varying types of facilities and spatial systems (e.g., houses, hospitals, universities, urban settlements). Most of these studies assume a detailed layout, that is, one with corridor structures, input/output points, windows, etc., so that the approach has rarely been used for designing.

When it comes to the use of Space Syntax techniques in healthcare, most work has been done in investigating wayfinding (ease of navigation) in hospitals (Haq & Luo, 2012; Haq &

Zimring, 2003; Peponis, Zimring, & Choi, 1990). An important relation between low depths and ease of navigation (and with it, increased traffic) is a recurrent finding. Another common area of study is nurse behavior (Lu & Zimring, 2012; Seo, Choi, & Zimring, 2011; Trzpuc & Martin, 2010; Yi, 2010) and nurse movement (Choudhary, Bafna, Heo, Hendrich, & Chow, 2010; Hendrich et al., 2009). Like with the work on wayfinding, lower depths lead to lower travel times and increased times at patient bedside. Visibility is also an important factor in these studies.

Other applications of space syntax research to healthcare settings include studies on privacy (Alalouch & Aspinall, 2007; Alalouch, Aspinall, & Smith, 2009), hospital evacuation (Ünlü, Ülken, & Edgü, 2005), whole-life costs (Kim & Lee, 2010), face-to-face contact between clinicians (Rashid, 2009), evaluation of layouts with respect to patient satisfaction (MacAllister, Zimring, & Ryherd, 2016; Zwart & Voordt, 2015), and the effects of cultural differences on nursing unit design (Cai & Zimring, 2017). A good source of applications using the most recent developments in Space Syntax analysis in healthcare can be found in Sadek & Shepley (2016).

The emphasis on description entails that no significant effort has been put in developing optimization models that improve the metrics and behaviors linked to spatial variables. Hillier (1998) presents a heuristic procedure for generating depth-minimizing and depth-maximizing designs. Besides that, only one instance was found where semi-automated layout generation methods based on optimization approaches were used to generate layouts (Helme, Derix, & Izaki, 2014) and one instance where discrete-event simulation was used in conjunction with a Space Syntax analysis to support the design of an emergency room (Morgareidge, Cai, & Jia, 2014).

2.3 Tabu Search for the Facility Layout Problem

The FLP has been proven to be NP-complete (Meller & Gau, 1996b), therefore, optimal solutions can only be obtained for small problems in general. As a result, heuristic and metaheuristic procedures have been favored over exact procedures when solving the FLP. Among the most commonly used are simulated annealing (Deb & Bhattacharyya, 2005; Kim & Goetschalckx, 2005; Kim & Kim, 1998; Sahin & Turkbey, 2009; Singh & Sharma, 2008; Tam, 1992; Wilhelm & Ward, 1987), genetic algorithms (Aiello et al., 2012; Azadivar & Wang, 2000; Drira, Pierreval, & Hajri-Gabouj, 2013; Kochhar, 1998; Kochhar et al., 1998; Liu & Meller, 2007; Norman et al., 2001; Norman & Smith, 2006; Ozdemir et al., 2003; Tate & Smith, 1995b; Wang, Yan, Zhang, Shangguan, & Xiao, 2008), and tabu search (Kulturel-Konak, 2012; Kulturel-Konak et al., 2007; Skorin-Kapov, 1990; Taillard, 1991; Yapicioglu & Smith, 2012a, 2012b). Tabu search has been shown to be very effective in solving the FLP (Arostegui, Kadipasaoglu, & Khumawala, 2006).

Tabu search (TS) was introduced into the optimization literature by Fred Glover (Glover, 1986). It consists of a neighborhood search accompanied by a “tabu” mechanism, which disallows returning to recently visited partial or whole solutions, thus preventing the search from getting trapped in local optima. The tabu mechanism seeks to mimic the behavior of short and long-term memory, and it is this exploitation of memory that is central to tabu search (Glover & Laguna, 1998). Short-term memory strategies include a tabu-list which can include complete solutions, attributes of certain solutions, or moves that lead from one solution to another; as well as aspiration criteria which allow overriding the tabu mechanism in certain desirable situations. Long-term memory strategies help balance diversification and intensification in a tabu search. Long-term strategies include frequency-based memory and re-starting the search (Laguna, 2018).

A canonical tabu search procedure is:

```
CurrentSolution = generateInitialSolution()
BestSolution = CurrentSolution
BestFitness = evaluateSolutionFitness(CurrentSolution)
TabuList = {}
iteration = 1
WHILE NOT stopping_condition
  neighborhood = generateNeighborhood(CurrentSolution)
  evaluateNeighborhood(neighborhood)
  Sort neighborhood by fitness
  FOR i=1 to neighborhood.size
       candidate = neighborhood(i)
       IF candidate is NOT Tabu OR aspiration_criteria THEN
            CurrentSolution = candidate
            IF CurrentSolution.fitness < BestFitness THEN
                 BestSolution = candidate
                 BestFitness = candidate.fitness
            END IF
       END IF
  END FOR
  iteration++
END WHILE
```

Algorithm 1: Canonical tabu search

There are several strategies available for handling constrained problems using a tabu search: solution encoding can be such that it guarantees feasibility, infeasible solutions can be discarded from the neighborhood, or a penalty function can be used to increase (in minimization problems) the objective function value of infeasible solutions. The latter strategy is the most common. (Kulturel-Konak, Norman, Coit, & Smith, 2004). Different penalty functions have been proposed, some focused on the number of constraint violations and others based on the distance from the region of feasibility (Joines & Houck, 1994). Kulturel-Konak et al. (2004) develop a memory-based penalty function that exploits the contents of the tabu-list to adjust a near-feasibility threshold that allows the search to explore parts of the infeasible region that are within the threshold. This is an application of the tabu-search principle of strategic oscillation (Laguna, 2018).

2.4 Gaps in the Literature

There is significant evidence that elements of the physical environment have an impact on the quality of care provided (Ulrich et al., 2008), yet none of the hospital design research encountered considers this. The models found were basically instances of the FLP applied to a hospital setting. Healthcare has many sector specific requirements that are not relevant in the industrial or retail sectors. Concerns related with infection control, visibility, traffic and wayfinding, inter-staff interaction and communication, among others, need to be included in the optimization models if they are to be of practical use to healthcare facility designers. Even for purely industrial applications, the models found in the literature are simplistic and limited, hampering their use in real-world applications. In consequence, the FLP has not often been thought of as a problem of itself, but rather as a testing ground for different optimization algorithms. This is evidenced by the research on the topic from the last few years, which shows a clear bias towards the development of algorithms and optimization procedures, while neglecting the actual design components of the FLP. Since the FLP has been approached mostly as an optimization problem—and not as a design problem—the models have been developed to be optimized, not to support the design process. As a result, the models cannot answer the types of questions that interest facility designers. They are of limited use in trying to study, for example, how making changes to the corridor network affects traffic, or how varying the shape of the facility impacts the operational objectives.

The following subsections describe some of the limitations identified in the approaches offered by the current literature, which will be addressed in this dissertation.

2.4.1 Limited understanding of how spatial organization relates to operation

Though the FLP is, in its essence, a spatial problem, it has rarely been studied as such from the engineering perspective. One significant exception is provided by Johnson (1992), where the relationship between mean trip times and the shape of the building is studied. On the architectural side, research has tended towards pure description, rather than prescription, and any optimization approach used in the engineering literature has been significantly absent.

What the engineering approaches have lacked is a means of characterizing space so that spatial characteristics can be mapped to operational characteristics. This dissertation argues that Space Syntax provides an appropriate framework for this task.

2.4.2 Modeling limitations

Given the limitations discussed in the previous subsection, it should come as no surprise that the facility models themselves are not designed to explicitly capture the spatial nature of the FLP. In addition, the transition of the FLP from the industrial sector to the healthcare sector has not been successfully achieved because the unique traits of healthcare have not been taken into consideration. The limitations in the prevailing models can be classified into three categories: (1) absence of spatial facility models, (2) limitations in objective functions and constraints considered, and (3) limitations in representational power.

2.4.2.1 Absence of spatial facility models

A facility is a spatial system that houses an operational system. It is the task of the facility designer to organize the spatial system in such a way that it supports the operational one. Besides the inability to describe space quantitatively (the lack of a “spatial language”) there is also an absence of facility models. That is, models that describe how the spatial and the operational systems interact. Without these types of models, it is extremely difficult to study how one can be changed for the benefit of the other.

2.4.2.2 Limited objective functions and constraints

The limitations in objective functions and constraints arise because the FLP has been used mostly in manufacturing. Hence, the objective functions that have been considered are those that are relevant to manufacturing, such as material handling costs. These are not necessarily important or even relevant to healthcare facility design. Moreover, the objective functions used in some service sector facilities (revenue, for instance) are not applicable to healthcare. Other traditional objectives are, on the other hand, important to healthcare, such as travel distances. A similar issue occurs with constraints. The constraints used in the FLP literature are mostly limited to constraints related to area and shape, or to proximity. Of these, only the proximity constraints reflect, to some extent, the link between the spatial arrangement of the facility (proximity being a spatial characteristic) and the operation taking place within. There are usually operational reasons for keeping certain areas adjacent or apart. At the same time, the way proximity has been modeled does not fully unlock the potential of using it to enforce operational requirements. In this work, proximity is used to address matters that are unique to healthcare, such as privacy, infection and noise control, among others. To do this, however, a more powerful way of addressing proximity constraints is required. Space Syntax metrics such as depth offer the means to do so.

2.4.2.3 Limited representational power

A parallel problem to that of the absence of proper spatial facility models is that of a lack of appropriate means of representing the facility. The representational methods discussed earlier are, for the most part, incapable of representing facilities that are not rectangular, and some impose restrictions on the types of layouts they can describe (slicing vs. non-slicing). Not one of the hospitals visited for this research was rectangular. Hence, the traditional methods of representing and encoding a facility need to be either replaced or extended to be useful in practice.

III. Theoretical Background

This chapter opens with a brief presentation of the most basic Space Syntax techniques. This is followed by an overview of some of the design requirements for physical rehabilitation hospitals that are not found in manufacturing settings. Then, a description of how Space Syntax is adapted to block layout design is given.

3.1 Space Syntax Techniques

The basic approach to a Space Syntax analysis can be described by the following steps:

1) Define a set of spatial units.
2) Define a set of spatial relations among those units.
3) Construct a graph of the facility in terms of each spatial relation.
4) Define a set of spatial metrics to be computed from the spatial relation graphs.
5) Compute the values of the spatial metrics.

A spatial unit is the most basic element of analysis. For the purpose of this research, the spatial unit used is a department. A spatial relation exists when there is some spatial characteristic linking two spatial units. For instance, a relation of adjacency exists between two neighboring departments. A facility (or *complex*, to use the Space Syntax terminology) is essentially a system of spatial elements linked by spatial relations. In the Space Syntax literature, it is also referred to as a spatial configuration. This system of relations can be represented using a graph, where each node represents a spatial unit and links indicate that a spatial relation exists between the two units.

An analysis based solely on the binary spatial relations focuses only on the parts of the whole. An analysis of the entire system of relations is necessary because of two properties of the system: (1) a complex is different when looked at from different points of view; and (2) local changes affect the structural properties of the whole (Hillier, 1998).

Figure 6 exemplifies property (1). It shows a facility viewed from department A and from department B and reveals how the entire complex looks differently from each one. Property (2) implies that a local change in the relation between two units results in an entirely new configuration.

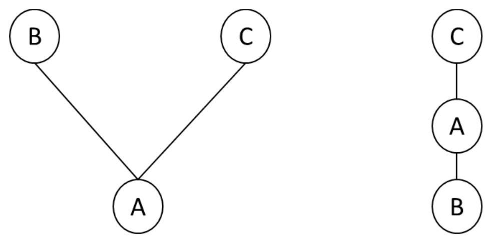


Figure 6: Complex seen from different spaces: (a) From department A (b) From department B

An important concept in Space Syntax is that of depth. The depth between two spatial units A and B is the least number of intervening nodes between both on the corresponding graph. It is the shortest number of nodes to get from one to another. Note that depth can be measured for any spatial relation, so that adjacency depth is measured on the adjacency graph, accessibility depth on the accessibility graph, and so on. Each one of these measures of depth provides different information about the relationship between spaces. Depth can also be calculated for different types of spatial elements (visibility polygons, for instance) and serve as a basis for characterizing them in spatial terms. These characteristics can then be used to find correlations with operational metrics.

One of the traits of a spatial complex mentioned previously was that it looks different from different points. This property is made explicit with justified graphs. A justified graph is constructed by setting a node as the root with all nodes that are directly connected to it (that is, nodes at depth 1 from it) aligned horizontally immediately above it, immediately above those, all nodes that are two levels deep from the root, and so on.

Figure 7 shows two justified graphs (one for node 1 and one for node 5), along with the original graph of a facility.

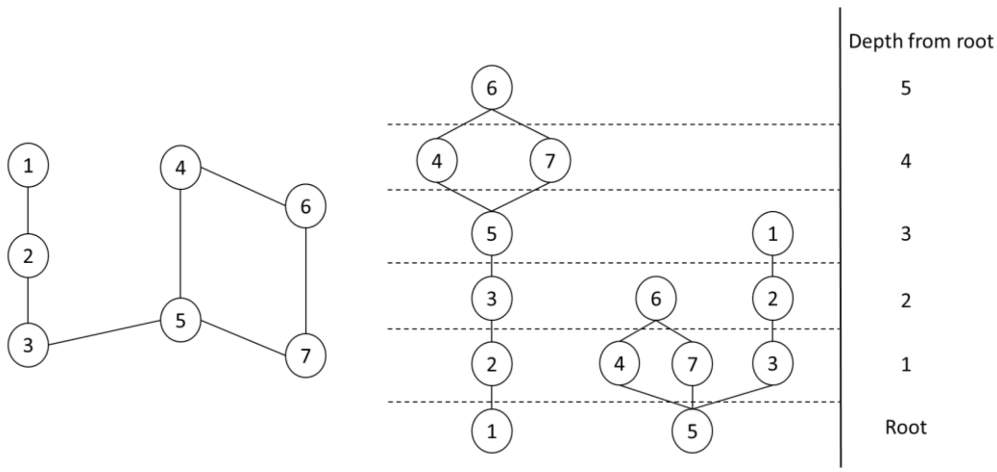
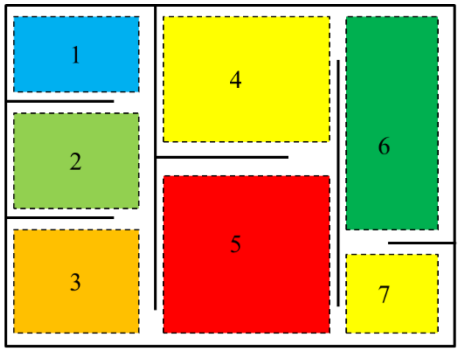


Figure 7: Justified graphs for rooms 1 and 5

Justified graphs give a more detailed and visual understanding of the depth relationships between different spaces. These depth relationships often reflect some desired functional aspect of the building. For instance, spaces with restricted access or with privacy requirements (such as an operating room or a medical records room) will tend to be found at a greater depth from other spaces. This greater depth serves as a means of controlling access and isolating the space. From a justified graph, it is easy to calculate the total depth or mean depth ($\bar{d}$) from one space to all others. This, in turn, can be used to develop a depth map, where each space is colored based on its mean depth. In complex facilities, this map can provide valuable insights into the distribution of depth throughout the facility. An example of a depth map is shown in Figure 8.



Space #	Mean depth
1	3.167
2	2.333
3	1.833
4	2.167
5	1.667
6	2.667
7	2.167

Figure 8: Sample depth map

3.2 Healthcare Facility Requirements

A series of meetings with staff and administrators of several different physical rehabilitation hospitals yielded the following list of functional requirements which can be supported by appropriate design of the facility: space availability, privacy, infection control, easy access to patients, and patient comfort. They are termed functional requirements because they are related to the goals of the function (or operation) that takes place in the facility. They can also be referred to as "operational requirements."

As with any other facility, healthcare facilities have space demands that must be met to provide the services required. These demands are defined, in part, by the type and size of equipment/furniture to be used within a given space. In the facility layout literature, this requirement is commonly met by imposing minimum area and aspect ratio constraints.

Privacy and infection control are healthcare-specific requirements. Infection control refers to the institution of mechanisms for preventing the spread of dangerous pathogens and infections across patients. Both requirements are met in practice through the separation and isolation of spaces, that is, spaces that are meant to be private are placed further away from public areas.

On the other hand, patients should be in areas with easy access for caretakers. The easier it is to access a certain area, the shorter the response times will be. Hence, patient rooms should be as accessible as possible to nurses, whose primary responsibility is to respond to patient needs in a timely manner. Ease of access is a function of the complexity of navigation required to go from one space to another. This depends on two factors: how close the two spaces are from each other, and how many turns are required to move from one to the other.

From a facility design perspective, two important ways in which the physical layout can affect patient comfort is by incorporating noise reduction and access to natural light. Noise reduction is achieved by keeping noisy areas of the facility far away from those where noise should be avoided. This is observed in actual hospitals by keeping kitchen and dining facilities apart from patient rooms. Natural lighting is known to have a positive impact on patient recovery. Windows are the simplest means of obtaining natural light in a facility. By maximizing the amount of space in contact with the outside, we are maximizing the potential space that can be used for windows. In practical terms, those spaces that require natural light are placed along the edges of the facility.

Of these requirements, the only ones that have been explicitly addressed in previous research on healthcare facility layout concern department areas and shape. Ease of access has indirectly been considered by seeking to minimize total travel distance. However, as was mentioned above, travel distance is only one component of the ease of access equation. In this research, a model that can handle all the above-mentioned requirements is presented. Space Syntax methods are used to achieve this.

3.3 The Space Syntax Approach vs. the Traditional Approach

To demonstrate the advantages offered by the Space Syntax approach, consider the problem of modeling closeness. The traditional FLP literature uses the notion of closeness ratings to do this (see 2.1.2.1.3 in Chapter 2). The efficiency of the layout is a function of the ratio of adjacency requirements that are met, and is defined as (Yapicioglu & Smith, 2012a):

$$\varepsilon = \frac{\sum_{i=1}^{n-1}\sum_{j=i+1}^{n}\left(c_{ij}^{+}x_{ij}\right) - \sum_{i=1}^{n-1}\sum_{j=i+1}^{n}\left(c_{ij}^{-}\left(1 - x_{ij}\right)\right)}{\sum_{i=1}^{n-1}\sum_{j=i+1}^{n} c_{ij}^{+} - \sum_{i=1}^{n-1}\sum_{j=i+1}^{n} c_{ij}^{-}} \tag{3.1}$$

Where

$$x_{ij}\begin{cases}1 & \text{if department } i \text{ is adjacent to department } j, \forall i, j; i > j\\ 0 & \text{otherwise}\end{cases}$$

The values for c_{ij}^{+} and c_{ij}^{-} are the closeness ratings, and they denote the desirability of the departments being adjacent. The common practice is to use the values shown in Table 6.

Table 6: Closeness Ratings Table

Rating	Definition	c_{ij}
A	Absolutely necessary	125
E	Especially important	25
I	Important	5
O, U	Ordinary closeness	1
X	Undesirable	-25
XX	Prohibited	-125

The efficiency is then used to penalize solutions where departments do not meet the closeness requirements. Notice, however, that this approach can only be used to model adjacency restrictions (A needs to be adjacent to B, or A should not be adjacent to B) and cannot be used to model closeness as such (A needs to be close or far from B).

We can use a Space Syntax approach to model closeness as follows. Using departments as the spatial unit and adjacency as the spatial relation between them, construct an adjacency graph. The depth between departments i and j is the number of intervening departments between both. To enforce closeness restrictions, add the constraints shown in Table 7 to the model.

Table 7: Closeness constraints

Scenario: departments i and j need to be…	Constraint
Close	$d_{i,j}^{adj} \leq a$
Far	$d_{i,j}^{adj} \geq a$
Adjacent	$d_{i,j}^{adj} = 1$

Not adjacent	$d_{i,j}^{adj} > 1$

Where $d_{i,j}^{adj}$ is the adjacency depth between departments i and j, and a is a constant provided as a parameter.

Using this approach allows modelers to impose adjacency restrictions (as the traditional method does) but also gives them better control over the way the departments can be related to each other.

3.4 A Space Syntax Approach to Block Layout Design

The Space Syntax methodology must be modified for use in design. The modified procedure is as follows:

1) Define the operational requirements.
2) Define the spatial principles that support meeting operational requirements.
3) Define the spatial unit to be used.
4) Define the spatial relations which can be used to model the spatial principles.
5) Define the spatial variable (or measure) that will quantify the spatial relation.
6) Define the objective function and/or constraints in terms of the spatial variables defined above.

The application of this procedure to the block layout problem for hospitals is as follows:

(a) The operational requirements of interest are privacy and infection control, ease of access, availability of natural light, and noise reduction.

(b) To achieve privacy and infection control, as well as noise reduction, a separation of private spaces from public or noisy areas is needed. Ease of access is a function both of closeness and of the complexity of movement required. Availability of natural light is achieved by placing departments along the outside perimeter. How close or far a space is from another is the principle called proximity. The complexity of movement is measured by how easy or hard it is to reach one space from another by moving in a straight line. This principle is called accessibility. Both of these are defined mathematically in d), e) and f).

(c) The spatial units used are departments and continuous department edges.

(d) The two spatial relations considered are adjacency and straight-line access. Two departments *i* and *j* are adjacent if they share a common edge. Two departments *i* and *j* are straight-line accessible if one department can be reached from the other by moving in an uninterrupted straight line along department edges, excluding those along the facility's perimeter (see Figure 9).

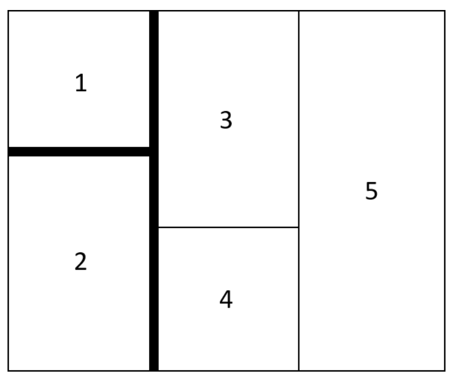


Figure 9: Straight-line access for department 1. The thick lines represent the department edge projections. Departments 2, 3 and 4 are accessible by moving in a straight line.

The straight-line access graph uses the department edge projections as its spatial unit, as seen in Figure 10 a) and b). The dotted lines represent the department edge projections (denoted with letters). Edge projections that intersect with each other are connected on the graph. Each department is associated with the edge projections that touch them. For instance, department 1 is associated with edge projections A and C, while department 5 with B and D. To go from 1 to 5, one must walk along edge A and then turn onto D. Thus, the accessibility depth between departments 1 and 5 is equal to two. Since departments are associated with edge projections, we can also represent the accessibility graph in terms of departments. In this representation, all departments that share an associated edge projection are connected (see Figure 10 c).

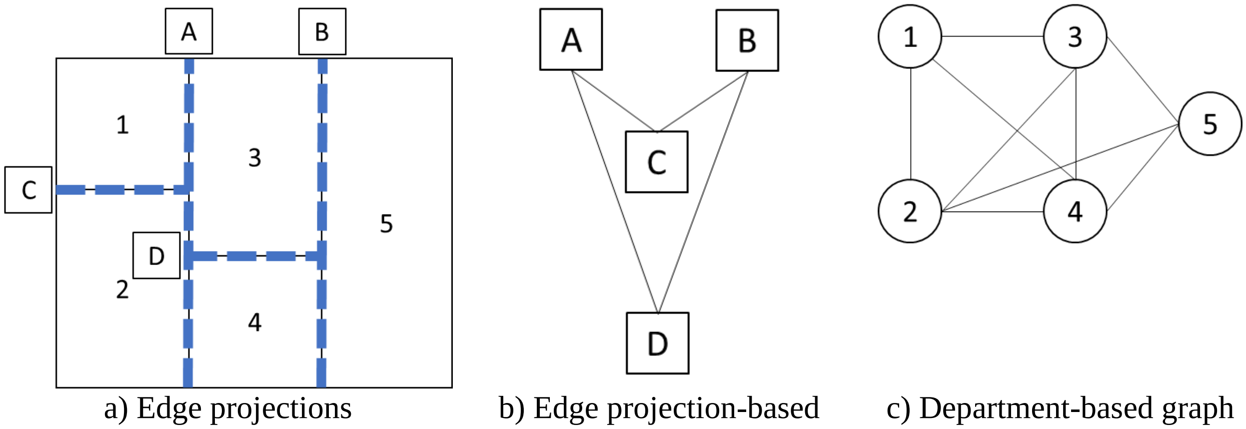


Figure 10: Straight-line access graph.

(e) The spatial measure to be computed is depth. Depth on the adjacency graph is called adjacency depth (denoted $d_{i,j}^{adj}$), and along the straight-line access graph, accessibility depth (denoted $d_{i,j}^{acc}$). Adjacency depth measures proximity (the deeper two spaces are from each other, the more intervening departments there are). When it is equal to 1, the two departments are adjacent. Accessibility depth measures how many continuous department edges must be walked along when traveling between two departments. The number of turns required to go from one department to another is equal to the accessibility depth minus one.

(f) Three objectives are considered. One is to minimize total accessibility depth, which aligns departments along their edges. This allows the imposition of corridor networks with fewer intersections that can be more easily navigated, an important requirement in physical rehabilitation hospitals. A second objective is to minimize total travel distance, measured using the rectilinear distance metric between department centroids, which reduces staff fatigue, improves staff responsiveness to patient summons, and increases time at bedside. A third objective is to minimize total trips weighted by the number of turns required per trip, which reduces congestion and facilitates moving patients on wheelchairs. These are combined into a single objective where each is weighted by a trade-off parameter. The objective is subject to constraints on department area, aspect ratio, and contact with the outside, as well as straight-line access and proximity constraints for certain departments.

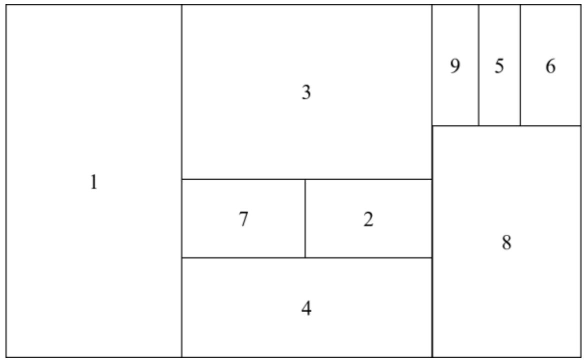
3.5 Graph Construction and Depth Calculation

An important part of the Space Syntax approach is the construction of the required graphs. The algorithm for constructing the adjacency graph is shown in Algorithm 2. For the accessibility graph, the algorithm is the same, except that the condition for adding an edge is that both departments are straight-line accessible (line 4 in the algorithm, shown in brackets).

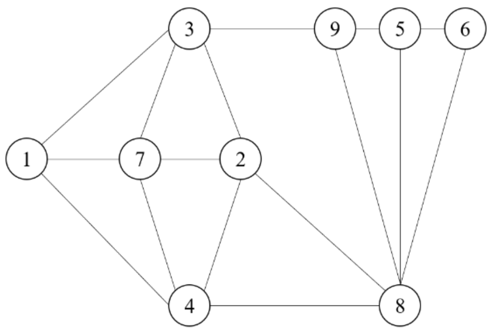
```
INPUT: node list containing a node for each department, empty list of edges
OUTPUT: completed graph
1: For i=1 to number of departments - 1
2:     For j=i+1 to number of departments
3:           If departments i and j are adjacent [straight-line accessible] Then
4:                 Add edge i,j to graph
5:           End If
6:     End For
7: End For
```

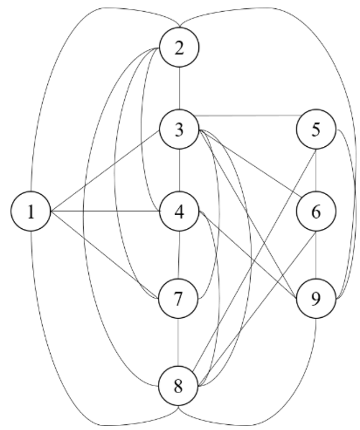
Algorithm 2: Spatial Relation Graph Construction

As an example, consider the block layout shown in Figure 11 a). Using the procedure described above, both the adjacency and the straight-line access graphs shown in Figure 11 b) and c) are constructed.



a) Block layout





b) Adjacency graph c) Straight-line accessibility graph

Figure 11: Block layout and its corresponding adjacency graph

To calculate depth, a weight of 1 to each edge on the graph is assigned, and the depth between two departments is the length of the shortest path between them. These can be calculated using any shortest path algorithm. This research used the Floyd-Warshall algorithm to find the shortest distance between every pair of nodes in the graph. An implementation of the algorithm that runs on the computer's graphic card is used to significantly accelerate the calculations. The implementation was obtained from the Alea GPU gallery and can be downloaded from http://www.aleagpu.com/release/3_0_4/doc/samples/aleasample_cs_floydwarshall.html. The implementation is based on the algorithm described in Lund and Smith (2010).

IV. Space Syntax-Based Block Layout Model for Physical Rehabilitation Hospitals

This chapter introduces the Space Syntax-based model developed to solve the block layout problem for physical rehabilitation hospitals. The notation is introduced first, followed by the model and the solution procedure used to solve it. The chapter concludes with the results of the experiments run to evaluate the impact of varying the trade-off parameters on the resulting designs.

4.1 Notation

Parameters

α: trade-off between total accessibility depth and total travel distance.

β: trade-off between total accessibility depth and total trips weighted by turns required.

n: total number of departments in the facility.

l_i, u_i: lower and upper bounds for the area of department i.

ar_i: aspect ratio limit for department i.

$p_{i,j}^{min}, p_{i,j}^{max}$: minimum and maximum proximity permitted between departments i and j.

c_i: required closeness of department i to the outside. Equals 1 if the department should be along the facility's edge. Otherwise, it can be given a very large value.

$t_{i,j}^{max}$: maximum number of turns allowed between departments i and j.

Decision Variables

$d_{i,j}^{adj}$: adjacency depth between departments i and j. If with the outside, $j = o$.

$d_{i,j}^{acc}$: accessibility depth between departments i and j.

$d_{i,j}$: distance between departments i and j.

$f_{i,j}$: number of trips between departments i and j.

w_i, h_i: width and height of department i.

a_i: area of department i.

4.1.1 Objective Function

The proposed objective function is:

$$Min\ Z = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \left(d_{i,j}^{acc} - 1\right) + \frac{1}{\alpha} \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} d_{i,j} f_{i,j} + \frac{1}{\beta} \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} d_{i,j}^{acc} f_{i,j} \qquad (4.1)$$

The first component of this objective function is the facility's total accessibility depth, the second component is the total travel distance (measured between department centroids using the rectilinear distance metric), and the third is the total number of trips weighted by the number of turns required per trip. For the remainder of this chapter, the first component will be denoted ACC, the second TD, and the third TT. Their corresponding units are total turns, feet (or some other continuous distance metric), and total turns per trip. The parameters α, β are provided by the facility designer, and measure by how many units of TD and TT the objective function needs to decrease in exchange for a unit increase in ACC. That is, they represent the trade-off between the first objective function component and the other two. β/α is the trade-off between TD and TT.

This is proven by the following:

Let $\Delta Z = (ACC - ACC_0) + \frac{1}{\alpha}(TD - TD_0) + \frac{1}{\beta}(TT - TT_0)$, the change in the objective function as its components change.

Let $ACC - ACC_0 = 1$, that is, ACC increases by one unit and $TT - TT_0 = 0$, indicating that there was no change in TT. Then,

$$\Delta Z = 1 + \alpha(TD - TD_0)$$

If the objective function remains constant, then $\Delta Z = 0$, and thus,

$$TD - TD_0 = -1/\alpha$$

In other words, an increase of one unit in ACC is worth a decrease of $1/\alpha$ units in TD. The same holds true for changes in TT (except the decrease is equal to $1/\beta$ units). The values of both α, β can range between 1 and ∞. Their specific value depends on the problem and on the designer's preference for one or the other objective function components. An arbitrarily high value (around one million, for instance) will practically eliminate the objective component from consideration, while a value of 1 will give it the highest priority. It is recommended to set these values at several very high and low combinations, together with a range of intermediate values to generate a variety of designs each with different characteristics.

4.1.2 Constraints

The objective function is subject to the following constraints:

$$l_i \leq a_i \leq u_i, \qquad i = 1, \dots, n \tag{4.2}$$

$$\frac{\max(w_i, h_i)}{\min(w_i, h_i)} \leq ar_i, \qquad i = 1, \dots, n \tag{4.3}$$

$$p_{i,j}^{min} \leq d_{i,j}^{adj} \leq p_{i,j}^{max}, \qquad i, j = 1, \dots, n \tag{4.4}$$

$$d_{i,o}^{adj} \leq c_i, \qquad i = 1, \dots, n \tag{4.5}$$

$$d_{i,j}^{acc} \leq t_{i,j}^{max}, \qquad i = 1, \dots, n \tag{4.6}$$

with $1 \leq p_{i,j}^{min} \leq p_{i,j}^{max} < n$ and $1 \leq t_{i,j}^{max} < n$.

Constraints (4.2) and (4.3) are the area and aspect ratio constraints, respectively. Constraint (4.4) is the proximity constraint. For each department pair, there can be a minimum and maximum required proximity. Departments that need to be close will be bounded from above, while those that need to be distant will be bounded from below. Constraint (4.5) ensures that departments that need contact with the outside are placed accordingly. Lastly, constraint (4.6) sets bounds on the straight-line access between departments.

4.2 Solution Procedure

To solve the block layout problem, a tabu search (TS) was developed. The problem was first addressed using a genetic algorithm (GA) which found layouts comparable to those in the literature for common benchmark instance problems. However, when tested with the problem instances developed for this research, it did not perform as well. A TS was developed to compare the two approaches and it outperformed the GA. The GA converged on feasible solutions less frequently than the TS, and the variability in the objective function values of the solutions was significantly higher, as can be seen in Figure 12.

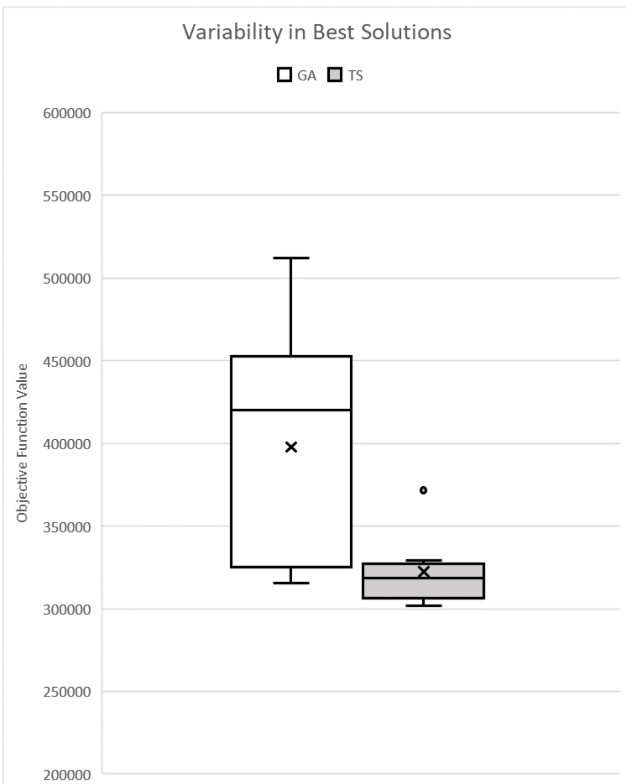
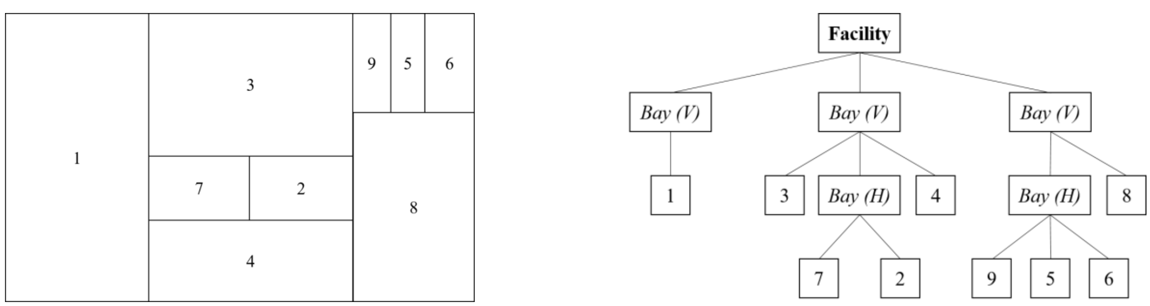


Figure 12: Variability in solutions GA vs. TS

4.2.1 Solution Encoding

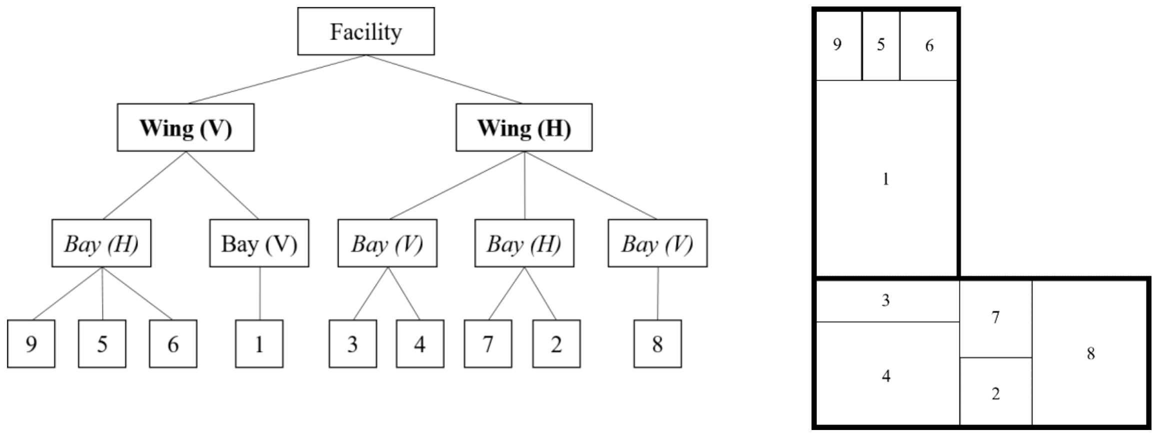
Solutions were encoded with a nested bay structure implemented using XML (eXtensible Markup Language). This representation was developed by the author. The nested bay layout representation is an extension of the flexible bay structure (FBS) (Tate & Smith, 1995b) that allows for bays to exist within other bays. The conceptual model of the nested bay structure is a tree, where the leaves are departments and the internal nodes are bays. The tree's root is the facility itself. This tree is called a nested bay tree. This representation can be used to encode more complex layouts than those permitted by FBS. For rectangular facilities it can encode the same layouts that can be represented using a slicing tree (see Friedrich et al., 2018 for a recent description of the slicing tree representation), but it can also represent non-rectangular facilities, something that is not possible with slicing trees. Figure 13 shows a nested bay layout along with its corresponding nested bay tree.



a) Sample layout b) Nested bay tree

Figure 13: Nested bay layout and its corresponding tree

To represent non-rectangular facilities, a first layer of "wings" (which can be thought of as bays with fixed dimensions and position) is added. Figure 14 shows a non-rectangular facility together with the nested-bay with wing structure representing it. The thicker lines are the wing boundaries.



a) Non-rectangular nested bay tree b) Non-rectangular layout

Figure 14: Non-rectangular layout representation using nested bays

The nested bay tree is implemented using XML. XML documents are structured as trees. Hence, the conceptual model and its implementation both possess the same structure, which means any change to the XML tree results in the same change to the layout itself. In addition, most modern programming languages include libraries that make manipulating and querying XML trees a simple task. Furthermore, additional information about the layout can be attached to the XML document such as department dimensions, constraint violations, corridor structure, etc. Figure 15 shows an example of the XML document of the facility shown in Figure 14. Besides the nested bay structure, it contains department dimensions (as attributes of the <department> tags) and details about the constraints being violated (in bold).

Decoding a layout from a nested bay tree consists of finding the dimensions and positions of each department in the layout. The nested bay representation makes use of the same principle used in FBS to determine departmental dimensions. The calculation is simplified by forcing departments in the same bay to have a common dimension. The other dimension can then be calculated using the department’s area. This idea is used recursively to find the size and location of each department in a nested bay representation. An important feature of the nested bay tree is that there is a one-to-one relationship between the nested bay tree and the actual layout. Lower level bays in the tree are nested within their parents in the actual layout.

```
<facility>
   <wing x=“0” y=“0” direction=“vertical”>
      <bay direction=“horizontal” constraints_violated=“0”>
         <department id=“9” x=“0” y=“0” />
         <department id=“5” x=“7.5” y=“0” />
         <department id=“6” x=“12.5” y=“0” />
      </bay>
      <bay direction=“horizontal” constraints_violated=“1”>
          <department id=“1” x=“0” y=“10” violates_area_constraint=“true” />
       </bay>
   </wing>
   <wing x=“0” y=“50” direction=“horizontal”>
      <bay direction=“vertical” constraints_violated=“1”>
         <department id=“3” x=“0” y=“50” violates_aspectratio_constraint=“true” />
         <department id=“4” x=“0” y=“55” />
      </bay>
      <bay direction=“vertical” constraints_violated=“0”>
          <department id=“7” x=“20” y=“50” />
          <department id=“2” x=“20” y=“60” />
       </bay>
       <bay direction=“vertical” constraints_violated=“0”>
           <department id=“8” x=“30” y=“50” />
       </bay>
   </wing>
</facility>
```

Figure 15: Sample XML document

The decoding algorithm is:

```
INPUT: wing dimensions and position, minimum department areas
OUTPUT: department and bay dimensions and positions
//Find department and bay areas
Assign each department its minimum area
For Each wing in facility
  If sum of department areas in the wing > wing area Then
    Excess area = sum of department areas – wing area
    While Excess area > 0
      Choose a department from the wing at random
      Area decrease = min{department’s area / 2, Excess area}
      Decrease department’s area by Area decrease
      Excess area = Excess area - Area decrease
   End While
  Else If sum of areas of departments in the wing < wing area Then
   Empty area = wing area – sum of department areas
   For Each department in the wing
      Department area = Department area + Empty area / Departments in wing
  End If
   For Each bay in wing
     Bay area = sum of department areas in the bay
```

```
  End For
End For
//Find department dimensions
For Each wing in facility
  For Each descendant in wing //Includes both bays and departments
    Get parent dimensions, location coordinates, and direction
    If direction is "horizontal" Then
//Set the dimensions
      Descendant's height = parent height and descendant's width =
          descendant's area / descendant's height
//Set the location
      Descendant's x = parent's x + sum of previous sibling's widths
      Descendant's y = parent's y
    Else //Direction is "vertical"
//Set the dimensions
      Descendant's width = parent width and descendant's height =
          descendant's area / descendant's width
//Set the location
      Descendant's x = parent's x
      Descendant's y = parent's y + sum of previous sibling's heights
    End If
  End For
End For
```

Algorithm 3: Decoding a nested bay representation

Department areas, as well as the facility's dimensions are provided as inputs. The facility's size is fixed, while department areas can be increased or decreased to fit their parent bay. The area constraints (both upper and lower limits) in the model ensure that department sizes meet the requirements provided by the facility designer.

The nested bay encoding was chosen over the slicing tree encoding for the following reasons: (1) The nested bay representation is more straightforward (the slicing tree represents 'how' the facility is constructed, while the nested bay represents how it looks). (2) The conceptual model and the data structure that implements it are both trees. (3) The nested bay structure can be, in turn, embedded in other structures (wings, floors) so that non-rectangular and multistory facilities can be represented (see Figure 14). (4) The simplicity in finding department dimensions that is one of the advantages of FBS is preserved while allowing for the representation of more complex layouts.

4.2.2 Move Operators and Neighborhood Definition

At each iteration of the TS, two neighborhoods are defined in sequence. The first neighborhood is generated from a 2-way swap between all departments in the layout, excluding those of the same type (two patient rooms will not be swapped with each other). The best solution resulting from the swap move is selected, and the second neighborhood is constructed by modifying the nested-bay tree. The tree is modified using five modification operators.

(1) At each level of the tree, an insertion of new bays between already existing bays is attempted. These potential new bays are shown as dotted squares in Figure 16. The new bay takes the first department from the bay to its right, and the last department from the bay to its left. For the first and last bays, these take two departments from their neighbors. If there are only two departments in an already existing bay, placement of a new bay in the two potential locations adjacent to it is not allowed.

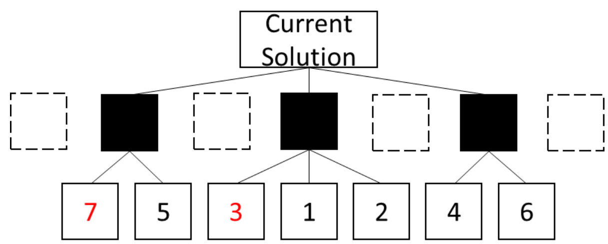
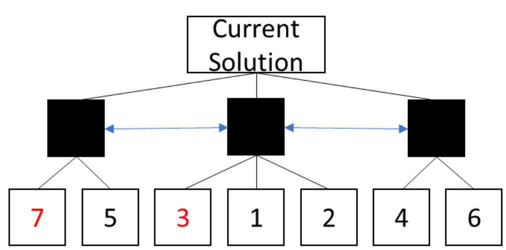
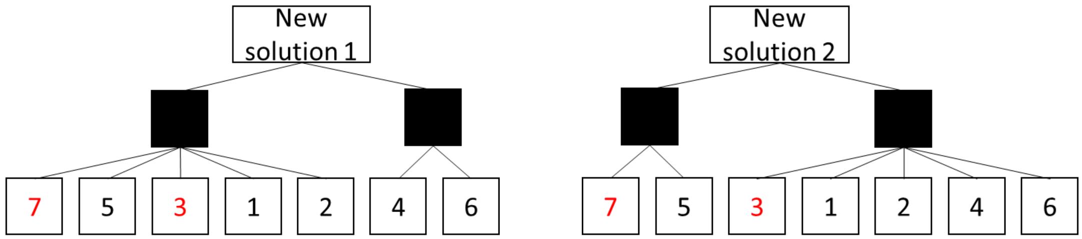


Figure 16: First tree modification operator

(2) At each level, merge adjacent bays (Figure 17).



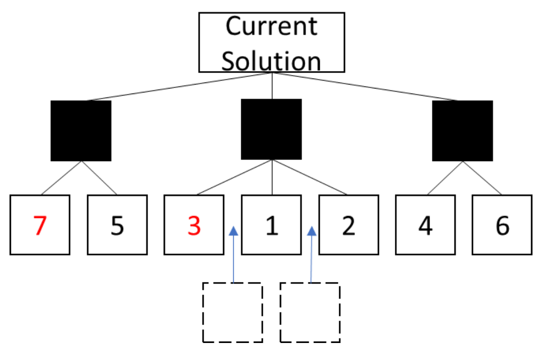
a) Original solution



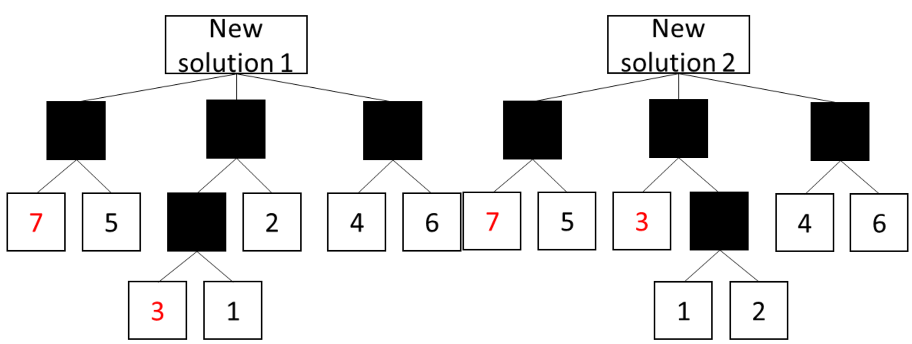
b) Two neighbor solutions obtained by merging bays

Figure 17: Second tree modification operator

(3) For each bay in the tree, if it has more than two elements, add sub-bays between them. The children elements of each new sub bay are taken one from the element to its left, and one from the element to its right.



a) Original solution



b) Two neighbor solutions obtained by adding sub-bays

Figure 18: Third tree modification operator

(4) For each bay in the tree, remove its sub-bays, one by one. Its descendants are moved to the higher level.

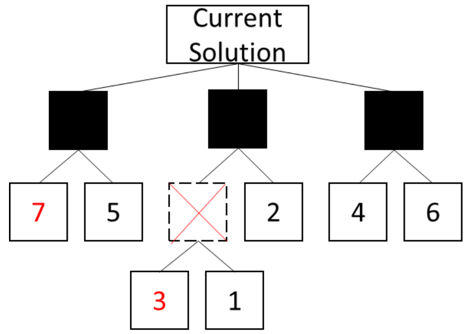


Figure 19: Fourth tree modification operator

(5) For each bay in the tree, if it has more than four elements, split the bay with half the child elements going to one new bay and the rest to the other.

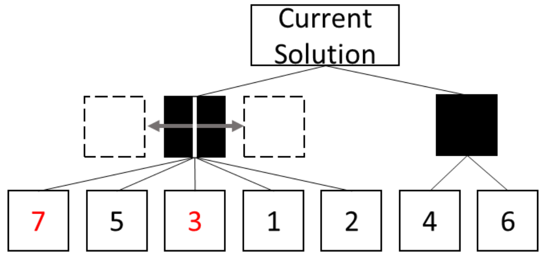


Figure 20: Fifth tree modification operator

The solutions thus generated are evaluated and the best one found becomes the current solution.

4.2.3 Tabu List Entries

Each tabu list entry records the two most recent department types that were swapped. The size of the tabu list is determined randomly between an upper and a lower bound and is changed every 75 iterations. The values of the bounds, as well as the number of iterations for the change were determined experimentally. The bounds are set to 15 and 30.

An aspiration criterion is also used to override the tabu mechanism. If a solution has a better fitness than the best solution yet found, it will become the current solution, even if it is tabu.

4.2.4 Fitness Function

The problem set used in this research is highly constrained and includes several different types of constraints: area, aspect ratio, proximity, contact with the outside (which can be considered a subclass of the proximity constraints), and straight-line access constraints. To handle this situation, the adaptive, memory-based penalty function of Kulturel-Konak et al. (2004) was modified and used in the fitness function. The penalty function is:

$$p(x) = \left[\sum_{i=1}^{m}\left(\frac{n_i}{NFT_i}\right)^{k_i}\right]\left(F_{feas} - F_{all}\right) \tag{4.7}$$

where,

$x =$ the current solution,

$m =$ the number of constraint types,

$n_i =$ the number of constraint violations of type i,

$NFT_i =$ the near-feasibility threshold for constraints of type i,

$k_i =$ penalty severity parameter,

F_{feas} = the objective function value for the best feasible solution found so far,

$F_{all} =$ the objective function value of the best solution found so far, regardless of its feasibility.

The initial values of the near-feasibility thresholds are provided as parameters by the facility designer, but they change as the search proceeds based on the number of feasible and infeasible solutions encountered over the tenure of the tabu list. The rate of change is also provided by the designer and is denoted $r, 0 < r < 1$.

At iteration j, the ratio of feasible solutions in the list is $R_j = F_j/T_j$, where F_j is the number of feasible solutions, and T_j is the number of solutions currently in the tabu list. For each constraint type i, if the current move results in a feasible solution, then we update the threshold:

$$NFT_{i,j+1} = NFT_{i,j}(1 + rRj)$$

If the current move produces an infeasible solution, then the threshold is updated as:

$$NFT_{i,j+1} = NFT_{i,j}\frac{(1 + rRj)}{1 + r}$$

This allows the threshold to be shrunk or increased by (approximately) r%, making it a generalization of the function presented in Kulturel-Konak et al. (2004), where the threshold was either shrunk or increased by a fixed 50%. If the search finds many infeasible solutions, the NFT is tightened and the search is pushed towards the region of feasibility. If the search finds many feasible solutions, it is loosened, and the search moves towards the region of infeasibility. The severity parameter is set to 3 and the initial NFTs vary from problem instance to problem instance, but range between 2 and 5 for each constraint type. The value of r is provided by the designer as a parameter of the TS.

4.2.5 Stopping Criteria

The TS has three stopping criteria: (1) a limit on non-improving iterations, (2) a limit on the number of iterations without improving feasible solutions, and (3) a limit on total number of iterations. Stopping criterion (1) stops the run if no feasible solutions are found within a reasonable number of iterations. Stopping criterion (2) overrides (1) if feasible solutions are found. It stops the run when the best feasible has not improved. Stopping criterion (3) overrides the other two and represents an absolute upper bound on the number of iterations allowed.

4.2.6 Initial Solutions

Initial solutions are generated randomly using the following procedure for minimizing the number of constraint violations:

(1) Generate a random permutation of the departments. Initialize the partial nested bay tree with a set of potential locations.

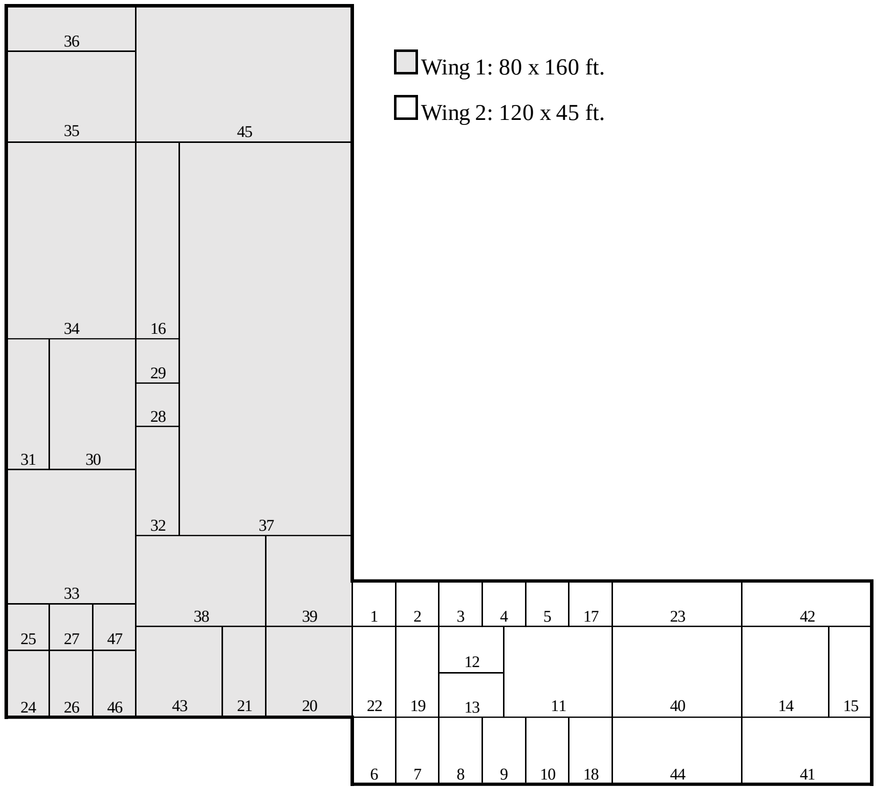
(2) Select the location which will result in the least number of constraint violations. Add the current department to that location.

(3) Repeat for the remaining departments.

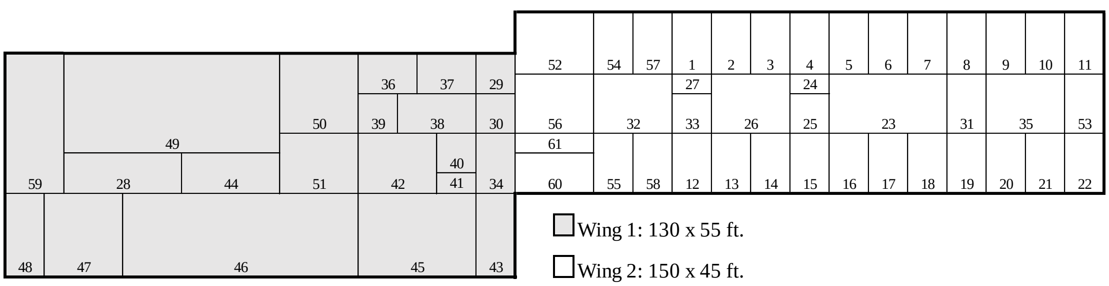
Using the nested bay tree structure alone, we can infer whether placing a department in a certain position will imply a constraint violation or not. For example, given the departments already placed, together with their known minimum required area, we can know whether adding a department will exceed the total permitted area, thus violating an area constraint. Likewise, we know that placing two departments next to each other in the same bay will imply that they are adjacent thus satisfying an adjacency constraint. Similar rules are used for all other constraint types.

4.3 Test Problems

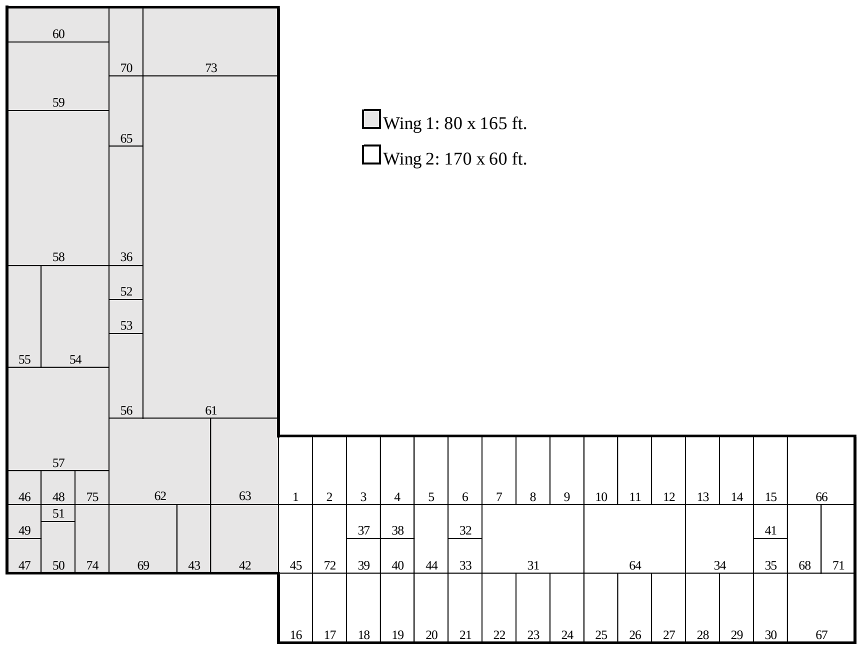
Four test problems of varying sizes and facility shapes are developed to test the proposed solution procedure. All four are based on real-world physical rehabilitation hospitals. The flow matrices were constructed based on the normal operations of the facilities as determined from meetings with staff. The characteristics of each hospital are summarized in Figure 21.



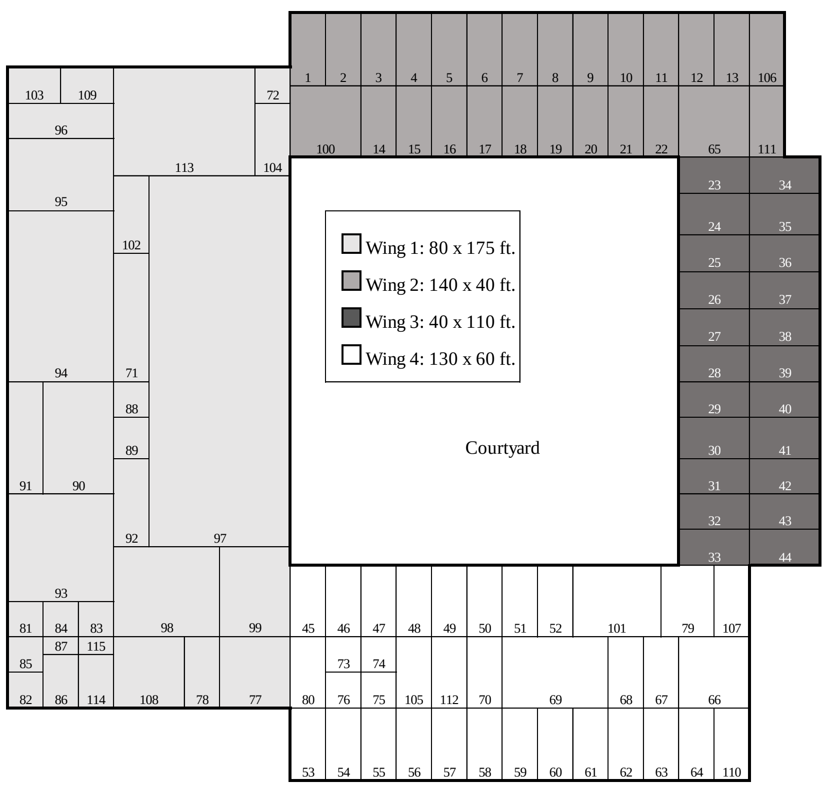
a) 10 Bed – 47 departments



b) 22 Bed – 61 departments



c) 30 Bed – 75 departments



d) 64 Bed – 115 departments

Figure 21: Problem Instances

Department areas were determined based on a review of several hospital floorplans. The maximum aspect ratio was set to 2 for patient rooms (most patient rooms in actual layouts have this ratio), and to 4 for all other departments. For the 64-bed instance, the maximum aspect ratio was set to 4 for all departments. This allows the search to converge on feasible solutions more easily.

Proximity constraints are summarized in Table 8.

Table 8: Proximity Constraints

Department Type	Adjacent to	Close to	Non-adjacent to	Far from
Patient Room		Nursing station	Therapy gym ADL	Dining Maintenance
Pharmacy	Pharmacy Admin			
Therapy Gym	Charting	ADL Therapy Recreation		
ADL		Charting Therapy Recreation		
Waiting room	Reception Exam Room			
Lobby	Desk			
Dining	Kitchen			
Kitchen	Freezer			

Patient rooms are required to have straight-line access of 1 with respect to the nursing station while values of straight-line access should be low with respect to the Therapy Gym, ADL, and Dining, since patients and staff are continuously moving to and from these locations. In terms of contact with the outside, patient rooms, dining area, the therapy gym and the lobby are all required to be along the edges of the facility. The total number of constraints depends on the total number of departments. The total number of constraints per problem is shown in Table 9.

Table 9: Total constraints per problem

Problem	Total number of constraints
10 Bed	221

22 Bed	355
30 Bed	463
64 Bed	820

The total number of trips among departments were generated using a uniform distribution, where the values of the lower and upper bounds were determined based on rough estimates of the number of people usually found within each department type and the number of trips they would take on an average day. For example, the number of trips from a patient room to the dining area was obtained using a $U(3,6)$, because there is usually a single patient in a room who goes to the dining area 3 to 6 times a day. Likewise, the trips between the nurse station and the staff lounge was obtained using a $U(20,25)$ because 4 to 5 nurses are expected to go to the lounge approximately 5 times a day. The data used for each problem instance are found in Appendix A.

4.4 Computational Experience

The TS was run on an Intel Xeon 3.5 GHz CPU with 64 GB of Ram. Since the dominating stopping criteria was the number of non-improving feasible solutions, each run had different runtimes, which varied greatly. Some runs converged on a solution rapidly while others took significantly longer. To compare the runtimes across problem instances, these were adjusted to be per iteration, as shown in Figure 22. Runtimes per iteration ranged from 6 seconds for the smallest problem instance, to almost 3 minutes for the largest. This means that a 500-iteration run of the largest instance can take up to 20 hours. Long computational times are not a concern for this problem, since the facility design process is usually a lengthy one.

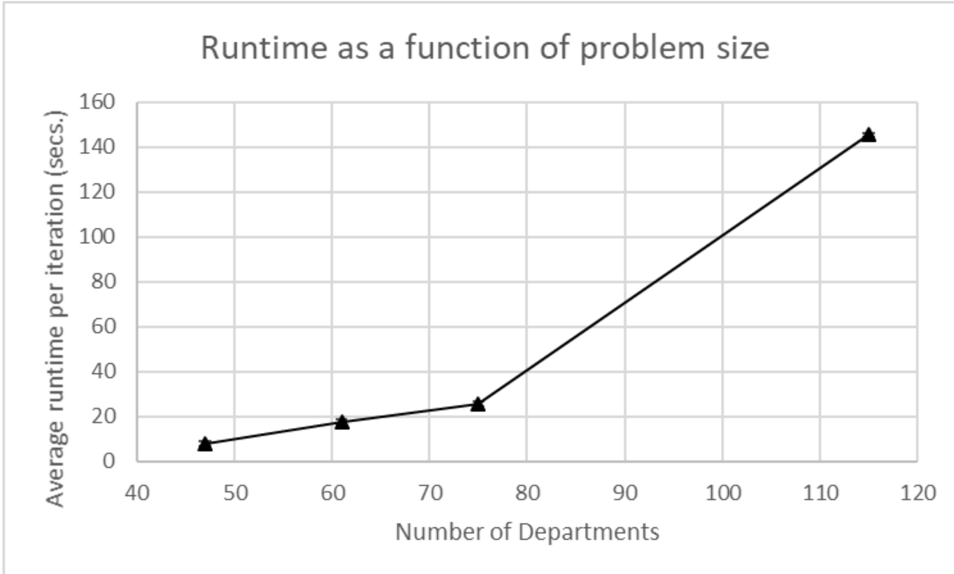


Figure 22: Mean runtime per iteration as function of problem size

For the larger instances, much of the computational effort was expended in finding feasible solutions. Given the highly constrained nature of the problem, better quality initial solutions have an important impact on the performance of the search. Nevertheless, feasible solutions were found for all problem instances, even when starting from randomly generated solutions. That said, not all runs found feasible solutions, and the proportion of runs that did decreased as problem size increased.

To verify that the TS converges, five long runs for each problem instance were performed. For each set of runs the spread in terms of the best fitness found was measured. Only area and aspect ratio constraints were considered, as well as the travel distance objective. This was done to significantly speed up the search, thus allowing it to run for more iterations. In addition, this relaxed version of the problem provides an approximate lower bound for the travel distance objective, which can be used to assess the quality of the solutions obtained in the fully constrained version of the problems. Table 10 summarizes the findings of this test. The spread is measured as the difference in percentage from the best and worst solutions found to the mean for each set of runs. The initial spread is that between the randomly generated, initial solutions.

Table 10: Convergence of the TS algorithm for each problem instance

Problem instance	Total iterations	Initial spread	Final spread	Best TD
10 Bed	15,000	341.25%	2.87%	667,503
22 Bed	7,500	454.70%	1.80%	1,218,000
30 Bed	5,000	499.80%	2.94%	2,170,972
64 Bed	1,500	195.56%	4.33%	8,776,921

Figure 23 shows the convergence of the five runs for the 64-Bed problem instance. Similar patterns were observed in all other problem instances.

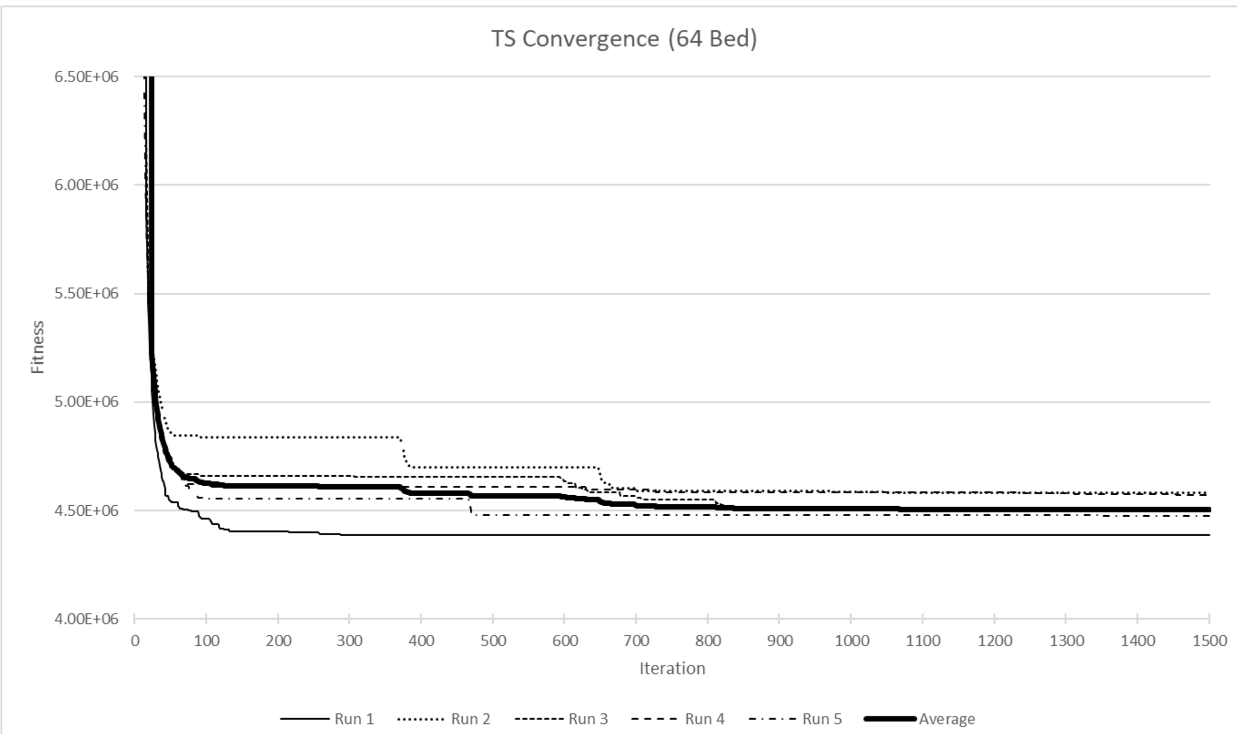
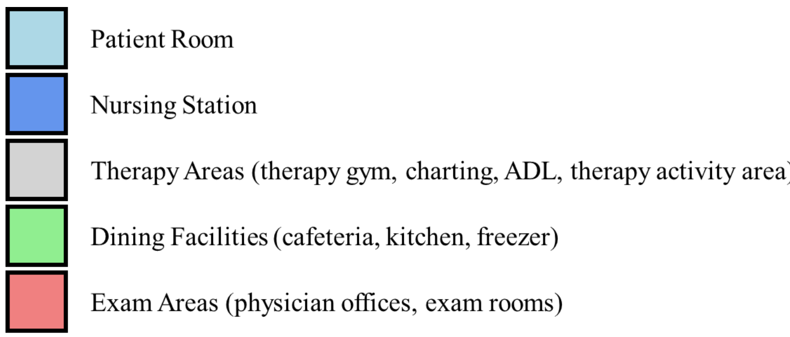


Figure 23: Convergence of the TS algorithm for the 64-Bed problem instance

4.5 Results

A series of experiments was conducted to evaluate the impact of changing the trade-off parameters. The first experiment evaluated setting the parameters to three different levels (low, medium, high) for the 10-bed problem. The second experiment set the parameters at two different levels (low and high) for the 22-bed problem. The third experiment set the parameters at both high and low levels for the 30 and 64-bed problems. To close the section on results, the quality of the solutions obtained by the TS algorithm was assessed by comparing the block layouts obtained in experiments 1 through 3 with the block layouts based on real hospitals described in Section 4.3.

The following color coding was used in all block layout figures:



4.5.1 Experiment 1

For each combination of the parameter values, ten runs of the TS were executed beginning with a randomly generated solution. The stopping criteria were set to 200 non-improving iterations, 25 non-improving feasible iterations, or a maximum of 500 total iterations. Initial NFT values were set to 2 for each constraint type. These values were determined because they produced good feasible solutions within relatively short runtimes. The parameter values were set according to Table 11.

Table 11: Experiment 1 trade-off values

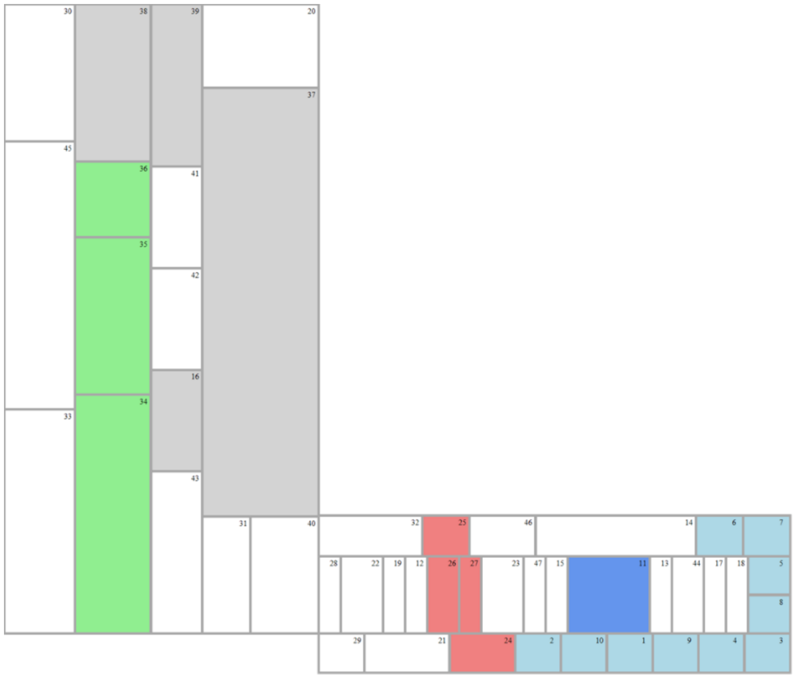
	Values	
Level	TD Tradeoff (α)	TT Tradeoff (β)
Low (L)	10,000	1,000
Medium (M)	1,000	10
High (H)	1	1

A value of 10,000 (the low level of TD tradeoff) means that an increase in ACC is permitted if it results in a decrease of 10,000 units of TD. In consequence, the TD component of the objective function has a very low influence in the objective function value. A summary of the results of this experiment is shown in Table 12.

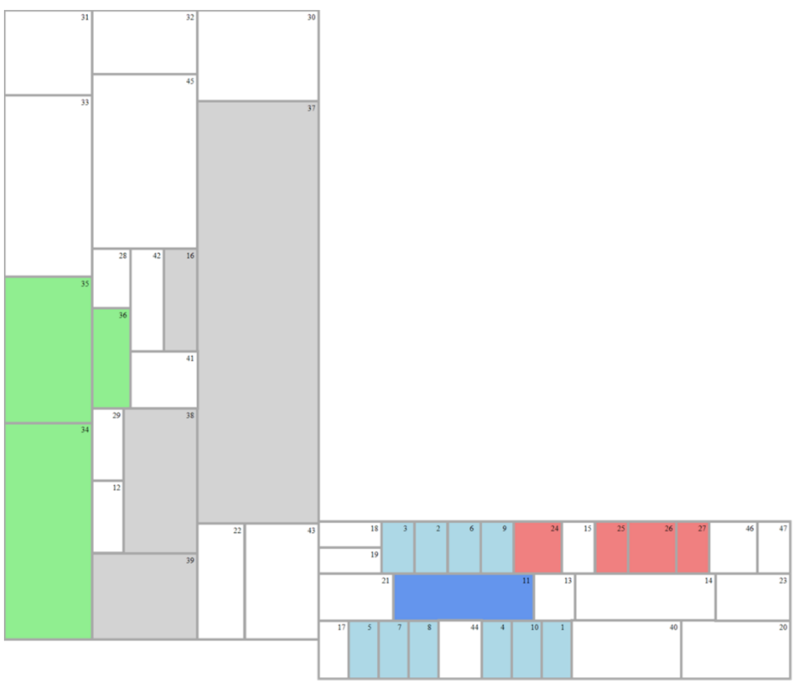
Table 12: Results experiment 1 (averages over 10 runs). An asterisk indicates that the best value of the corresponding objective function component was found at that level combination.

α level	β level	TD	ACC	TT
L	L	947588.61	1828*	4267
L	M	1057437.75	2466	4377
L	H	1016992.89	2371	3541
M	L	900178.82	2326	5570
M	M	956309.59	2299	4079
M	H	1049734.19	2538	4781*
H	L	917300.78	3141	8424
H	M	875923.07*	3103	7059
H	H	872612.87	3220	7045

The solutions with the best values for each objective function component are shown in Figure 24. Not surprisingly, the best ACC solution was found when both TD and TT tradeoffs were at their low values, when ACC was the most significant component of the objective function. Likewise, the best TD and TT were found when their corresponding tradeoffs were at a high level.



a) Best ACC – 1038 (TD = 854,811, TT = 2044)



b) Best TD – 757,530 (ACC = 2984, TT = 4910)

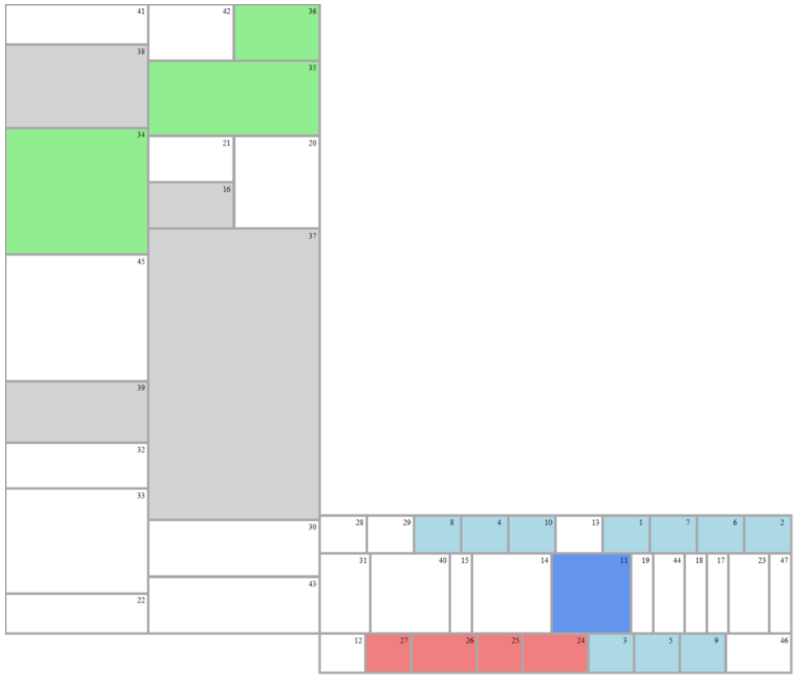

c) Best TT – 1960 (TD = 1,038,079, ACC = 1470)

Figure 24: Best solutions found – experiment 1

Several design patterns also seem to be related to the tradeoff levels. Solutions with low tradeoffs for both TD and TT such as Figure 24 a), have little if any nesting. All departments are in the highest-level bays resulting in a very shallow nested bay tree. Solutions with high TD tradeoff, on the other hand, make heavy use of nesting to reduce TD at the expense of ACC. TT has an intermediate effect. Figure 25 shows the best solution for each tradeoff value combination, starting with low level on the upper left corner.

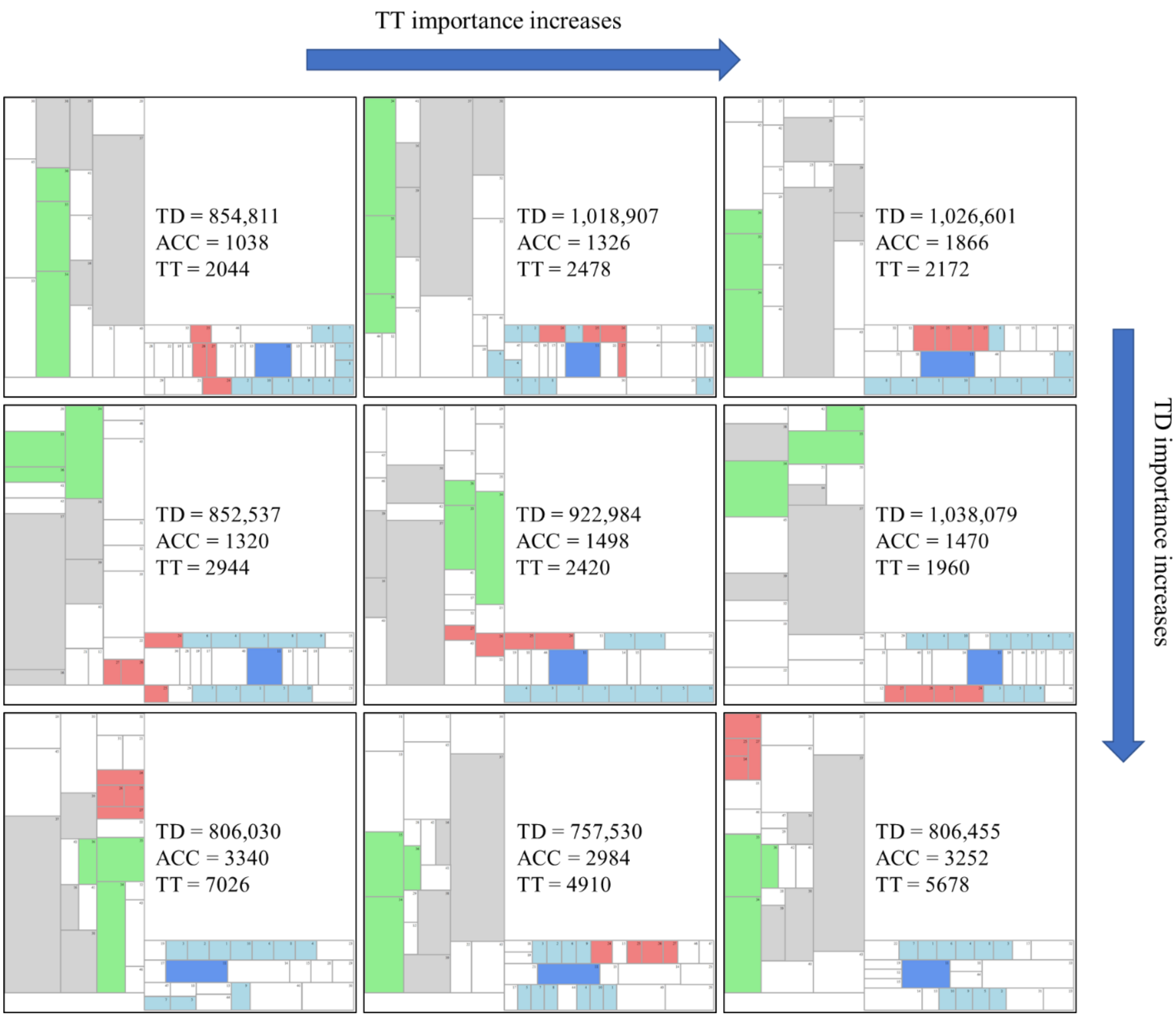


Figure 25: Layout changes as tradeoffs change

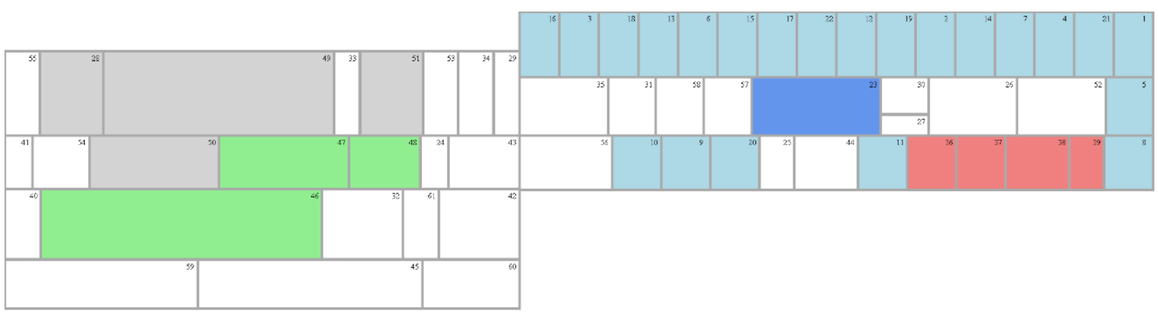
4.5.2 Experiment 2

A similar experiment was performed with the 22-bed instance. This experiment considered only the low and high levels of the tradeoffs. The experiment size was reduced due to the longer runtimes required for solving the larger problem instance. The results observed were similar, as is shown in Table 13.

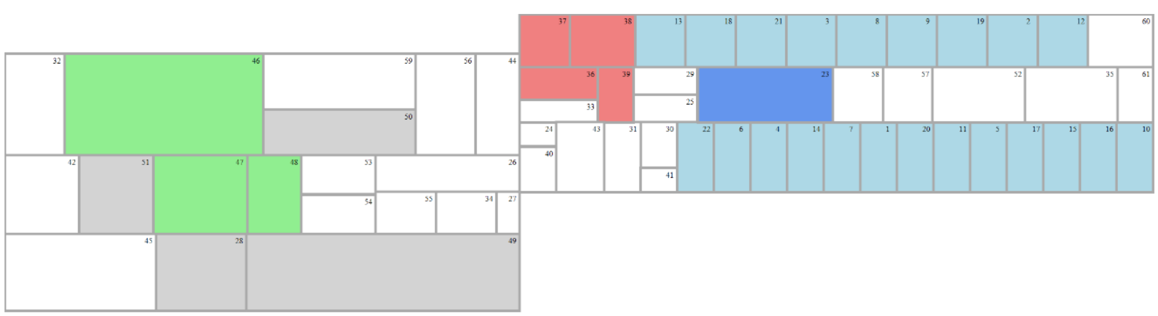
Table 13: Results experiment 2 (averages over 10 runs). An asterisk indicates that the best value of the corresponding objective function component was found at that level combination.

α level	β level	TD	ACC	TT
L	L	1,575,710	3,092	8,184
L	H	1,711,750	3,377*	6,588*
H	L	1,451,884*	3,784	11,317
H	H	1,452,354	3,725	10,350

As with experiment 1, the layouts with the best value of ACC are found when the other two components have a minor role in the objective function. However, the solution with the best ACC of all was found when the TT component was at a high level. It is also the best solution for TT (see Figure 26). Despite this specific case, not all solutions with good ACC have good TT.



a) Best ACC – 2060, Best TT – 4534 (TD = 1,790,595)



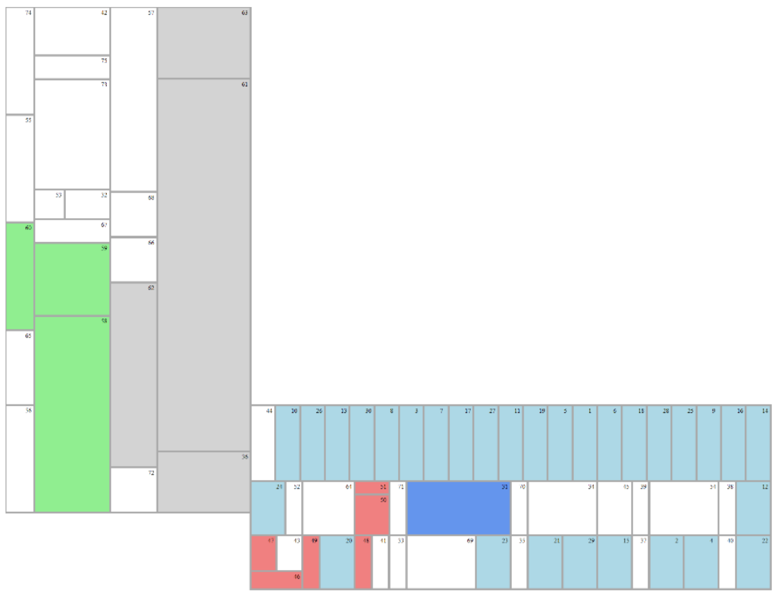
b) Best TD – 1,371,120 (ACC = 3540, TT = 11,664)

Figure 26: Best solutions found – experiment 2

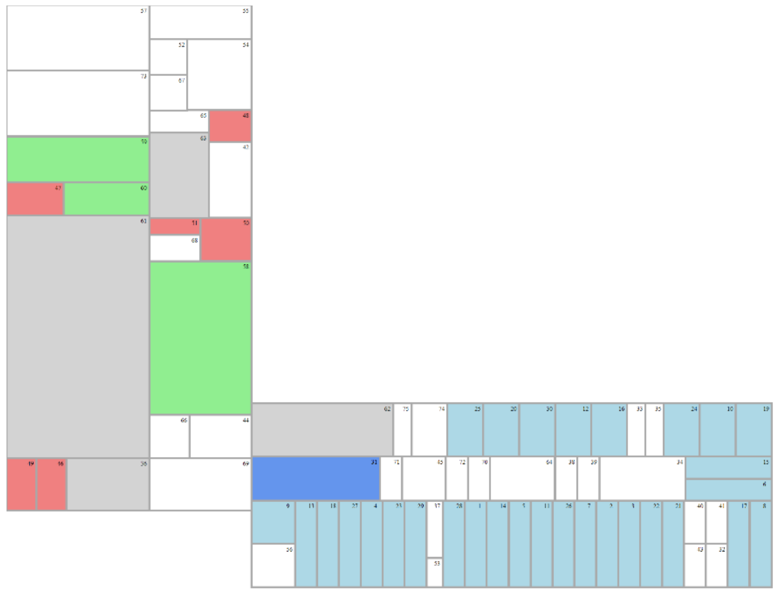
Once again, the lower the value of the ACC component of the objective, the less nesting is observed. The higher the TD component value, the more nesting results.

4.5.3 Experiment 3

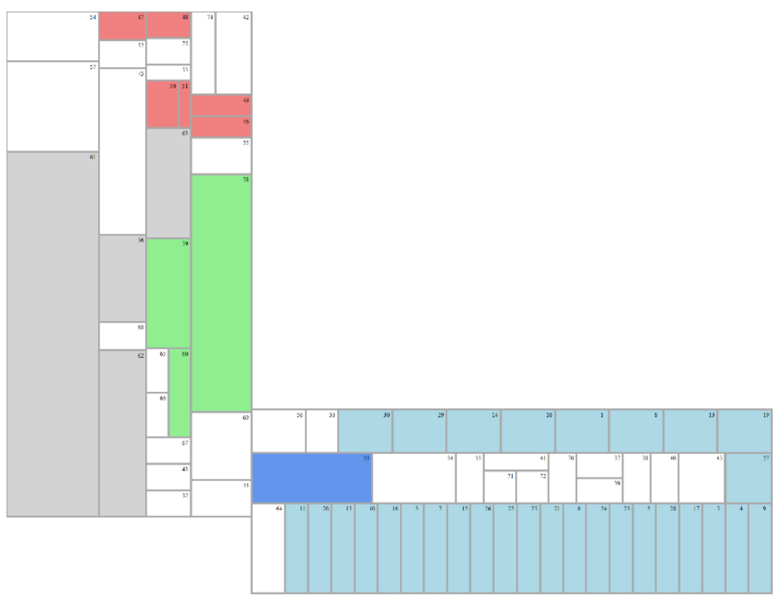
For the two largest problem instances, five runs for the low-low, low-high, and high-low combinations of parameter values were executed. The results found for the smaller instances once again were confirmed. The best layouts are shown in Figure 27, where it is once again noticeable that those with best TD have more nesting than those with better ACC, while those with best TT are somewhere in between.



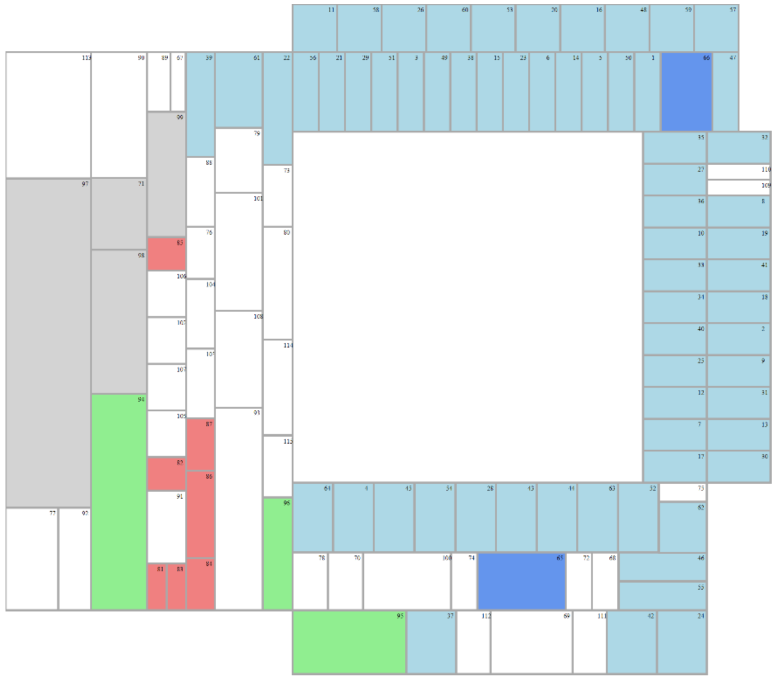
a) Best ACC = 6644 (TD = 2,701,032, TT = 18,254) – 30 Bed



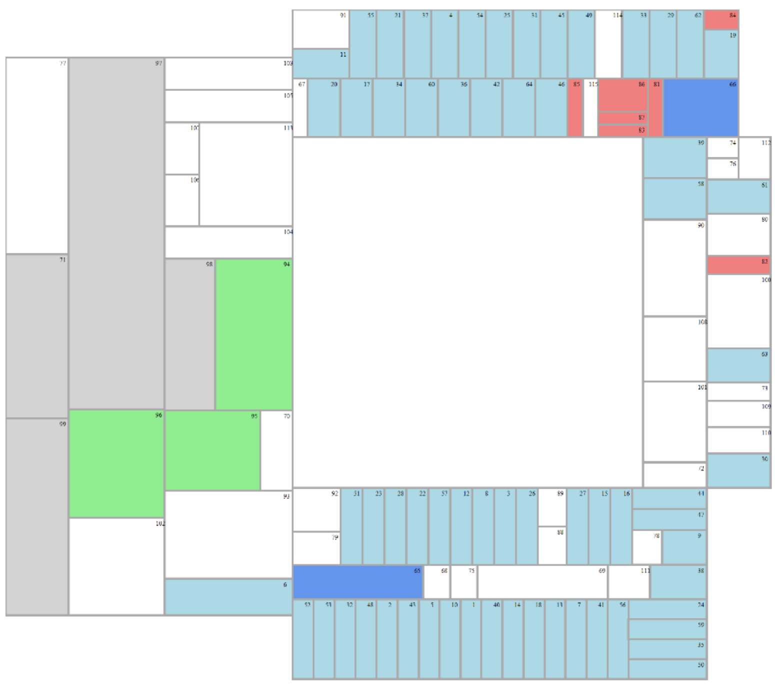
b) Best TT = 13,190 (ACC = 7886, TD = 2,942,119) – 30 Bed



c) Best TD = 2,385,060 (ACC = 7706, TT = 15,886) – 30 Bed



d) Best ACC = 15,918, Best TT = 98,054 (TD = 10,484,697) – 64 Bed



e) Best TD = 9,957,551 (ACC = 27,100, TT = 128,668) – 64 Bed

Figure 27: Best layouts for 30 and 64 bed problem instances

4.5.4 Solution Quality

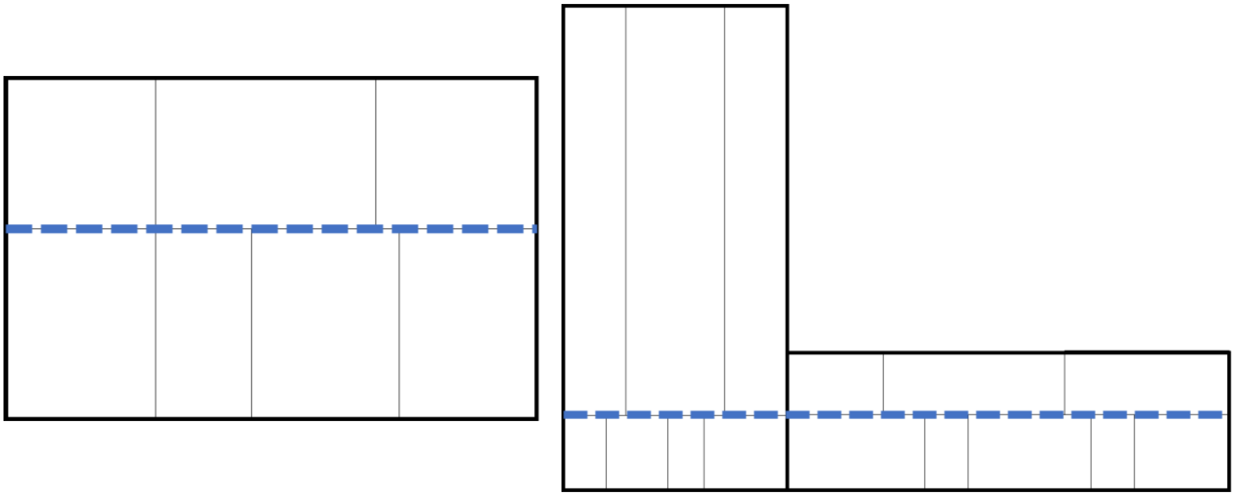
To assess the quality of the solutions obtained in the previous experiments, several comparisons took place. First, the effectiveness of the TS is analyzed in terms of the TD objective. The values of TD obtained by running the TS on a simplified version of the problem (considering only area and aspect ratio constraints) for many iterations are used as rough estimates of a lower bound. Recall that these runs were shown to converge in Section 4.4 for all problem instances.

Table 14: Best TD found for complete problem vs. best TD for simplified problem

Problem Instance	Best TD – Complete	Best TD – Simplified	Gap
10 Bed	757,530	667,503	13.49%
22 Bed	1,371,120	1,218,000	12.57%
30 Bed	2,385,060	2,170,972	9.86%
64 Bed	9,957,551	8,776,921	13.45%

The runs using the complete constraint set were very short, and yet, for all problem instances, solutions were found within 15% of the bound estimate.

For the ACC objective, the following reasoning was used to determine a reference value for the objective. If all departments in a facility are straight-line accessible from one another, ACC has a value of zero. This would require all departments to have an edge aligned along a central axis (see Figure 28 a). In facilities with multiple wings, this would result in layouts with extremely elongated departments (thus violating the aspect ratio constraints), as seen in Figure 28 b). An alternative is to have departments within each wing have a straight-line accessibility of zero with respect to each other, while setting it to a specific value for those in other wings, for example, to one for those in adjacent wings, two for those in wings adjacent to adjacent wings, and so on.



a) Rectangular facility b) Non-rectangular facility

Figure 28: Facilities with ACC = 0. All departments are aligned along the central axis indicated with a dashed line.

For the problem set used in this dissertation, the total number of departments was evenly split across wings, their individual accessibilities were calculated using the procedure described above, and summed to obtain the baseline accessibility $ACC_{baseline}$.

Table 15: $ACC_{baseline}$ calculations for problem instances

Problem	Departments	$ACC_{baseline}$
10 Bed	47	$47(23 * 0 + 23 * 1) = 1081$
22 Bed	61	$61(30 * 0 + 30 * 1) = 1830$
30 Bed	75	$75(37 * 0 + 37 * 1) = 2775$
64 Bed	115	$115(28.5 * 0 + 28.5 * 1 + 28.5 * 1 + 28.5 * 2) = 16387.5$

$ACC_{baseline}$ has a straightforward interpretation. For instance, for the 10 Bed problem, it represents the ACC value of a facility where half the departments are directly accessible from each department, while the other half is accessible by at most one turn. For the 64 Bed problem, it represents the ACC of a facility where a quarter of the departments are directly accessible, while half (a quarter for the two adjacent wings) are accessible by at most one turn, and a quarter by at most two turns (those in the furthermost wing).

For the 10 and 64 bed problem instances, solutions were found with ACC lower than the baseline (1038 vs. 1081 for the 10 bed, and 15,918 vs. 16387 for the 64 bed), while for the 22 and 30 bed they were higher (2060 vs. 1830 for the 22 bed, and 6644 vs. 2775 for the 30 bed).

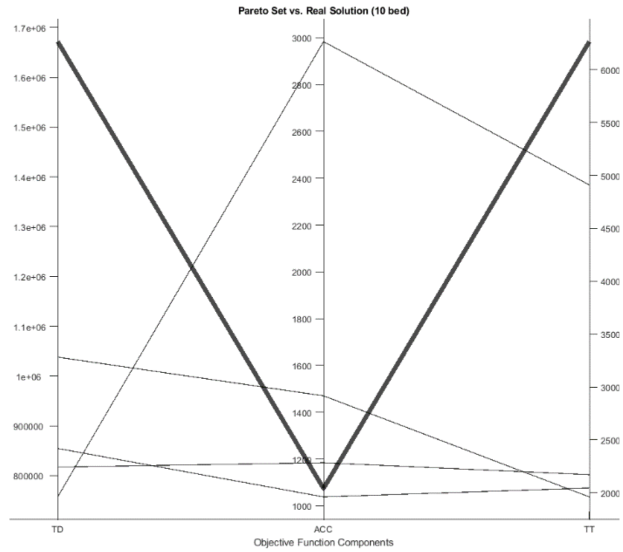
A second assessment was performed by comparing the solutions found by the TS with the block layouts based on real hospitals. The results are shown in Table 16.

Table 16: Comparison between TS solutions and real hospitals

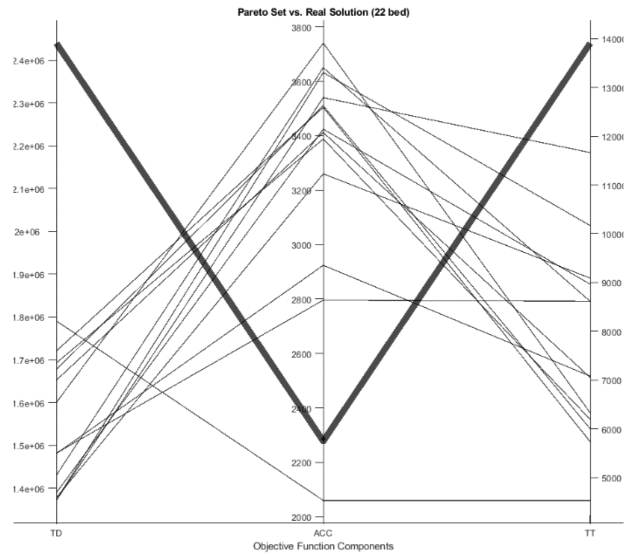
Problem	TD		ACC		TT	
	Real	TS	Real	TS	Real	TS
10 Bed	1,671,536	757,530	1074	1038	6266	1960
22 Bed	2,439,535	1,371,120	2278	2060	13906	4534
30 Bed	4,374,247	2,385,060	3592*	6644	16482	13190
64 Bed	11,602,748	9,957,551	14808*	15918	70534*	98054

The solutions obtained with the TS are better than the real hospitals except for those values marked with an asterisk in the table. This only occurs for the two largest problem instances. It is possible that allowing the TS to run for longer might find better solutions for those situations as well.

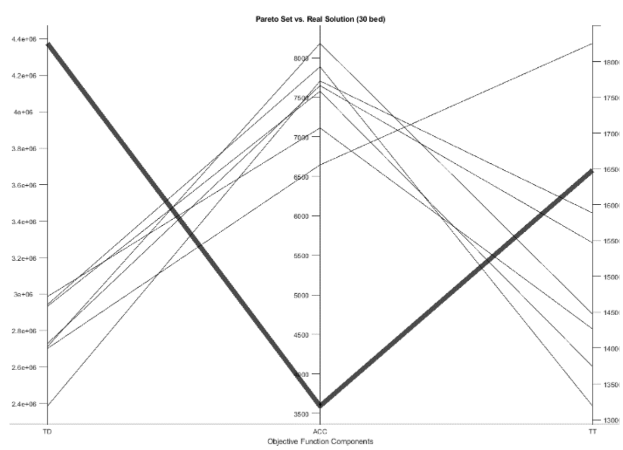
A further comparison between the real hospitals and the Pareto set for each problem instance is shown in the parallel coordinate plots of Figure 29, where the thick line represents the real hospital. Even though the real hospitals for the 10 and 22 bed instances are dominated by solutions found by the TS, we can see that they are close (especially for the 10-bed case) to the Pareto front. The real hospitals perform particularly well for the ACC component (the middle axis) but poorly for both TD (left axis) and TT (right axis), except for the 64-bed case.



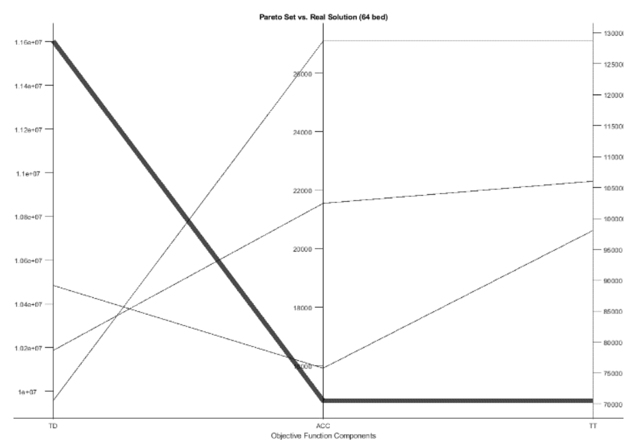
a) 10-bed problem instance



b) 22-bed problem instance



c) 30-bed problem instance



d) 64-bed problem instance

Figure 29: Parallel coordinate plots of Pareto sets vs. real solutions for all problem instances

4.6 Discussion

The Space Syntax-based block layout model proposed in this chapter represents a valuable tool for facility designers. The experiments described demonstrate how designers can use different configurations of the parameters to obtain different designs. In addition, the designs obtained show several similarities with actual designs. For instance, the patient room wards of all four problem instances follow a pattern like the one found in the actual hospitals, with patient rooms lined up in two parallel rows, with a third row in between where the nurse station is located. The inclusion of the ACC component of the objective function also plays an important role in maintaining an overall alignment of departments along their edges, which favors the construction of corridor networks, as will be seen in Chapter 5. The nesting of bays and departments is also a feature seen in actual designs, with functionally related departments clustered together to minimize travel distance between them. These results show that the model proposed captures many of the design principles that hospital designers use in actual practice.

Furthermore, both the TS algorithm used to generate block layouts, as well as the model itself, are highly configurable, so that practitioners can experiment with different values of the parameters to obtain different designs with different characteristics. This, combined with the insights gained from the experiments described above about the effects of varying the parameters, provide a valuable support for the design task. The computational experience showed that the proposed approach can be used to solve real-life problems in a reasonable amount of time, and the comparison with real hospitals shows that practitioners can use it to find designs that perform better in most respects than those already existing.

V. The Corridor Network Model

This chapter introduces the corridor network generation model. Movement inside a physical rehabilitation hospital takes place, for the most part, along corridors. Designing good corridor networks is important since most patients are limited in their mobility and need to be moved around in wheelchairs. In consequence, congestion along the corridors becomes an issue. Furthermore, easily navigable corridors allow staff members to reach patients more rapidly, thus enabling them to spend more time at patient bedside.

The corridor model is used to construct an aisle network on an already existing block layout. Once the network is built, the metrics found for the block layout (total travel distance, total accessibility depth, total trips weighted by turns required) are re-calculated using the aisle structure. This chapter introduces the corridor network model, and investigates the relationship between the block layout's metrics, and the metrics based on the aisle network. First, the notation used in the model is introduced, followed by a description of how the different sets and parameter values are determined. The model is then presented, and a series of experiments are described with their results.

5.1 Notation

Parameters

$\mathcal{N}$ = set of candidate nodes, indexed over $1,2,\ldots,\|\mathcal{N}\|$.

$\mathcal{D}$ = set of department nodes, indexed over $1,2,\ldots,\|\mathcal{D}\|$.

$\overline{\mathcal{D}}$ = set of non-department nodes, indexed over $1,2,\ldots,\|\mathcal{N}\|-\|\mathcal{D}\|$.

$\mathcal{A}$ = set of candidate arcs, indexed over $1,2,\ldots,\|\mathcal{A}\|$. $\mathcal{A}=\mathcal{A}_C\cup\overline{\mathcal{A}_C}$.

$\mathcal{A}_C$ = set of candidate arcs in or out of a department centroid.

$\overline{\mathcal{A}_C}$ = set of candidate arcs that do not include a department centroid.

$\mathcal{I}_2$ = set of candidate 2-way intersections, indexed over $1,2,\ldots,\|T_2\|$.

$\mathcal{I}_3$ = set of candidate 3-way intersections, indexed over $1,2,\ldots,\|T_3\|$.

$\mathcal{I}_4$ = set of candidate 4-way intersections, indexed over $1,2,\ldots,\|T_4\|$.

S_{ij} = supply/demand matrix, where each element s_{ij} is the demand at node j for trips originating at node i. All non-centroid nodes have demand equal to zero.

d_{ij} = distance between candidate nodes i and j.

io_i^{max} = max I/O points per department.

α_C = trade-off between turns and total flow. The subscript is to differentiate it from the parameter used for the block layout.

β_C = trade-off between turns and max traffic on a turn.

γ = trade-off between an intersection and a corridor-ending node.

Decision Variables

$a_{ij} = \{0,1\}$ whether arc (i,j) is in network.

$x_{ijk}^b = \{0,1\}$ whether the 2-way intersection defined by arcs (i,j) and (i,k) is present in the network. Its index in $\mathcal{I}_2$ is b.

$y_{ijkl}^b = \{0,1\}$ whether the 3-way intersection defined by arcs (i,j), (i,k), and (i,l) is present in the network.

$z_{ijklm}^b = \{0,1\}$ whether the 4-way intersection defined by arcs (i,j), (i,k), (i,l), and (i,m) is present in the network.

$f_{a_{ij}\in\mathcal{A}}^k$ = flow originating in department k moving along arc a_{ij}.

$aux_{i\in\overline{\mathcal{D}}}$ = auxiliary variables for counting corridor terminating nodes.

t_i = traffic on intersection node i.

5.2 Construction of required sets and parameters

The model requires a collection of candidate arcs, candidate nodes, and potential intersections. These are obtained using the following procedure, which will be exemplified using the block layout found in Figure 30:

(1) For each department, add candidate I/O nodes along each edge in the corner and center positions.

(2) Remove overlapping nodes (such as those along the edges of two adjacent departments).

(3) Connect all nodes along the edge of a department to its centroid. Every node along a department's edges is considered a potential I/O point.

(4) Connect all non-centroid nodes to their nearest neighbor. The resulting network is the initial candidate network. The initial candidate network for the example is shown in Figure 31.

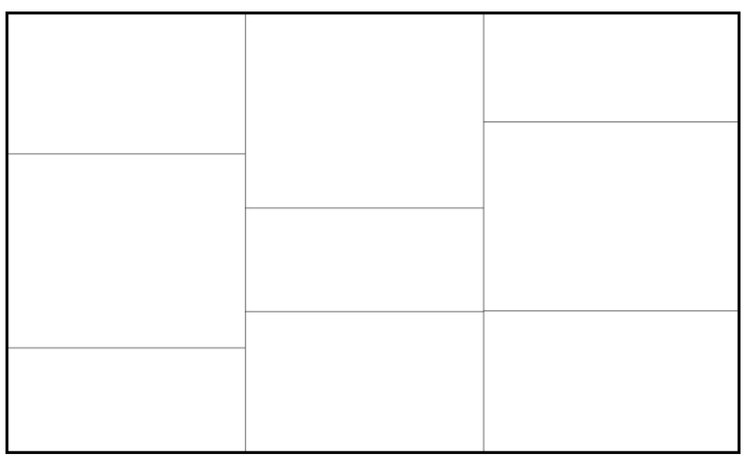


Figure 30: Given block layout

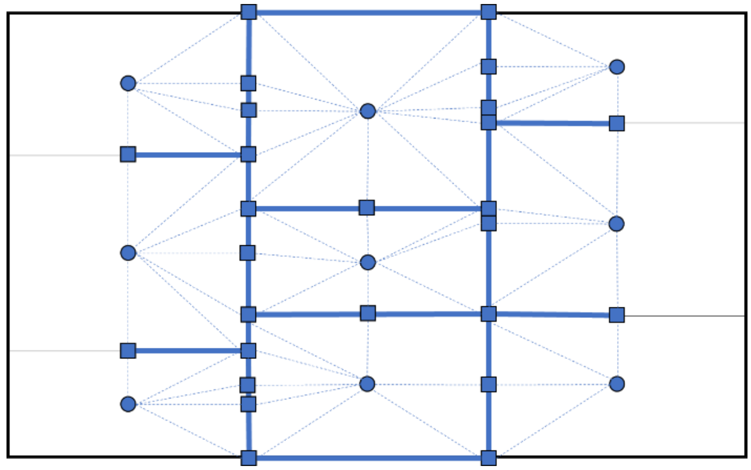


Figure 31: Initial candidate network

Centroid nodes are shown as circles and candidate I/O points as squares. The dotted lines represent connections between centroids and I/O points, while solid lines represent connections between candidate I/O points. All nodes form the set $\mathcal{N}$ of candidate nodes, which is composed of two sets of nodes, the set of department or centroid nodes $\mathcal{D}$ and its complement, the set of non-centroid nodes. All the arcs connecting nodes form set $\mathcal{A}$ with the subset of centroid arcs being those that include a centroid. The set of 2-way intersection candidates are found by checking all non-centroid nodes and their adjacent arcs. The following figure shows all the nodes that are on a potential 2-way or greater intersection as open squares. Note that some of these nodes are also on potential 3-way and 4-way intersections. Since it is assumed that corridors can only intersect at 90-degree angles, the maximum number of corridor fragments intersecting at a node is equal to four.

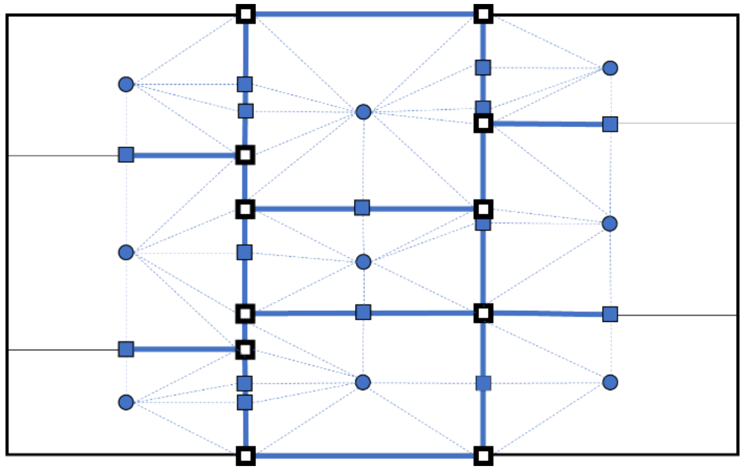


Figure 32: Nodes that are on potential 2-way or greater intersections shown as open squares

The set of 3-way intersection candidates is found by finding all those nodes that are on at least three candidate arcs.

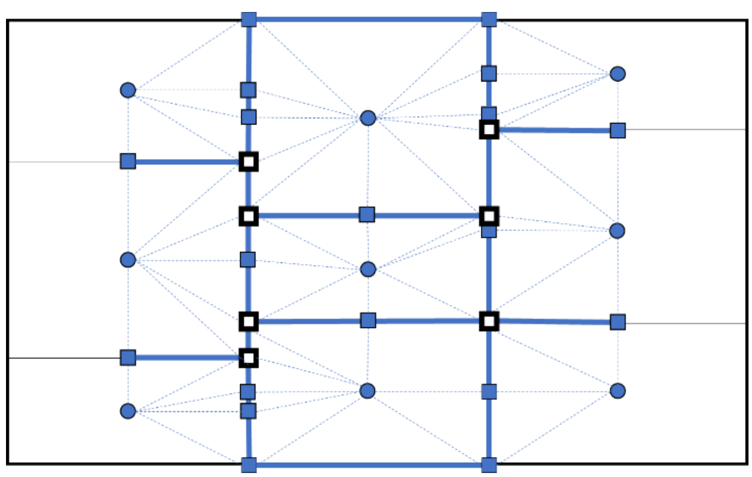


Figure 33: Nodes that are on potential 3-way or greater intersections shown as open squares

Lastly, the set of 4-way intersections is constructed by finding all the nodes that are on four candidate arcs.

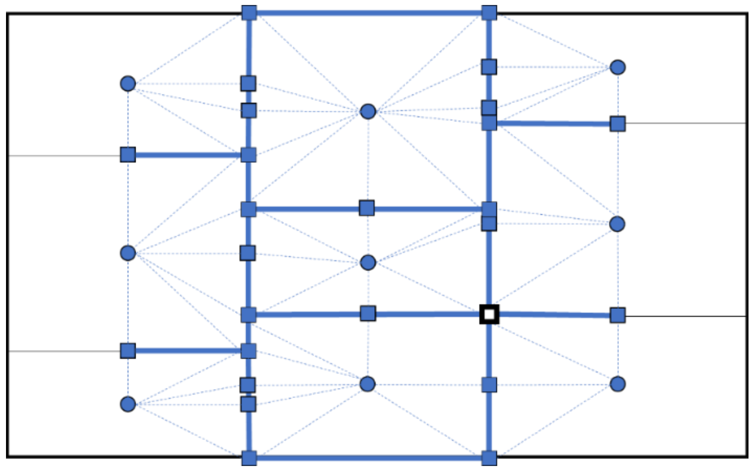


Figure 34: Nodes that are on potential 4-way intersections shown as open squares

Once these sets are defined, the MIP model can generate a final aisle network, such as the following:

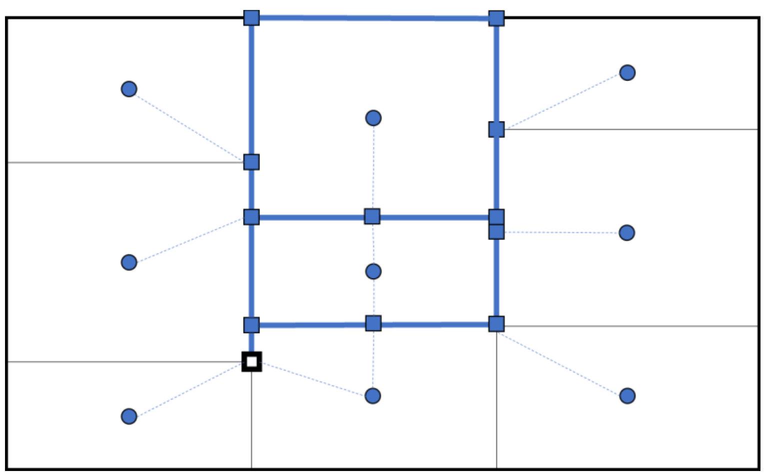


Figure 35: Final aisle network as found by the MIP model

A corridor terminating node (CTN) is a node that is connected to only one other non-department node, such as the one highlighted by an open square in Figure 35. The designer's preference for these kinds of nodes is controlled by the trade-off parameter γ, as a terminating node might be considered undesirable in certain circumstances. The parameter's value is provided by the facility designer.

5.3 MIP Corridor Generation Model

The corridor generation model was formulated as a MIP and solved using CPLEX 12.8. This section describes the model in detail.

5.3.1 Objective Function

The objective function is analogous to the one used for the block layout. It is a single objective function with three components: (1) the number of turns plus corridor-terminating nodes, (2) the total flow on the network, and (3) the maximum flow on a turn node. Its functional form is:

$$Min\,Z = \left(TI + \frac{1}{\gamma}TCTN\right) + \frac{1}{\alpha_C}TF + \frac{1}{\beta_C}MFI \tag{5.1}$$

Each one of these components is calculated as follows:

$$TI = \sum_{b=1}^{\|\mathcal{I}_2\|} x_{ijk}^{b} - \sum_{b=1}^{\|\mathcal{I}_3\|} y_{ijkl}^{b} + \sum_{b=1}^{\|\mathcal{I}_4\|} z_{ijklm}^{b} \tag{5.2}$$

$$TCTN = \sum_{i=1}^{NN-ND} aux_i \tag{5.3}$$

$$TF = \sum_{i=1}^{\|\mathcal{D}\|} \sum_{a\in\mathcal{A}} f_a^i\, d_{ij} \tag{5.4}$$

$$MFI = \max t_a\,, \forall a \in T_2 \tag{5.5}$$

Both parameters α_C and γ are provided by the facility designer, with their specific values depending on the problem instance. After analyzing the model, it was determined that the parameter β_C should not be configurable by the facility designer. If MFI is given too much influence on the objective function, the model will produce unreasonable designs. There are two ways in which MFI can improve, either by altering the structure of the network (by changing the first component of the objective function) or by changing TF. The latter is achieved be re-routing flow. If the trade-off between TF and MFI is too small, flow will be routed in inefficient ways simply to avoid turn nodes when possible. The only significant cause for change in MFI should be changing the network's structure. To enforce this, the parameter is set to $10 \times \alpha_C$, without designer input.

5.3.2 Constraints

The objective function described above is subject to the following constraints:

Flow preservation on the network:

$$\sum_{a_{jk} \in \mathcal{A}} f^i_{jk} - \sum_{a_{kj} \in \mathcal{A}} f^i_{kj} = s_{ij}, \quad \forall i \in \mathcal{D}, \forall j \in \mathcal{N} \tag{5.6}$$

No flow allowed unless arc is in the network:

$$\sum_{i \in \mathcal{D}} f^i_{a_{jk}} = 0, \quad if\ a_{jk} = 0 \quad \forall a_{jk} \in \mathcal{A} \tag{5.7}$$

If no flow on arc, remove arc from the network:

$$a_{jk} = 0, \quad if \sum_{i \in \mathcal{D}} f^i_{a_{jk}} = 0, \quad \forall a_{jk} \in \mathcal{A} \tag{5.8}$$

Bi-directionality of arcs:

$$a_{ij} = a_{ji} \quad \forall a_{ij} \in \mathcal{A} \tag{5.9}$$

Limit on number of I/O points per department:

$$\sum_{a_{ij} \in \mathcal{A}_C} a_{ij} \leq io^{max}_i, \quad \forall i \in \mathcal{D} \tag{5.10}$$

Ensure that departments are not used as intermediate nodes:

$$\sum_{a_{jk} \in \mathcal{A}_C} f^i_{a_{jk}} + \sum_{a_{kj} \in \mathcal{A}_C} f^i_{a_{kj}} = -s_{ik} \quad \forall i \in \mathcal{D} \tag{5.11}$$

2-way intersection linearization:

$$\begin{aligned} x^b_{ijk} &\leq a_{ij} && \forall b \in \mathcal{I}_2 \\ x^b_{ijk} &\leq a_{ik} && \forall b \in \mathcal{I}_2 \\ x^b_{ijk} + 1 &\geq a_{ij} + a_{ik} && \forall b \in \mathcal{I}_2 \end{aligned} \tag{5.12}$$

3-way intersection linearization:

$$\begin{aligned} y_{ijkl}^{b} &\leq a_{ij} && \forall b \in \mathcal{I}_3 \\ y_{ijkl}^{b} &\leq a_{ik} && \forall b \in \mathcal{I}_3 \\ y_{ijkl}^{b} &\leq a_{il} && \forall b \in \mathcal{I}_3 \\ y_{ijkl}^{b} + 2 &\geq a_{ij} + a_{ik} + a_{il} && \forall b \in \mathcal{I}_3 \end{aligned} \tag{5.13}$$

4-way intersection linearization:

$$\begin{aligned} z_{ijklm}^{b} &\leq a_{ij} && \forall b \in \mathcal{I}_4 \\ z_{ijklm}^{b} &\leq a_{ik} && \forall b \in \mathcal{I}_4 \\ z_{ijklm}^{b} &\leq a_{il} && \forall b \in \mathcal{I}_4 \\ z_{ijklm}^{b} &\leq a_{im} && \forall b \in \mathcal{I}_4 \\ z_{ijklm}^{b} + 3 &\geq a_{ij} + a_{ik} + a_{il} + a_{im} && \forall b \in \mathcal{I}_4 \end{aligned} \tag{5.14}$$

Corridor-ending node auxiliary variable:

$$aux_i = \begin{cases} 0, & \sum_{j \in \overline{\mathcal{D}}} a_{ij} \neq 1 \quad \forall i \\ 1, & \sum_{j \in \overline{\mathcal{D}}} a_{ij} = 1 \quad \forall i \end{cases} \tag{5.15}$$

Flow on intersection nodes:

$$t_i = \begin{cases} 0, & x_{ijk}^{b} = 0 \quad \forall b \in \mathcal{I}_2 \\ \sum_{n=1}^{\|\mathcal{D}\|} \sum_{a_{li} \in \mathcal{A}} f_{a_{li}}^{n}, & x_{ijk}^{b} = 1 \quad \forall b \in \mathcal{I}_2 \end{cases} \tag{5.16}$$

5.4 Results

Several tests were run using the model. First, a series of sensitivity analyses were performed on the 10-bed hospital to determine the effects and best values of the parameters. Second, corridor networks were constructed on the block layouts obtained in experiments 1 and 2 from Chapter 4. The correlation between the block layout model's metrics and those calculated on the corridor network is calculated to assess how the block layouts metrics function as proxies of the network metrics. Lastly, some tests were run on the larger problem instances to verify the scalability of the proposed approach.

For all experiments, the corridor model was run until the optimality gap was less than 2.5% or 10 minutes had passed. Running for longer had no significant effect on the quality of the solutions. Proving optimality often takes several hours (even for the 10-bed problem) and was not proved in the larger instances. However, good feasible solutions are found quickly.

5.4.1 Effects of parameters α_C and γ

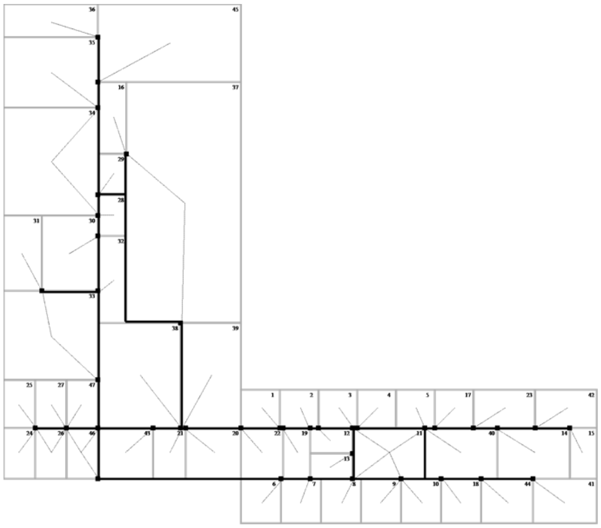
Parameter α_C controls the trade-off between number of turns on the network and the total travel distance. Parameter γ controls the desirability of corridor-ending nodes versus loops. The first experiment using the corridor generation model sought to investigate the designs resulting from different values of the parameters. The values used for α_C ranged from 100 to 10,000, while γ was set to 1, 3, and 5. The block layout used was the 10-bed real hospital shown in Figure 21 a).

A summary of results is shown in Table 17.

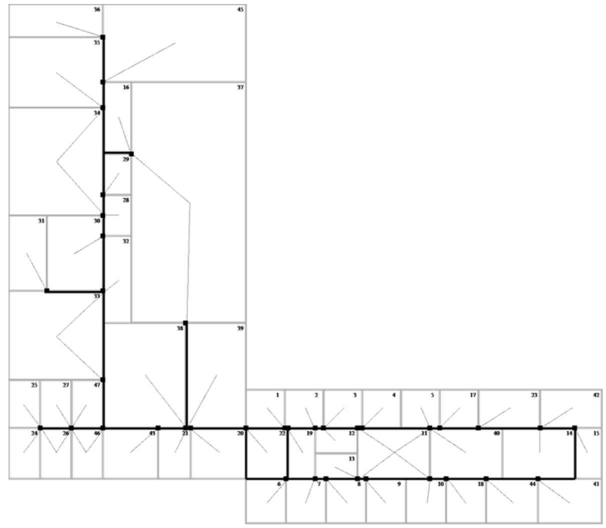
Table 17: Summary of results of varying α_C, γ. TI = total number of intersections, TCTN = total corridor terminating nodes.

α_C	γ	TI	TCTN
100	1	12	6
500		19	3
1000		13	4
10000		10	5
100	3	22	2
500		23	2
1000		18	1
10000		13	2
100	5	21	3
500		26	1
1000		21	1
10000		17	1

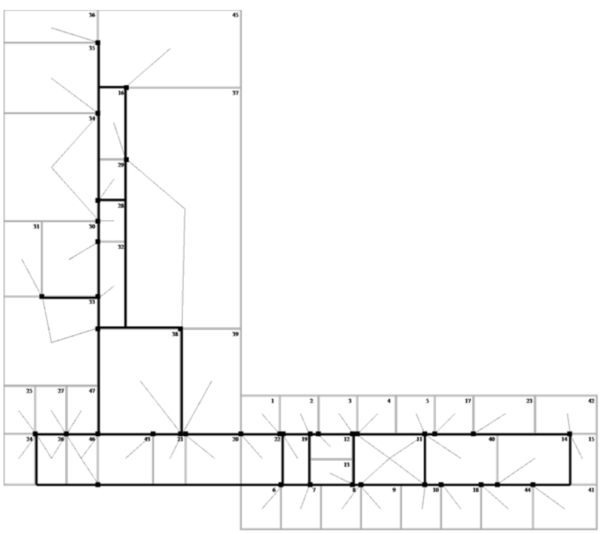
It is clear from these results that lower values of α_C yield, in general, more intersections on the network. This makes sense, since more intersections generally imply more possible routes between departments, which in turn imply shorter travel distances. Decreasing γ results in more corridor terminating nodes, as it makes them equivalent to intersections. The resulting networks for the extreme values of α_C and for all three values of γ are seen in Figure 36.



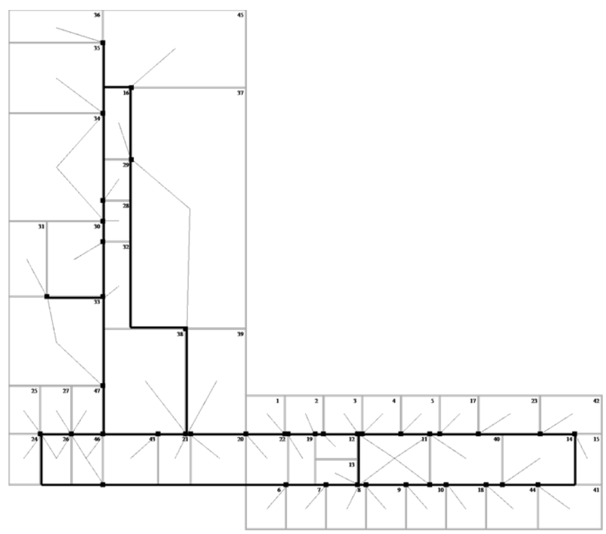
a) $\alpha_C = 100, \gamma = 1$
TF = 1,726,204, TI = 12, TCTN = 6, MFI = 5761



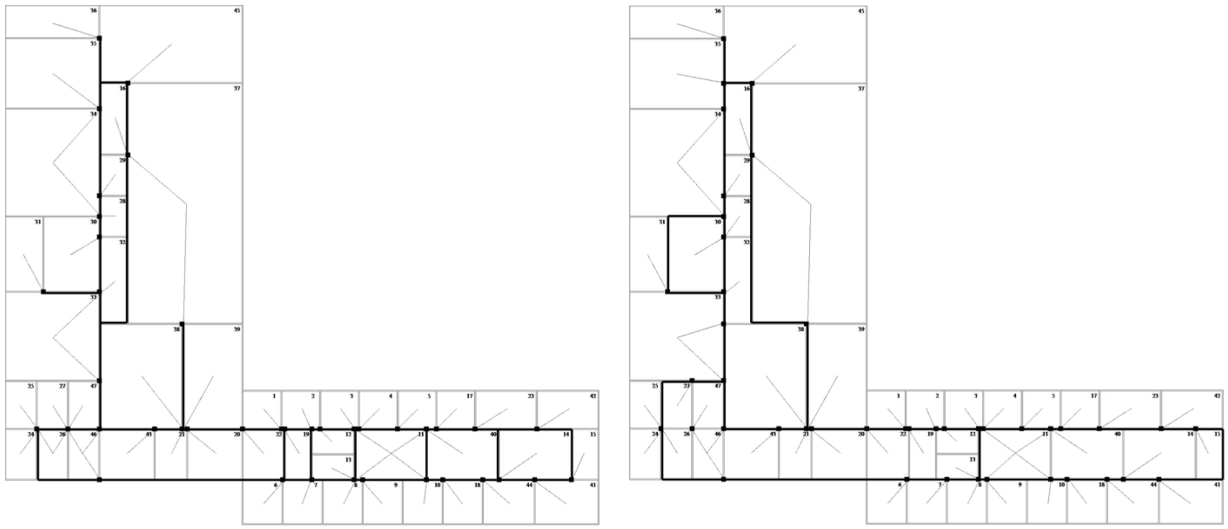
b) $\alpha_C = 10000, \gamma = 1$
TF = 1,696,170, TI = 10, TCTN = 5, MFI = 6822



c) $\alpha_C = 100, \gamma = 3$
TF = 1,705,479, TI = 22, TCTN = 2, MFI = 6822



d) $\alpha_C = 10000, \gamma = 3$
TF = 1,699,450, TI = 13, TCTN = 2, MFI = 6822



e) $\alpha_C = 100, \gamma = 5$
TF = 1,689,808, TI = 21, TCTN = 3, MFI = 6790

f) $\alpha_C = 10000, \gamma = 5$
TF = 1,700,309, TI = 17, TCTN = 1, MFI = 6740

Figure 36: Corridor networks for varying values of α_C, γ

The main way in which travel distance is reduced is by adding cross-aisles. For instance, compare the corridors at the bottom of Figure 36 e) and f). In f) there is only one cross-aisle, while in e) four additional cross-aisles have been added. Similar patterns are observed for the other layouts shown in the figure.

The effect of changing γ becomes even more evident if we see the corridor network resulting from setting it equal to 0.5 (Figure 37). The low value of the parameter eliminates all loops from the network. Additional insight can be gained by looking at the impact this change has on the distribution of flow through the network (Figure 38). The absence of loops in a) means that there are no alternative routes from one department to another, so that all traffic must go through the same corridors. The loops found in b) distribute the traffic, resulting in less congestion. The maximum traffic on a turn node for $\gamma = 0.5$ is only 2% higher than the one with $\gamma = 5$ (going from 6740 to 6842, which means 102 additional trips on that node), but the average traffic per corridor is 45% larger (increasing from 8637 to 15576). The absence of loops also has an impact on the total flow, which increases to 1,719,081 for the network with $\gamma = 0.5$, from 1,700,309 for the one with $\gamma = 5$.

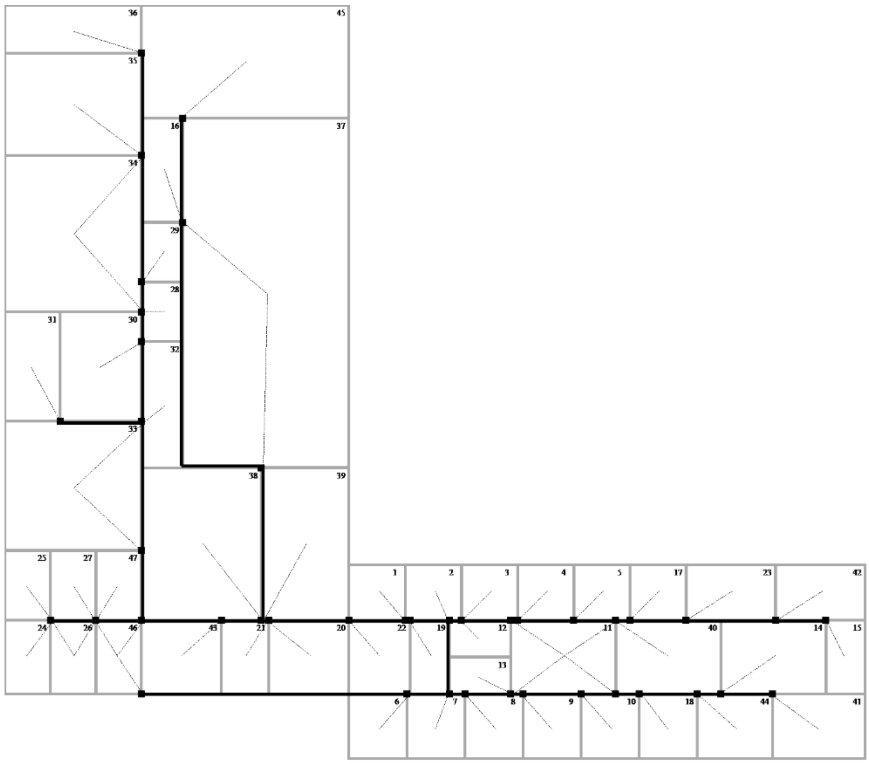
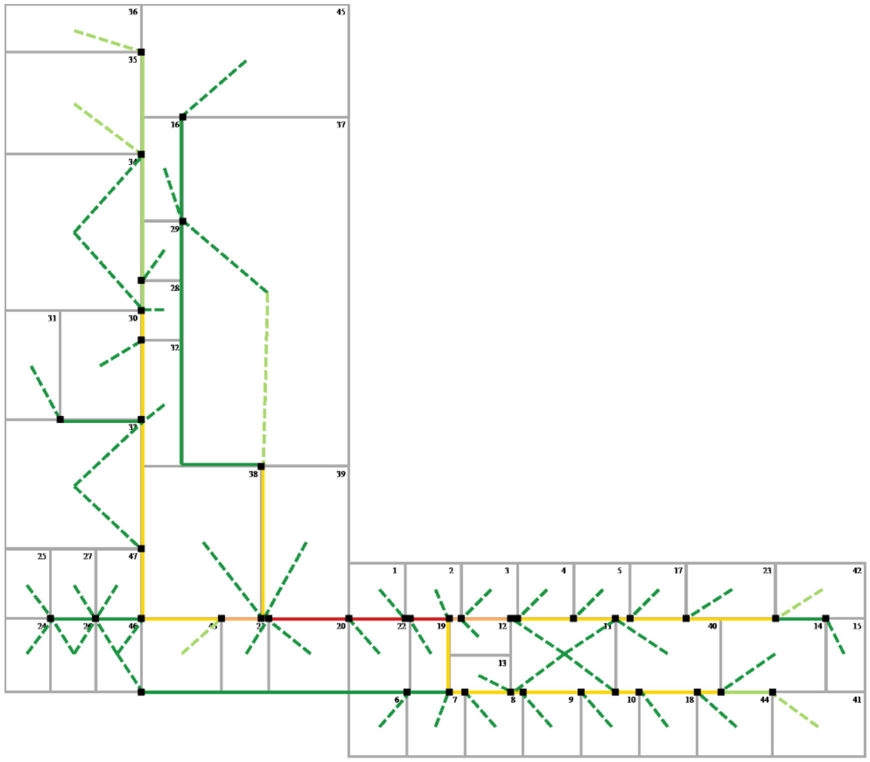
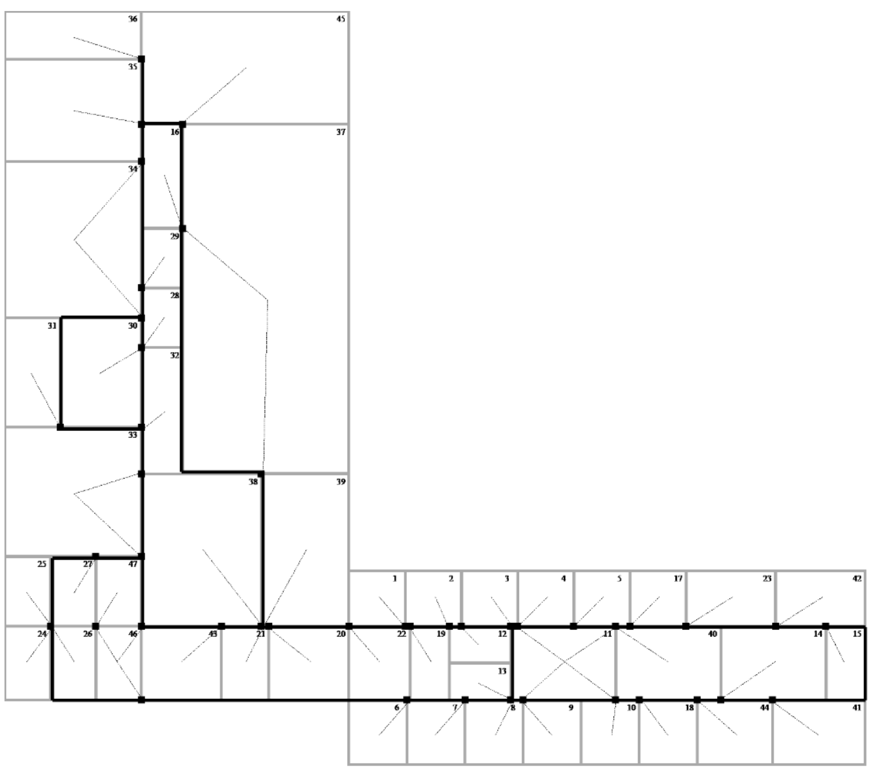
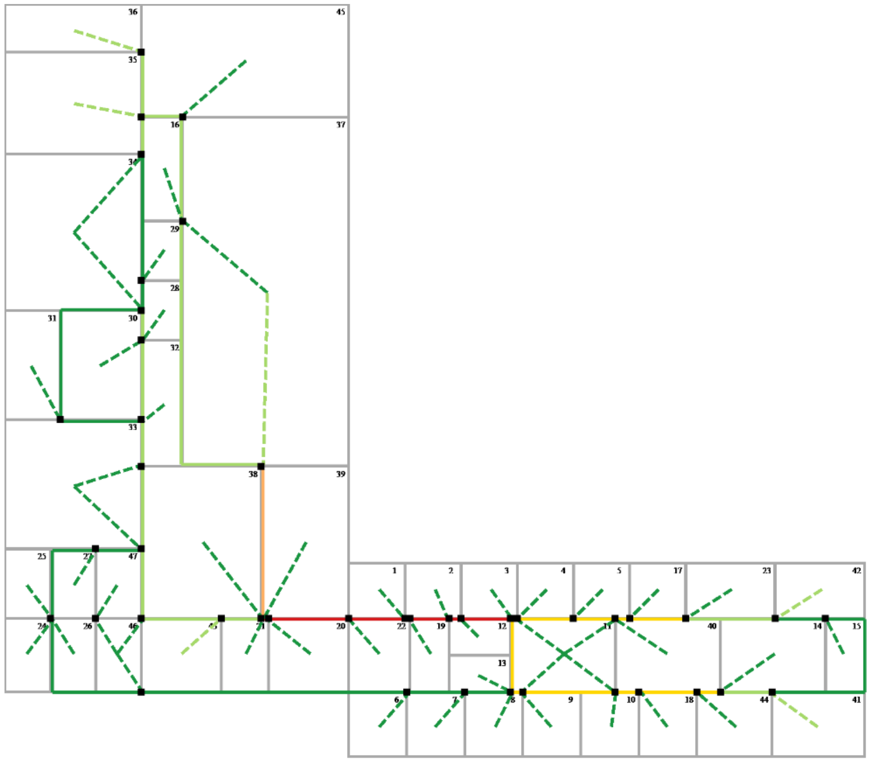


Figure 37: Corridor network for $\alpha_C = 10000, \gamma = 0.5$
TF = 1,719,081, TI = 7, TCTN = 7, MFI = 6842



a) Flow distribution for $\alpha_C = 10000, \gamma = 0.5$





b) Corridor network and flow distribution for $\alpha_C = 10000, \gamma = 5$
TF = 1,700,309, TI = 17, TCTN = 1, MFI = 6740

Figure 38: Flow distribution on networks with different values of γ. Red fragments are in the 80th traffic percentile, orange in the 60th, yellow in the 40th, and light green in the 20th.

This experiment demonstrates that using the proposed model, a facility designer can adjust the parameters α_C and γ to control not only the structure of the corridor network, but the distribution of traffic as well.

5.4.2 Relationship between block layout metrics and corridor network metrics

The block layout model described in Chapter 4 uses the rectilinear distance metric to calculate distances, where the distance is measured from department centroid to department centroid. The corridor distance metric is more realistic. However, generating corridor networks for each candidate layout generated by the TS algorithm is not practical, as it can take several minutes to construct these networks, even on the smallest problem instance. Likewise, the straight-line access metric measured on the block layout is based on the department edges, assuming they represent potential corridor fragments. Once the corridor structure is in place, however, the straight-line access metric needs to be updated. Instead of using department edges to measure it, the actual corridors (including the I/O points for each department) are used instead. It is important to evaluate whether the metrics measured in the block layout phase are good proxies of those measured on the corridor network. Otherwise, there is no assurance that a good block layout will remain good once the corridor network is added.

To test this, corridor networks were constructed on the 130 block layouts obtained in experiments 1 and 2 from Chapter 4. The following metrics were computed on these corridor networks: the three objective function components (TF, TI+γTCTN, and MFI), as well as straight-line access using corridors (ACC_C) and the total traffic on intersection nodes (TTI). The results for the block layouts from experiment 1 are shown in Table 18.

Table 18: Corridor network for block layouts from experiment 1. α_C and γ are the corridor generation parameters, whereas α_B and β_B are the block layout model parameters. ACC_C is straight-line access measured on the corridor network, TTI is the total traffic on intersection nodes, and the remaining columns are the components of the objective function.

α_C	α_B level	β_B level	TI+γTCTN	TF	ACC_C	MFI	TTI
1	L	L	41.6	1,046,841	3764	4828	17388
	L	M	35.6	1,149,869	3927	4546	17540
	L	H	38.1	1,091,344	3839	4657	16660
	M	L	44.4	1,024,692	4136	4274	18313
	M	M	38.7	1,057,468	3915	4978	17194
	M	H	37.7	1,126,513	3968	4936	17399
	H	L	44.7	1,077,186	4302	5160	19690
	H	M	42.5	1,006,685	4175	4870	18482
	H	H	49.0	996,751	4204	4655	18343
1000	L	L	26.2	1,048,401	3628	4784	17377
	L	M	24.9	1,140,682	3877	4273	17099
	L	H	24.0	1,090,629	3798	4230	16703
	M	L	25.5	1,027,447	3899	3983	18107
	M	M	24.3	1,055,807	3935	4521	17361

	M	H	23.6	1,129,417	3971	4815	17447
	H	L	31.1	1,071,563	4195	4712	19602
	H	M	30.3	1,008,877	4156	4341	18674
	H	H	33.1	999,589	4165	4289	18355
7000	L	L	17.8	1,065,715	3735	4910	17677
	L	M	17.5	1,163,346	3956	4927	17539
	L	H	17.3	1,102,658	3859	5226	17037
	M	L	18.6	1,042,703	3898	4729	18163
	M	M	17.5	1,074,279	3962	5138	17461
	M	H	17.7	1,134,510	4009	4835	17773
	H	L	24.6	1,084,960	4413	4942	20264
	H	M	24.5	1,019,687	4213	4928	18822
	H	H	26.4	1,013,077	4261	4343	18641
100000	L	L	12.2	1,326,595	3889	4619	18919
	L	M	14.1	1,386,587	4131	5092	19056
	L	H	11.0	1,333,504	3942	4296	18280
	M	L	10.5	1,332,437	4145	4198	20318
	M	M	11.5	1,343,566	4066	4130	18760
	M	H	16.0	1,350,965	4204	4510	18743
	H	L	18.2	1,391,413	4676	4717	22303
	H	M	18.9	1,328,096	4434	4995	20345
	H	H	19.8	1,319,289	4571	4680	20207

For TF, layouts obtained when both α_B and β_B were at a high level are the best. For ACC_C layouts obtained when both α_B and β_B were at their low levels are consistently the best. For MFI, a medium level of both α_B and β_B is likely to be best. In terms of the complexity of the network, that is, the number of intersections plus corridor-terminating nodes, it is invariably associated with the high level of α_B. These results are mostly consistent with the equivalent results for the block layouts. In terms of the correlation between block layout and corridor-based metrics, the results are shown in Table 19. Each column title indicates which metrics are being compared.

Table 19: Correlation between block and corridor metrics for 10-bed

α_C	TD/TF	ACC_B/ACC_C	TT/MFI	TT/TTI
1	0.8875	0.6345	0.0274	0.7379
1000	0.8920	0.7099	0.0426	0.7653
7000	0.8714	0.6856	-0.0097	0.7322
100000	0.5932	0.6529	0.3041	0.5118

The correlations between the rectilinear travel distance and the corridor total flow are high (TD/TF column), which means that the rectilinear distance is a good proxy. The reason for this is that the straight-line access constraints, the accessibility depth minimization in the block layout model, as well as the bay structure used to encode the layouts, all force departments to be aligned along their edges. Such alignment naturally lends itself to reasonable corridor structures. However, as the value of α_C increases, the correlation decreases. This makes sense since higher values of α_C mean that the travel distance is less influential on the objective function as the corridor network is being constructed.

The correlation between ACC_B and ACC_C is moderately high. MFI, on the other hand, is not correlated to the TT metric measured on the block layout. However, the TTI metric has a moderate correlation to the one for the block layouts.

For the 22-bed problem instance, a summary of the corridor-based metrics when $\alpha_C = 1000$ is shown in Table 20.

Table 20: Corridor network on block layouts from experiment 2

α_B level	β_B level	TI+$\gamma TCTN$	TF	ACC_C	MFI	TTI
L	L	17.13	1,691,666	6120	4952	31974
L	H	19.00	1,805,977	6322	5902	30847
H	L	24.50	1,598,028	6493	6097	33081
H	H	24.70	1,587,210	6580	5454	32833

Like with the 10-bed problem instance, corridor network complexity increases with the greater importance of the TF component. ACC_C is at its best with the low level of both α_B and β_B, while TTI profits from α_B at the low level and β_B at its high level. With respect to correlations, that corresponding to TD/TF is very high (0.9097), for ACC_B/ACC_C is moderate (0.6761), for TT/TTI high (0.8502), while for MFI/TTI, once more, low (0.2452).

5.4.3 Scalability of the model

To test the scalability of the model, corridor networks were constructed on the best block layouts for the 30 and 64 bed instances. The optimizer was stopped if the runtime reached one hour or if the optimality gap was lower than 5% for the 30-bed, and 10% for the 64-bed. A representative example of the outcome of this test is shown in Table 21 and Table 22, for the 30 and 64 bed problem instances, respectively.

Table 21: Results for the 30-bed problem instance ($\alpha_C = 1000$)

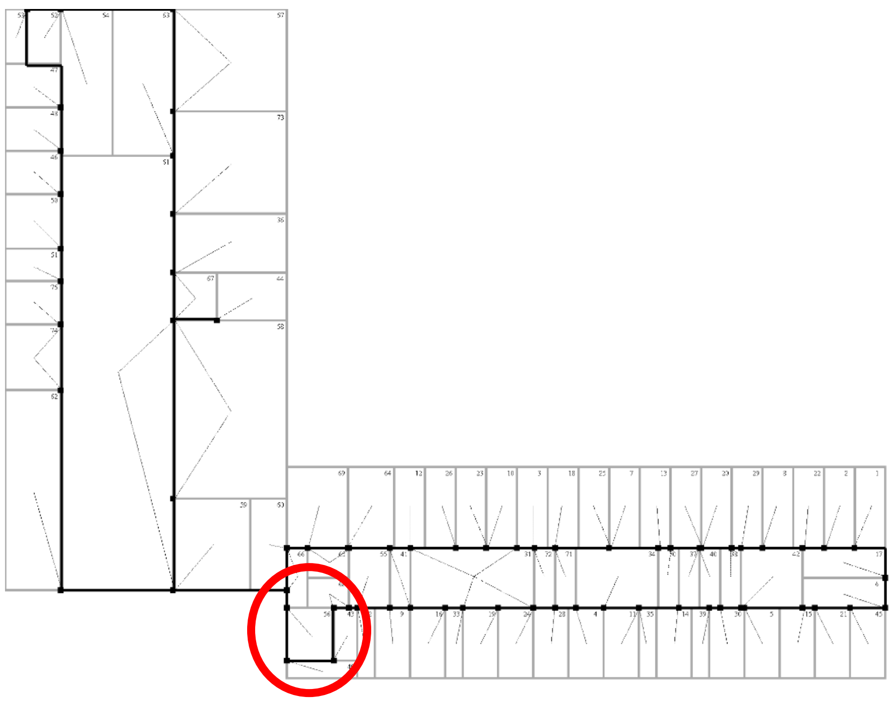
Best block layout with respect to…	TI+γTCTN	TF	ACC_C	MFI	TTI
ACC	38	2,902,997	10098	8366	45550
TT	50	2,968,493	10156	9976	41790
TD	44	2,503,147	10090	6428	44204

Table 22: Results for the 64-bed problem instance ($\alpha_C = 1000$)

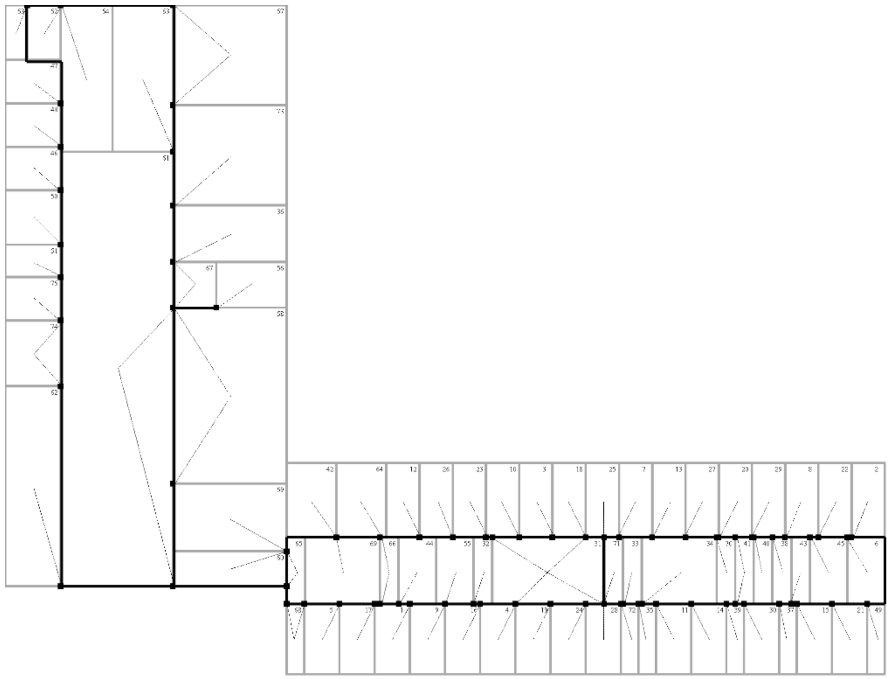
Best block layout with respect to…	TI+γTCTN	TF	ACC_C	MFI	TTI
ACC	53	12,354,537	25660	23268	153,782
TT	53	12,354,537	25660	23268	153,782
TD	79	12,207,198	31032	25137	156,474

As can be seen in the tables, those layouts with the best values under the block metrics tend to have better values with the corresponding corridor-based metrics. The correlation between block and corridor-based metrics also lend support to the notion that block layouts that perform well with respect to some objective also do so when corridors are added. The correlation between TD and TF is 0.8239 and 0.7973 for the 30 and 64 bed problem instances, respectively. It is 0.6753 and 0.8308 for ACC_B and ACC_C, and 0.8696 and 0.8329 for TT and TTI. As with the smaller instances, TT and MFI are not strongly correlated.

An exception occurs under extreme values of α_C. For example, when $\alpha_C = 100000$, so that the number of turns is minimized without considering travel distance, the block layout with the best TD has a worse corridor-based TF than the one that had the worst block TD. The reason is simple. The nested departments at the bottom of the layout (circled in Figure 39 a) require a corridor to wrap around them, which then causes all departments on the bottom to have to travel around them since adding a cross-aisle (which would alleviate this situation) would imply an additional intersection.



a) Block layout with best TD but worse TF (TD = 2,674,566, TF = 3,383,640)

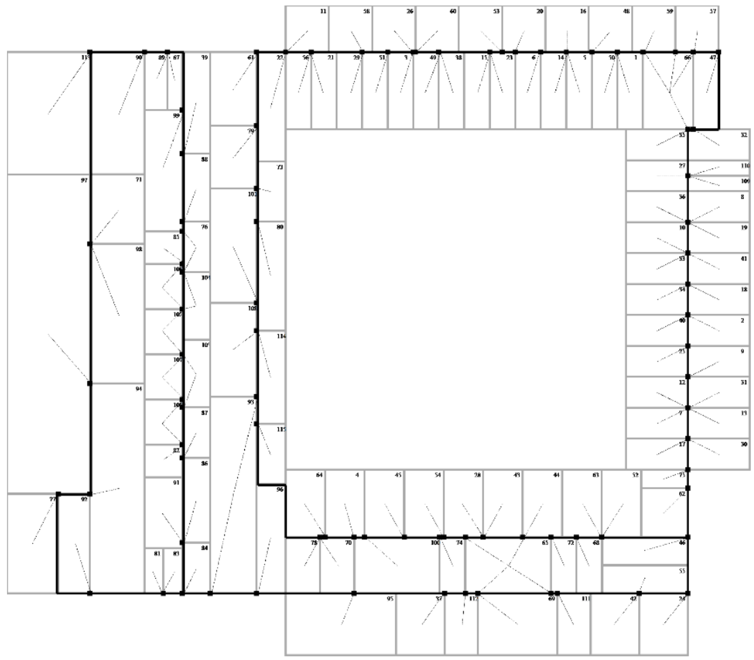


b) Block layout with worse TD and better TF (TD = 2,861,255, TF = 3,249,125)

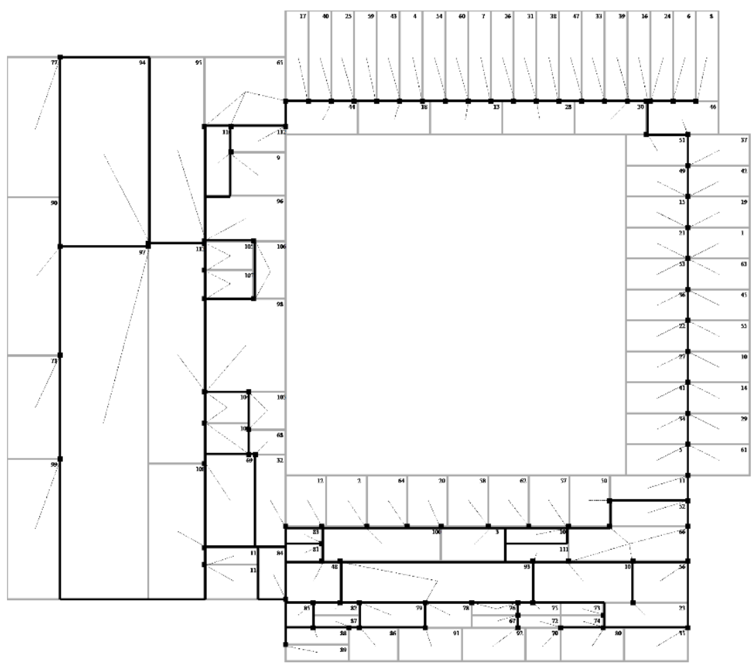
Figure 39: 30-bed layout with corridors built with $\alpha_C = 100000$

This example provides a useful rule of thumb: if the ultimate design goal is to minimize intersections on the network without concern for travel distance, then minimizing accessibility in the block layout can yield a network with better corridor-based TF than one for a block layout optimized for rectilinear TD.

Some of the corridor networks obtained for the 64-bed instance are shown in Figure 40.



a) Intersection minimizing corridor design ($\alpha_C = 100000$) on block layout with low ACC
TF = 15,133,291, I = 15, TCTN = 0, MFI = 28087, ACC_C = 29562



b) TF minimizing corridor design ($\alpha_C = 1000$) on block layout with low TD
TF = 12,207,198, I = 64, TCTN = 5, MFI = 25137, ACC_C = 31032

Figure 40: 64-bed instance corridor networks

5.4.4 Computational Experience

The model was run in CPLEX 12.8, on the same computers as the TS. For the 10-bed problem instance, runs were stopped at 10 minutes or if the optimality gap dropped below 2.5%. In those cases where the run timed out, the optimality gap was within 5%. The same was true for the 22-bed problem instance.

For the 30 and 64-bed instances, the runs were stopped after 1 hour or if the optimality gap was within 2.5%. For both instances, all runs that timed out had an optimality gap below or very near 10% (the highest recorded being 11.84%).

Like with the TS described above, these runtimes are reasonable for the type of problem being solved and show that the proposed model can be used in real-sized problems.

5.4.5 Discussion

Like the block layout model presented in the previous chapter, the corridor generation model, with its configurable parameters and reasonable runtimes, offers a useful tool for facility designers. The tunable parameters allow control of the structure of the corridor network and, with it, the total travel distance, the distribution of traffic, and the overall navigability of the network.

Some interesting relationships were discovered among the parameters and the resulting corridor networks. The best way to reduce travel distance is by adding cross-aisles, however, designs with the lowest TD for the block layout also tend to have nested departments and bays. This means that to connect two parallel corridors, cross-aisles with multiple turns are required. See the left wing of the layout shown in Figure 41 for an example (circled in red). Nested departments cause the corridors to twist and turn to reach the other side. This results in a higher number of turns required to move from one side of the facility to the other, as well as making it more complicated to navigate. When the parameters are set to give the number of intersections more importance than the travel distance, these block layouts will change the most because it was the nesting that was helping reduce TD when using the rectilinear distance metric.

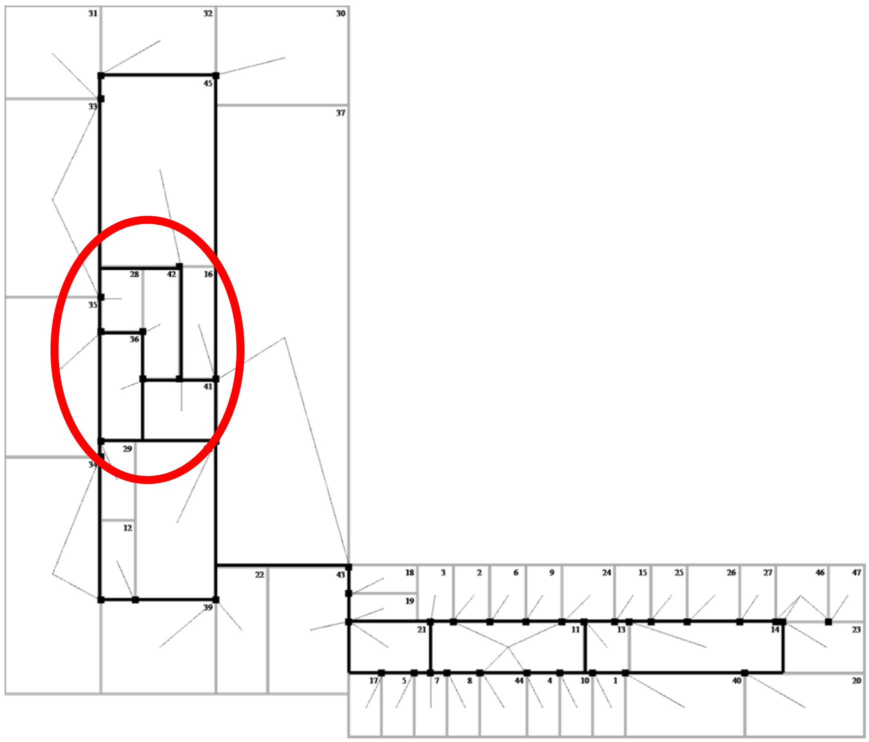
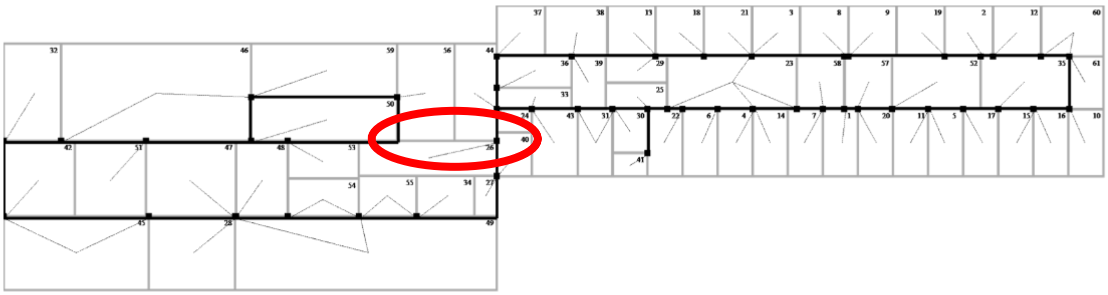
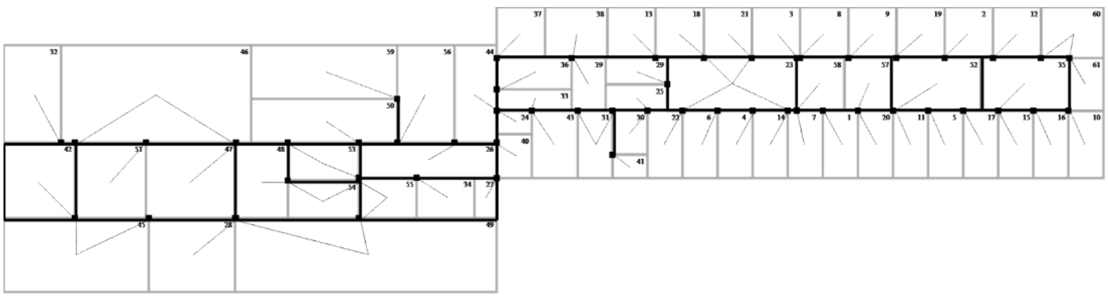


Figure 41: 10-bed best TD block layout with corridors

Block layouts with low TD also tend to produce unreasonable corridor networks (networks that require extremely long or winding trips) when minimizing the number of intersections (very high α_C values). See Figure 42 a) for an example of an unreasonable network design. The "unreasonableness" of the design can be fixed by simply adding a corridor fragment to the area circled in red. If α_C takes a low value, as it does in Figure 42 b), the resulting network is more reasonable.



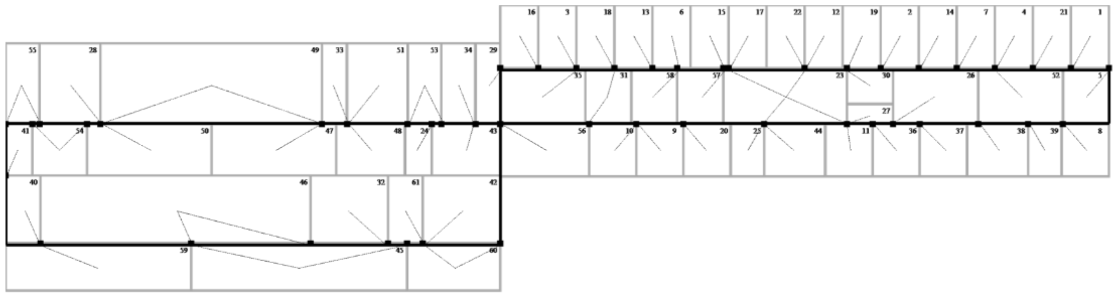
a) 22-bed block layout with best TD. Corridors generated with $\alpha_C = 100{,}000$



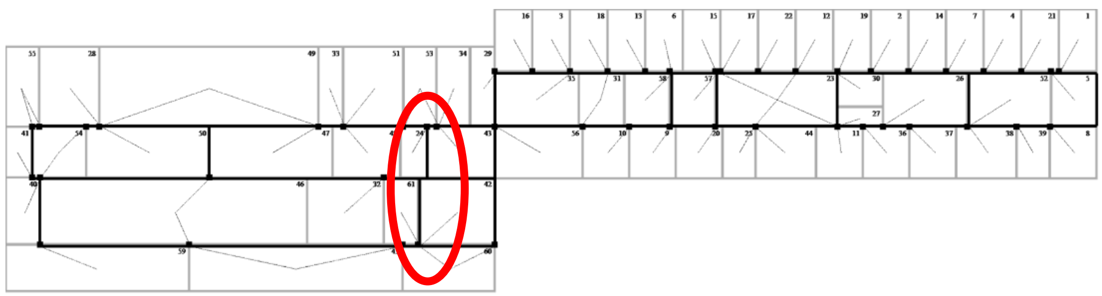
b) Same block layout with corridors generated using $\alpha_C = 1000$

Figure 42: Effect of low TD in block layouts with high α_C

This does not occur as often with block layouts with low ACC such as the one in Figure 43 a), where department edges are better aligned making it easier for the model to construct intersection-minimizing networks, and where simple cross-aisles help reduce TD without adding too many intersections (Figure 43 b). Notice too that small adjustments to department areas could be used to reduce the number of intersections by allowing continuous cross-aisles such as in the area circled in the figure. This minor change in department areas would likely reduce TD as well because trips from the upper to the lower portion of the facility (and vice versa) could reach their destination using a shorter path. This is observed across many different ACC minimizing designs.



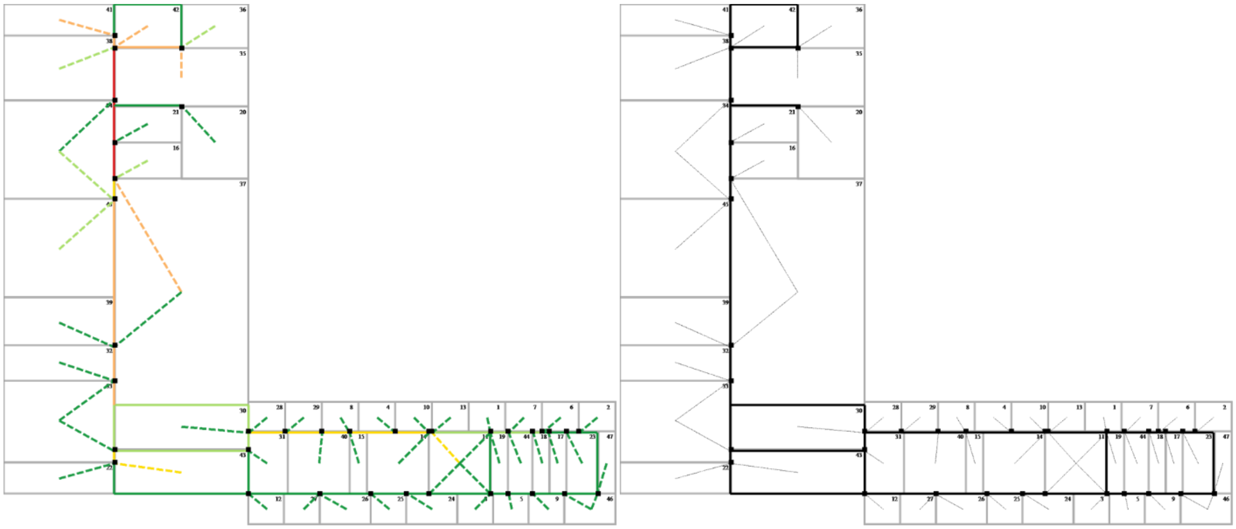
a) 22-bed block layout with best ACC. Corridors generated with $\alpha_C = 100{,}000$.



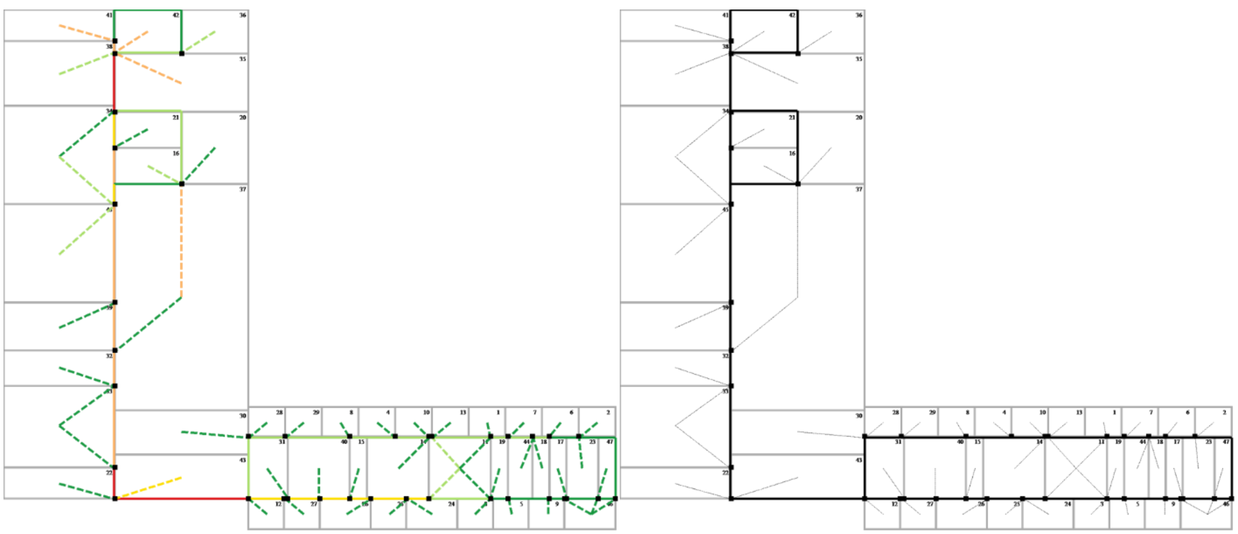
b) Same block layout with corridors generated using $\alpha_C = 1000$.

Figure 43: Effect of low ACC in block layouts with high α_C

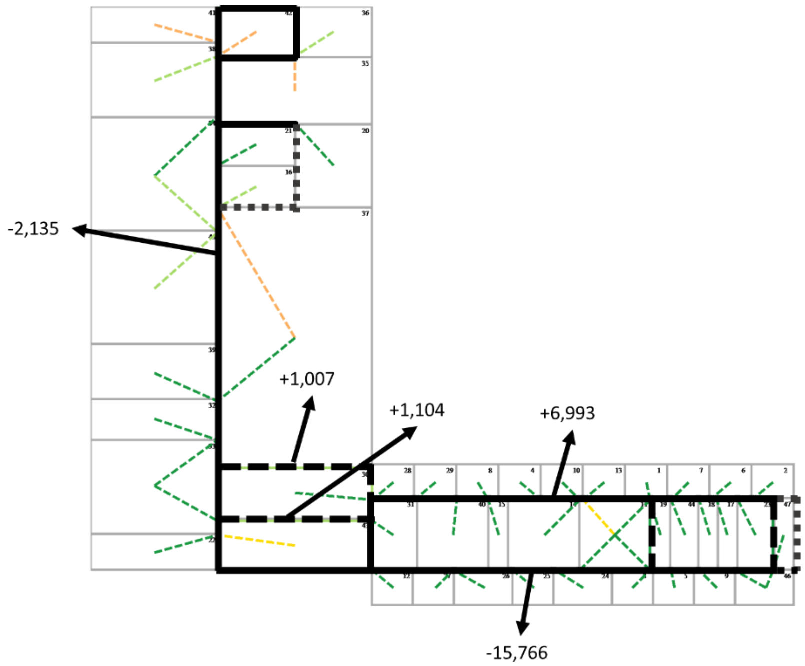
While fewer intersections generally imply easier navigation, they do so at the expense of concentrating traffic. As is seen in Figure 44, the network in a) has additional loops at the bottom-left corner, and thus additional intersections, which diffuse traffic compared to b) where they are absent. This simple addition of a loop (and with it, intersections) in one area affects the traffic distribution in the entire facility. The decrease in traffic on the left- and bottom-most corridors outweighs the increase in traffic on the other corridors, including the traffic being re-routed through the new corridors (see Figure 44 c).



a) Number of intersections = 16, corridor-ending nodes = 1 with better traffic diffusion



b) Number of intersections = 13, corridor-ending nodes = 0 with more concentrated traffic

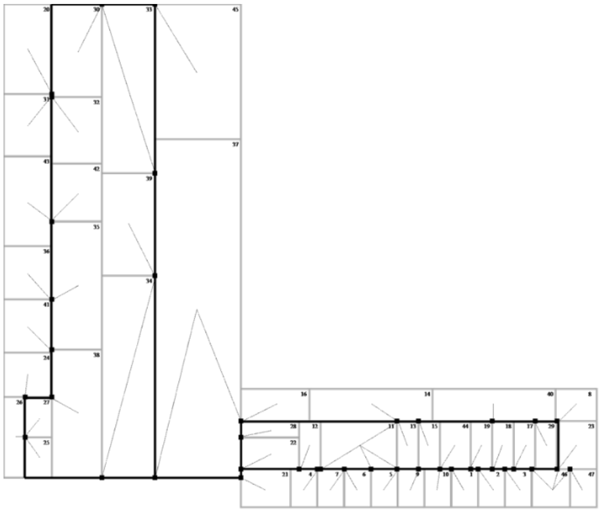


c) Changes in traffic volume from b) to a). Dotted lines represent corridor fragments that were removed, dashed lines corridor fragments that were added. Numbers indicate the change in traffic on the corridor.

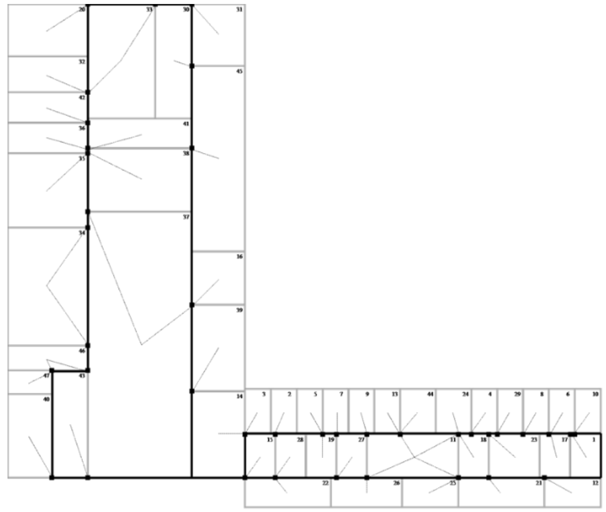
Figure 44: Traffic concentration in relation to the number of intersections

These insights offer facility designers a set of guidelines for configuring the parameter values based on their specific design objectives. Analyses like the ones here presented can help develop more detailed guidelines.

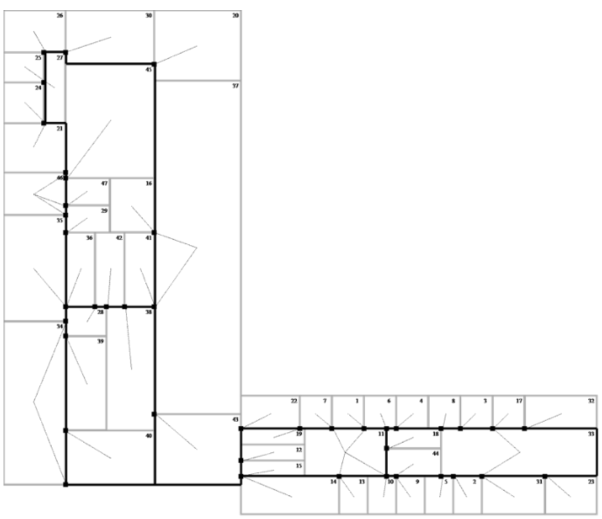
An additional finding using this model is that certain corridor network patterns seem to be robust, that is, they fit block layouts generated over all combinations of the parameters α and β and are preserved across varying values of α_C. For the 10-bed problem instance, there are two common patterns, one in which two major loops (one for each wing) are connected by a single corridor (Figure 45), and another where the entire facility is joined by one main loop (Figure 46). As α_C decreases, cross-aisles and minor loops are added, but the overall network structure remains.



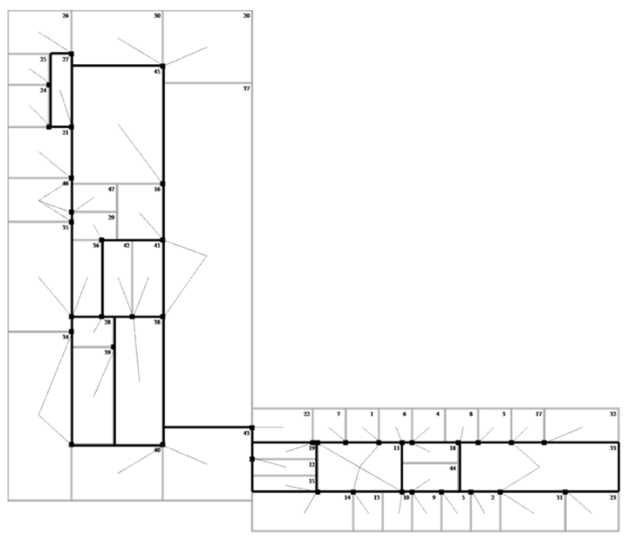
a) Block layout obtained with the LL combination of parameters, $\alpha_C = 100{,}000$



b) Block layout obtained with the LH combination of parameters, $\alpha_C = 100{,}000$

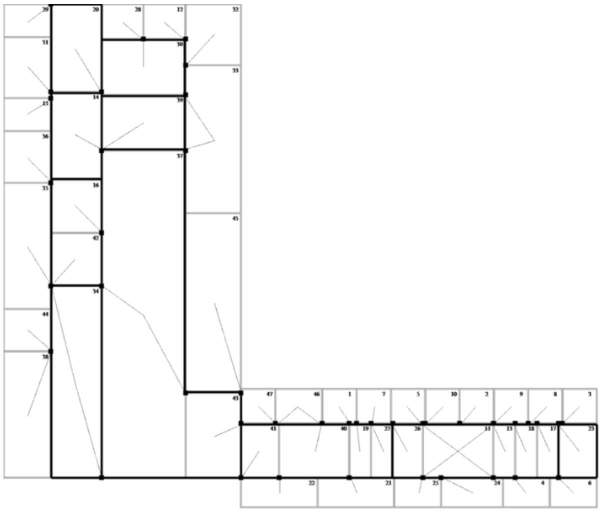


c) Block layout obtained with the HH combination of parameters, $\alpha_C = 100{,}000$

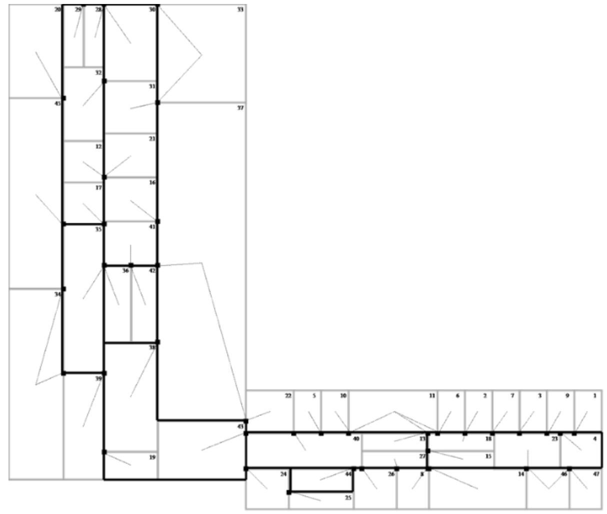


d) Same layout as c) but with $\alpha_C = 1000$

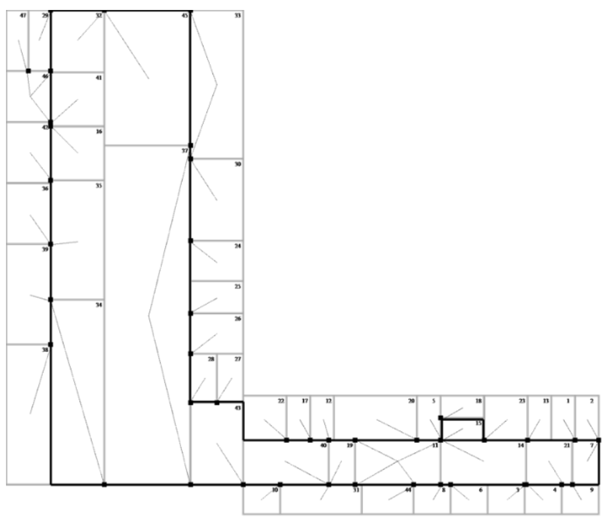
Figure 45: Double-loop pattern for the 10-bed problem



a) Block layout obtained with the LL combination of parameters, $\alpha_C = 1000$



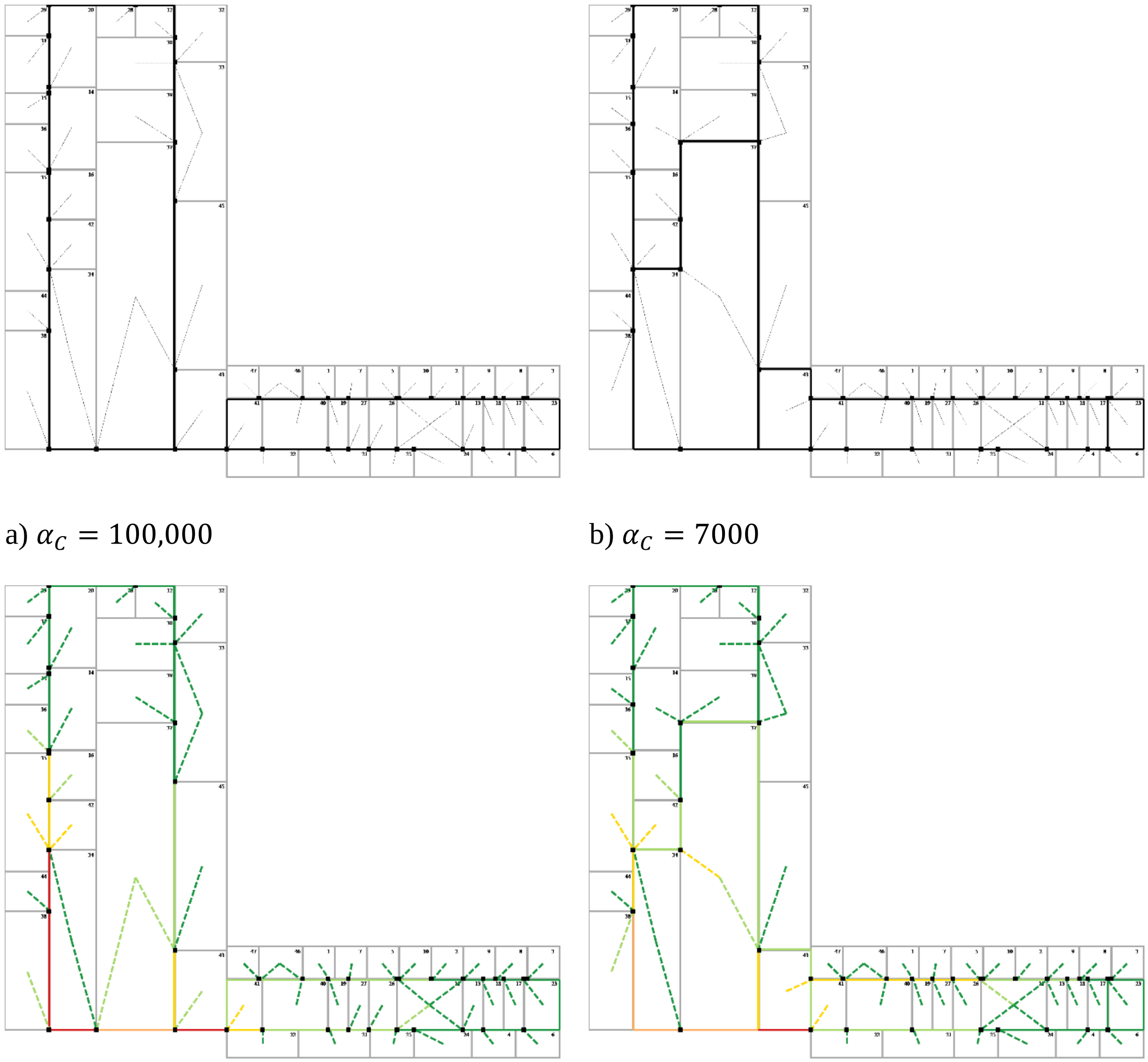
b) Block layout obtained with the HH combination of parameters, $\alpha_C = 7000$



c) Block layout obtained with the LH combination of parameters, $\alpha_C = 100{,}000$

Figure 46: Single loop pattern for the 10-bed problem

Transitions between one pattern and another can occur as the value of α_C changes. A two-looped corridor structure will become a single-loop structure as the number of intersections increases (Figure 47 a and b). This transition also results in a better traffic distribution (Figure 47 c and d).



a) $\alpha_C = 100{,}000$

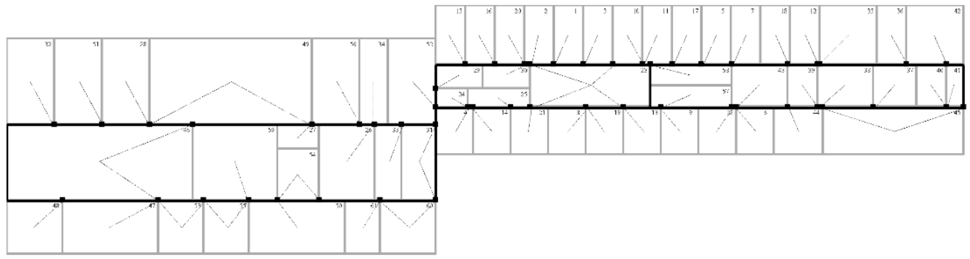
b) $\alpha_C = 7000$

c) Traffic distribution for a)

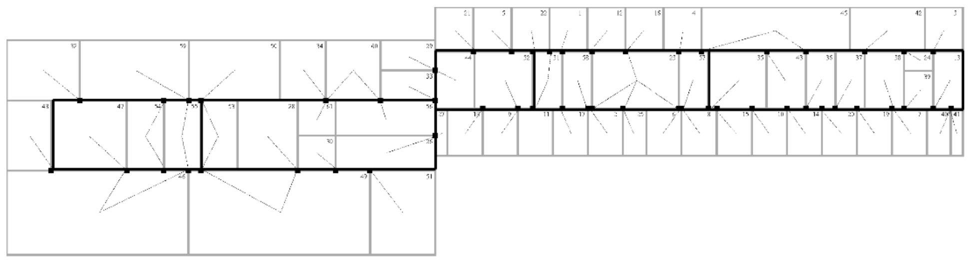
d) Traffic distribution for b)

Figure 47: Transition from one corridor pattern to another

Corridor network patterns are also observed in the 22-bed problem instance. The two-loop and single-loop patterns occur as well. See Figure 48 for an example.



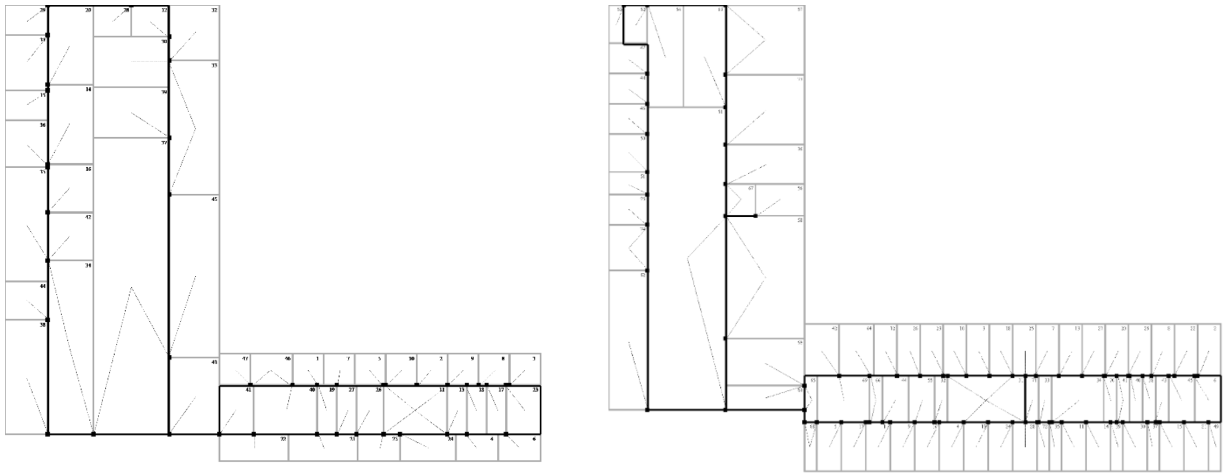
a) Block layout obtained with the HH combination of parameters, $\alpha_C = 100{,}000$



b) Block layout obtained with the LL combination of parameters, $\alpha_C = 7000$

Figure 48: Example patterns for the 22-bed problem instance

The corridor network patterns discovered for the 10-bed problem also recur in the 30-bed problem (see Figure 49). Note that both instances have the same facility shape (though different sizes). The patterns for the 22-bed problem instance also consist of loops, but whereas the loops for the 10- and 30-bed instances are at 90° from each other, they are parallel in the 22-bed case. That is, the loops follow the shape of the facility. In the 64-bed instance, the leftmost and bottom wings, which have a similar shape to the 10- and 30-bed instance, have similar corridor patterns (single and double-loops). The other two wings, however, are thin and often contain only two rows of departments aligned along a central corridor.



a) 10-bed double loop example b) 30-bed double loop example

Figure 49: Comparison of double-loop pattern for 10- and 30-bed problems

The present chapter presented and analyzed a corridor generating model for block layouts. The model allows the facility designer to input the value of control parameters which have an important effect on the structure of the corridor network, which in turn affects the travel distance and navigability of the corridors. These effects were described and analyzed, resulting in recommendations for facility designers. The model finds good solutions in a reasonable amount of time even for the largest problem instances and, hence, provides a valuable tool for practitioners.

VI. Conclusions and Future Work

6.1 Conclusions

This research introduced a new methodology for the facility layout problem in a healthcare setting. The main finding from the literature review was that current approaches to this problem were insufficient because they consisted primarily of using similar techniques and models used for manufacturing. Thus, they do not address healthcare-specific requirements that are important for hospital design. The models proposed in this dissertation, on the other hand, were developed specifically with those requirements in mind. Another important finding from the literature review was that the facility layout problem has not been properly posed as a spatial problem. Thus, the models and techniques developed so far are deficit in explaining the relationship between the spatial characteristics of the facility and the operations taking place within it. By being the first to analytically use Space Syntax, this research has begun to bridge that gap.

Space Syntax was used as the basis for the block layout proposed in this dissertation. By considering two spatial metrics, adjacency depth and straight-line access depth, the approach developed can enforce certain requirements that were not enforceable with the models found in the literature. In particular, adjacency depth allows for the modeling of proximity (how close or how far certain departments are from each other), which is a more general case than the adjacency/non adjacency constraints found in the literature. The straight-line access metric developed in this dissertation is entirely novel and results in block layouts where departments tend to be aligned along their edges. This facilitates the placement of corridor networks that are more easily navigable that those that would otherwise result. This metric represents an important contribution to the facility layout literature. Using these two spatial metrics, the constraint types used in the block layout model allow hospital designers to model specific requirements that are of great importance in healthcare, such as privacy, noise, access and infection control, natural lighting, etc.

The second model proposed, which generates corridor networks on block layouts was also developed considering healthcare specific requirements. The objective function of the corridor generation model considers the number of intersections (reflecting the complexity of the corridor network) as well as the number of turns required, both of which are important concerns in physical rehabilitation hospitals. Effective movement through the facility is vital to provide patients with quality care, as it allows caregivers to reach patient bedsides promptly. The model also considers traffic on turn nodes, which is also a concern in rehabilitation hospitals. This research is, to the best of our knowledge, the first one to include these concerns.

An important dimension that was also studied in this dissertation was the relationship between block layouts and the detailed layouts obtained by constructing corridor networks. It was found that the straight-line access spatial metric is useful in producing layouts where department edges are aligned, which facilitate the overlaying of realistic corridor networks. Another finding was that the travel distance calculated for the block layouts is a good proxy for the travel distance calculated using the corridor network. The same holds true for the metrics of accessibility and total trips weighted by turns. This means that the layouts obtained by optimizing these easy to calculate metrics will maintain their superiority even when the details of the aisles are added. This justifies the use of a sequential approach to facility design.

The models proposed allow the study of the effects of changes to the spatial configuration of the facility, both at the block and detailed stages of design. The models can be easily configured by changing the value of the objective function parameters (these are provided as inputs by the facility designer), which results in the generation of different designs. This allows designers to compare and contrast different designs easily. We envision a facility designer using the models to generate and evaluate alternative designs. This research also produced some insights about the relationship between parameters and layouts that can be developed into design principles which facility designers can use in thinking about which designs might best accomplish their goals. Examples of these include how nesting departments often reduces travel distance but can result in complex corridor networks and how block layouts with low accessibility tend to have corridor networks with fewer turns. A graphic user interface for facility designers to run the models and evaluate different designs is currently being developed and will be provided as open-source software once it is completed.

6.2 Future work

This dissertation has laid the groundwork for a series of different research avenues. The first one consists in developing new Space Syntax-based models that consider additional spatial metrics for both block and detailed layouts. The spatial metrics used in this research are only two of many different ones available in the Space Syntax literature. Spatial relations such as permeability (whether a space can be accessed directly from another) would give designers more fine-grained modeling capabilities for situations involving access or traffic control. An extension of the current corridor generating model to incorporate Space Syntax measures would also provide a valuable contribution to the field. This could be achieved by developing new constraints and objective functions that consider spatial metrics such as permeability or accessibility based on the corridor network.

A second avenue for future work is the corridor model. One of the decisions made in developing this model was that corridor width would not be taken into consideration. Since the focus of the model had more to do with navigability in terms of intersections and turns, this was done to keep the model simple. However, corridor width is important. In fact, federal regulations dictate certain corridor widths for healthcare facilities. An additional extension of the corridor model would consider iterating with the block layout and potentially altering department areas to accommodate better aisle networks.

A third research path involves the development of traffic models. Surprisingly, there is almost no research about traffic within facilities in the literature. Understanding the relationship between corridor network structures and the traffic patterns that arise would represent an important contribution to the field of facility design.

An important contribution of this research was the loosening of the rectangularity constraint, which is prevalent in the facility layout literature. An extension of the block layout model and its solution approach could allow for the overall facility's shape to be discovered as part of the search. This would require no significant modification of the nested bay tree, since any new wings in the facility would simply require adding a top-level node to the tree. By loosening the specification of facility shape (and size), interesting insights might be gained into the effect that shape has on both spatial and operational metrics of interest. This would also allow a more detailed study of the relationship between the different corridor network patterns that were identified in Chapter 5 and the facility's shape. Likewise, the nested bay tree representation could be extended to include multiple stories, which, though undesirable in physical rehabilitation hospitals, can be functional in other types of healthcare facilities.

References

Abune'Meh, M., El Meouche, R., Hijaze, I., Mebarki, A., & Shahrour, I. (2016). Optimal construction site layout based on risk spatial variability. *Automation in Construction*, 70, 167–177.

Ahmed, M. A. & Alkhamis, T. M. (2009). Simulation optimization for an emergency department healthcare unit in Kuwait. *European Journal of Operational Research*, 198(3), 936–942.

Aiello, G., Enea, M., & Galante, G. (2002). An integrated approach to the facilities and material handling system design. *International Journal of Production Research*, 40(15), 4007–4017.

Aiello, Giuseppe, Scalia, G. L., & Enea, M. (2012). A multi objective genetic algorithm for the facility layout problem based upon slicing structure encoding. *Expert Systems with Applications,* 39(12), 10352–10358.

Alagoz, O., Norman, B. A., & Smith, A. E. (2008). Determining aisle structures for facility designs using a hierarchy of algorithms. *IIE Transactions*, 40(11), 1019–1031.

Alalouch, C. & Aspinall, P. (2007). Spatial attributes of hospital multi-bed wards and preferences for privacy. *Facilities*, 25(9/10), 345–362.

Alalouch, C., Aspinall, P., & Smith, H. (2009). On locational preferences for privacy in hospital wards. *Facilities*, 27(3/4), 88–106.

Arapoglu, R. A., Norman, B. A., & Smith, A. E. (2001). Locating input and output points in facilities design-a comparison of constructive, evolutionary, and exact methods. *IEEE Transactions on Evolutionary Computation*, 5(3), 192–203.

Arnolds, I. V. & Gartner, D. (2018). Improving hospital layout planning through clinical pathway mining. *Annals of Operations Research*, 263(1), 453–477.

Arostegui, M. A., Kadipasaoglu, S. N., & Khumawala, B. M. (2006). An empirical comparison of tabu search, simulated annealing, and genetic algorithms for facilities location problems. *International Journal of Production Economics*, 103(2), 742–754.

Azadivar, F. & Wang, J. (2000). Facility layout optimization using simulation and genetic algorithms. *International Journal of Production Research*, 38(17), 4369–4383.

Benitez, G. B., Fogliatto, F. S., Cardoso, R. B., Torres, F. S., Faccin, C. S., & Dora, J. M. (2018). Systematic layout planning of a radiology reporting area to optimize radiologists' performance. *Journal of Digital Imaging*, 31(2), 193–200.

Benson, B. & Foote, B. (1997). DoorFAST: A constructive procedure to optimally layout a facility including aisles and door locations based on an aisle flow distance metric. *International Journal of Production Research*, 35(7), 1825–1842.

Bozer, Y. A., Meller, R. D., & Erlebacher, S. J. (1994). An improvement-type layout algorithm for single and multiple-floor facilities. *Management Science*, 40(7), 918–932.

Buffa, E. S., Armour, G. C., & Vollmann, T. E. (1964). Allocating facilities with CRAFT. *Harvard Business Review*, 42(2), 136–158.

Burkard, R.E. & Bönniger, T. (1983). A heuristic for quadratic Boolean programs with applications to quadratic assignment problems. *European Journal of Operational Research*, 13(4), 374–386.

Burkard, R.E. & Stratmann, K.-H. (1978). Numerical investigations on quadratic assignment problems. *Naval Research Logistics Quarterly*, 25(1), 129–148.

Burkard, R.E. & Rendl, F. (1984). A thermodynamically motivated simulation procedure for combinatorial optimization problems. *European Journal of Operational Research*, 17(2), 169–174.

Butler, T.W., Karwan, K. R., & Sweigart, J. R. (1992). Multi-level strategic evaluation of hospital plans and decisions. *Journal of the Operational Research Society*, 43(7), 665–675.

Butler, T.W., Karwan, K. R., Sweigart, J. R., & Reeves, G. R. (1992). An integrative model-based approach to hospital layout. *IIE Transactions*, 24(2), 144–152.

Cai, H. & Zimring, C. (2017). Cultural impacts on nursing unit design: a comparative study on Chinese nursing unit typologies and their US counterparts using space syntax. *Environment and Planning B: Urban Analytics and City Science*, 1–22.

Centers for Medicare and Medicaid Services (CMS). (2012). *The Inpatient Rehabilitation Facility - Patient Assessment Instrument (IRF-PAI) Training Manual*.

Cheng, M.-Y., & Lien, L.-C. (2012). A hybrid AI-based particle bee algorithm for facility layout optimization. *Engineering with Computers*, 28(1), 57–69.

Choudhary, R., Bafna, S., Heo, Y., Hendrich, A., & Chow, M. (2010). A predictive model for computing the influence of space layouts on nurses' movement in hospital units. *Journal of Building Performance Simulation*, 3(3), 171–184.

Chraibi, A., Osman, I. H., & Kharraja, S. (2019). Adaptive layout for operating theatre in hospitals: different mathematical models for optimal layouts. *Annals of Operations Research*, 272(1–2), 493–527.

Deb, S. K. & Bhattacharyya, B. (2005). Solution of facility layout problems with pickup/drop-off locations using random search techniques. *International Journal of Production Research*, 43(22), 4787–4812.

Delon, G. L. (1970). A methodology for total hospital design. *Health Services Research,* (Fall 1970), 210–223.

Drira, A., Pierreval, H., & Hajri-Gabouj, S. (2013). Design of a robust layout with information uncertainty increasing over time: a fuzzy evolutionary approach. *Engineering Applications of Artificial Intelligence*, 26(3), 1052–1060.

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

Enea, M., Galante, G., & Panascia, E. (2005). The facility layout problem approached using a fuzzy model and a genetic search. *Journal of Intelligent Manufacturing*, 16(3), 303–316.

Fortenberry, J. C. & Cox, J. F. (1985). Multiple criteria approach to the facilities layout problem. *International Journal of Production Research*, 23(4), 773–782.

Francis, R. L., McGinnis, L. F., & White, J. A. (1992). *Facility Layout and Location: An Analytical Approach*. Prentice Hall.

Friedrich, C., Klausnitzer, A., & Lasch, R. (2018). Integrated slicing tree approach for solving the facility layout problem with input and output locations based on contour distance. *European Journal of Operational Research*, 270(3), 837–851.

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

Glover, F. & Laguna, M. (1998). Tabu search. In *Handbook of Combinatorial Optimization* (pp. 2093–2229). Springer.

Gue, K. R., Ivanović, G., & Meller, R. D. (2012). A unit-load warehouse with multiple pickup and deposit points and non-traditional aisles. *Transportation Research Part E: Logistics and Transportation Review*, 48(4), 795–806.

Hahn, P. M. & Krarup, J. (2001). A hospital facility layout problem finally solved. *Journal of Intelligent Manufacturing*, 12(5), 487–496.

Haq, S. & Luo, Y. (2012). Space syntax in healthcare facilities research: A review. *HERD: Health Environments Research & Design Journal*, 5(4), 98–117.

Haq, S. & Zimring, C. (2003). Just down the road a piece: The development of topological knowledge of building layouts. *Environment and Behavior*, 35(1), 132–160.

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

Helme, L., Derix, C., & Izaki, A. (2014). Spatial configuration: Semi-automatic methods for layout generation in practice. *The Journal of Space Syntax*, 5(1), 35–49.

Hendrich, A., Chow, M. P., Bafna, S., Choudhary, R., Yeonsook, H., & Skierczynski, A. (2009). Unit-related factors that affect nursing time with patients: spatial analysis of the time and motion study. *Health Environments Research & Design Journal (HERD)*, 2(2), 5–20.

Heragu, S. (2008). *Facilities Design* (3rd ed.). Boca Raton, FL: CRC Press.

Hillier, B. (1998). *Space Is the Machine: A Configurational Theory of Architecture*. Cambridge University Press.

Hillier, B., & Hanson, J. (1993). *The Social Logic of Space*. Bell & Bain.

Hillier, Bill. (1998). A note on the intuiting of form: three issues in the theory of design. *Environment & Planning B: Planning & Design Anniversary Issue*, 25(7), 37–40.

Hillier, Bill, Hanson, J., & Graham, H. (1987). Ideas are in things - an application of the Space Syntax method to discovering house genotypes. *Environment & Planning B: Planning & Design*, 14(4), 363–385.

Jankovits, I., Luo, C., Anjos, M. F., & Vannelli, A. (2011). A convex optimisation framework for the unequal-areas facility layout problem. *European Journal of Operational Research*, 214(2), 199–215.

Johnson, R. V. (1992). Finding building shapes that minimize mean trip times. *Computer-Aided Design*, 24(2), 105–113.

Joines, J. A. & Houck, C. R. (1994). On the use of non-stationary penalty functions to solve nonlinear constrained optimization problems with GA's. *Proceedings of the First IEEE Conference on Evolutionary Computation. IEEE World Congress on Computational Intelligence*, 579–584. IEEE.

Kim, J. G. & Goetschalckx, M. (2005). An integrated approach for the concurrent determination of the block layout and the input and output point locations based on the contour distance. *International Journal of Production Research*, 43(10), 2027–2047.

Kim, J. G. & Kim, Y. D. (1999). A branch and bound algorithm for locating input and output points of departments on the block layout. *The Journal of the Operational Research Society*, 50(5), 517–525.

Kim, J.-G. & Kim, Y.-D. (1998). A space partitioning method for facility layout problemswith shape constraints. *IIE Transactions*, 30(10), 947–957.

Kim, J.-G. & Kim, Y.-D. (2000). Layout planning for facilities with fixed shapes and input and output points. *International Journal of Production Research*, 38(18), 4635–4653.

Kim, Y. & Lee, H. W. (2010). Analyzing user costs in a hospital: methodological implication of Space Syntax to support whole-life target value design. *Lean Construction Journal*, 55–65.

Kochhar, J. (1998). MULTI-HOPE: a tool for multiple floor layout problems. *International Journal of Production Research*, 36(12), 3421–3435.

Kochhar, J. S., Foster, B. T., & Heragu, S. S. (1998). HOPE: A genetic algorithm for the unequal area facility layout problem. *Computers & Operations Research*, 25(7–8), 583–594.

Konak, A., Kulturel-Konak, S., Norman, B. A., & Smith, A. E. (2006). A new mixed integer programming formulation for facility layout design using flexible bays. *Operations Research Letters*, 34(6), 660–672.

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

Kulturel-Konak, S., Smith, A. E., & Norman, B. A. (2004). Layout optimization considering production uncertainty and routing flexibility. *International Journal of Production Research*, 42(21), 4475–4493.

Kulturel-Konak, S. (2007). Approaches to uncertainties in facility layout problems: perspectives at the beginning of the 21st Century. *Journal of Intelligent Manufacturing*, 18(2), 273.

Kulturel-Konak, S. (2012). A linear programming embedded probabilistic tabu search for the unequal-area facility layout problem with flexible bays. *European Journal of Operational Research*, 223(3), 614–625.

Kulturel-Konak, S., & Konak, A. (2011). A new relaxed flexible bay structure representation and particle swarm optimization for the unequal area facility layout problem. *Engineering Optimization*, 43(12), 1263–1287.

Kulturel-Konak, S., Norman, B. A., Coit, D. W., & Smith, A. E. (2004). Exploiting tabu search memory in constrained problems. *INFORMS Journal on Computing*, 16(3), 241–254.

Kulturel-Konak, S., Smith, A. E., & Norman, B. A. (2007). Bi-objective facility expansion and relayout considering monuments. *IIE Transactions*, 39(7), 747–761.

Laguna, M. (2018). Tabu and scatter search: principles and practice. In *Recent Advances in Optimization and Modeling of Contemporary Problems* (pp. 130–157). INFORMS.

Lee, R. & Moore, J. (1967). CORELAP - Computarized relationship layout planning. *Journal of Industrial Engineering*, 18(3), 1994–2000.

Lee, Y. H. & Lee, M. H. (2002). A shape-based block layout approach to facility layout problems using hybrid genetic algorithm. *Computers & Industrial Engineering*, 42(2–4), 237–248.

Li, J. & Smith, A. E. (2018). Block layout for attraction-based enterprises. *European Journal of Operational Research*, 266(3), 1100–1112.

Liang, L. Y. & Chao, W. C. (2008). The strategies of tabu search technique for facility layout optimization. *Automation in Construction*, 17(6), 657–669.

Liu, Q. & Meller, R. D. (2007). A sequence-pair representation and MIP-model-based heuristic for the facility layout problem with rectangular departments. *IIE Transactions*, 39(4), 377–394.

Lu, Y. & Zimring, C. (2012). Can intensive care staff see their patients? An improved visibility analysis methodology. *Environment and Behavior*, 44(6), 861–876.

Lund, B. & Smith, J. W. (2010). *A multi-stage CUDA kernel for Floyd-Warshall*. arXiv.

MacAllister, L., Zimring, C., & Ryherd, E. (2016). Environmental variables that influence patient satisfaction. *HERD: Health Environments Research & Design Journal*, 10(1), 155–169.

Masel, D. & Marinchek, D. (2016). Non-traditional aisle design for a manufacturing facility layout. *14th IMHRC Proceedings*. Presented at the Institute for Material Handling Research Colloquium, Karlsruhe, Germany.

McKendall, A. R., & Liu, W.-H. (2012). New tabu search heuristics for the dynamic facility layout problem. *International Journal of Production Research*, 50(3), 867–878.

Meller, R.D., & Gau, K.-Y. (1996a). Facility layout objective functions and robust layouts. *International Journal of Production Research*, 34(10), 2727–2742.

Meller, R.D., & Gau, K.-Y. (1996b). The facility layout problem: recent and emerging trends and perspectives. *Journal of Manufacturing Systems*, 15(5), 351–366.

Montreuil, B. (1990). A modeling framework for integrating layout design and flow network design. *Proceedings of the Material Handling Research Colloquium*, 43-58. Hebron, KY.

Montreuil, B. & Ratliff, H. D. (1989). Utilizing cut trees as design skeletons for facility layout. *IIE Transactions*, 21(2), 136–143.

Morgareidge, D., Cai, H., & Jia, J. (2014). Performance-driven design with the support of digital tools: applying discrete event simulation and space syntax on the design of the emergency department. *Frontiers of Architectural Research*, 3(3), 250–264.

Mowrey, C. H., Parikh, P. J., & Gue, K. R. (2019). The impact of rack layout on visual experience in a retail store. *INFOR: Information Systems and Operational Research*, 57(1), 75–98.

Norman, B.A., Smith, A. E., Yildirim, E., & Tharmmaphornphilas, W. (2001). An evolutionary approach to incorporating intradepartmental flow into facilities design. *Advances in Engineering Software*, 32(6), 443–453.

Norman, B.A, Arapoglu, R. A., & Smith, A. E. (2001). Integrated facilities design using a contour distance metric. *IIE Transactions*, 33(4), 337–344.

Norman, B.A., & Smith, A. E. (2006). A continuous approach to considering uncertainty in facility design. *Computers & Operations Research*, 33(6), 1760–1775.

Ozdemir, G., Smith, A. E., & Norman, B. A. (2003). Incorporating heterogeneous distance metrics within block layout design. *International Journal of Production Research*, 41(5), 1045–1056.

Ozgormus, E., & Smith, A. E. (2018). A data-driven approach to grocery store block layout. *Computers & Industrial Engineering*. (In press)

Öztürkoğlu, Ö. & Hoser, D. (2019). A discrete cross aisle design model for order-picking warehouses. *European Journal of Operational Research*, 275(2), 411–430.

Peer, S. K., & Sharma, D. K. (2008). Human–computer interaction design with multi-goal facilities layout model. *Computers & Mathematics with Applications*, 56(9), 2164–2174.

Peponis, J., Zimring, C., & Choi, Y. K. (1990). Finding the building in wayfinding. *Environment and Behavior*, 22(5), 555–590.

Raman, D., Nagalingam, S. V., & Gurd, B. W. (2009). A genetic algorithm and queuing theory based methodology for facilities layout problem. *International Journal of Production Research*, 47(20), 5611–5635.

Rashid, M. (2009). Hospital design and face-to-face interaction among clinicians: a theoretical model. *Health Environments Research & Design Journal (HERD)*, 2(4), 62–84.

Rismanchian, F., & Lee, Y. H. (2017). Process Mining–Based Method of Designing and Optimizing the Layouts of Emergency Departments in Hospitals. *HERD: Health Environments Research & Design Journal*, 10(4), 105–120.

Rosenblatt, M. J. (1979). The facilities layout problem: a multi-goal approach. *International Journal of Production Research*, 17(4), 323–332.

Rosenblatt, M. J. & Lee, H. L. (1987). A robustness approach to facilities design. *International Journal of Production Research*, 25(4), 479–486.

Sadek, A. H. & Shepley, M. M. (2016). Space Syntax analysis. *HERD: Health Environments Research & Design Journal*, 10(1), 114–129.

Sahin, R. & Turkbey, O. (2009). A new hybrid tabu-simulated annealing heuristic for the dynamic facility layout problem. *International Journal of Production Research*, 47(24), 6855–6873.

Scholz, D., Jaehn, F., & Junker, A. (2010). Extensions to STaTS for practical applications of the facility layout problem. *European Journal of Operational Research*, 204(3), 463–472.

Seo, H.-B., Choi, Y.-S., & Zimring, C. (2011). Impact of hospital unit design for patient-centered care on nurses' behavior. *Environment and Behavior*, 43(4), 443–468.

Singh, S. P. & Sharma, R. R. K. (2008). Two-level modified simulated annealing based approach for solving facility layout problem. *International Journal of Production Research*, 46(13), 3563–3582.

Skorin-Kapov, J. (1990). Tabu search applied to the quadratic assignment problem. *ORSA Journal on Computing*, 2(1), 33.

Taillard, E. (1991). Robust taboo search for the quadratic assignment problem. *Parallel Computing*, 17(4–5), 443–455.

Tam, K. Y. (1992). A simulated annealing algorithm for allocating space to manufacturing cells. *International Journal of Production Research*, 30(1), 63.

Tate, D. M. & Smith, A. E. (1995a). A genetic approach to the quadratic assignment problem. *Computers & Operations Research*, 22(1), 73–83.

Tate, D. M. & Smith, A. E. (1995b). Unequal-area facility layout by genetic search. *IIE Transactions*, 27(4), 465–472.

Tretheway, S. J. & Foote, B. L. (1994). Automatic computation and drawing of facility layouts with logical aisle structures. *The International Journal of Production Research*, 32(7), 1545–1555.

Trzpuc, S. J. & Martin, C. S. (2010). Application of space syntax theory in the study of medical-surgical nursing units in urban hospitals. *HERD: Health Environments Research & Design Journal*, 4(1), 34–55.

UB Foundation Activities, Inc. (2013). *UDS-PRO* [Uniform Data System for Medical Rehabilitation].

Ulrich, R. S., Zimring, C., Barch, X. Z., Dubose, J., Seo, H.-B., Choi, Y.-S., … Joseph, A. (2008). A review of the research literature on evidence-based healthcare design. *HERD: Health Environments Research & Design Journal*, 1(3), 61–125.

Ulutas, B. H. & Kulturel-Konak, S. (2012). An artificial immune system based algorithm to solve unequal area facility layout problem. *Expert Systems with Applications*, 39(5), 5384–5395.

Ünlü, A., Ülken, G., & Edgü, E. (2005). A space syntax based model in evacuation of hospitals. *Proceedings, 5th International Space Syntax Symposium, Delft*, 161.

Urban, T. L. (1987). A multiple criteria model for the facilities layout problem. *International Journal of Production Research*, 25(12), 1805.

Wang, G., Yan, Y., Zhang, X., Shangguan, J., & Xiao, Y. (2008). A simulation optimization approach for facility layout problem. *IEEE International Conference on Industrial Engineering and Engineering Management*, 734–738.

Wang, M.-J., Hu, M. H., & Ku, M.-Y. (2005). A solution to the unequal area facilities layout problem by genetic algorithm. *Computers in Industry*, 56(2), 207–220.

Wang, T.-K., Yang, T., Yang, C.-Y., & Chan, F. T. (2015). Lean principles and simulation optimization for emergency department layout design. *Industrial Management & Data Systems*, 115(4), 678–699.

Wilhelm, M. R. & Ward, T. L. (1987). Solving quadratic assignment problems by 'simulated annealing.' *IIE Transactions*, 19(1), 107–119.

Wong, K. Y. (2010). Applying ant system for solving unequal area facility layout problems. *European Journal of Operational Research*, 202(3), 730–746.

Wong, S. S. Y. & Chan, K. C. C. (2009). EvoArch: An evolutionary algorithm for architectural layout design. *Computer-Aided Design*, 41(9), 649–667.

Wu, Y. & Appleton, E. (2002). The optimisation of block layout and aisle structure by a genetic algorithm. *Computers & Industrial Engineering*, 41(4), 371–387.

Yang, T. & Hung, C.-C. (2007). Multiple-attribute decision making methods for plant layout design problem. *Robotics and Computer-Integrated Manufacturing*, 23(1), 126–137.

Yapicioglu, H. & Smith, A. E. (2012a). A bi-objective model for the retail spatial design problem. *Engineering Optimization*, 44(3), 243–266.

Yapicioglu, H. & Smith, A. E. (2012b). Retail space design considering revenue and adjacencies using a racetrack aisle network. *IIE Transactions*, 44(6), 446–458.

Yeh, I.-C. (2006). Architectural layout optimization using annealed neural network. *Automation in Construction*, 15(4), 531–539.

Yi, L. (2010). Measuring the structure of visual fields in nursing units. *Health Environments Research & Design Journal (HERD)*, 3(2), 48–59.

Zuo, X., Li, B., Huang, X., Zhou, M., Cheng, C., Zhao, X., & Liu, Z. (2019). Optimizing hospital emergency department layout via multiobjective tabu search. *IEEE Transactions on Automation Science and Engineering*.

Zwart, J. van der & Voordt, T. J. M. van der. (2015). Pre-occupancy evaluation of patient satisfaction in hospitals. *HERD: Health Environments Research & Design Journal*, 9(1), 110–124.

Appendix A: Problem Data

10-bed Problem Instance

Id	Department Type	Max I/Os	Min Area	Max Area	Contact with Outside	Aspect Ratio
1	PatientRoom	1	100	200	TRUE	2
2	PatientRoom	1	100	200	TRUE	2
3	PatientRoom	1	100	200	TRUE	2
4	PatientRoom	1	100	200	TRUE	2
5	PatientRoom	1	100	200	TRUE	2
6	PatientRoom	1	100	200	TRUE	2
7	PatientRoom	1	100	200	TRUE	2
8	PatientRoom	1	100	200	TRUE	2
9	PatientRoom	1	100	200	TRUE	2
10	PatientRoom	1	100	200	TRUE	2
11	NurseStation	4	400	800	FALSE	4
12	NurseManager	1	100	200	FALSE	4
13	MedPrep	1	100	200	FALSE	4
14	Pharmacy	1	400	800	FALSE	4
15	PharmacyAdmin	1	100	200	FALSE	4
16	Charting	1	200	400	FALSE	4
17	SpeechTherapy	1	100	200	FALSE	4
18	SpeechTherapy	1	100	200	FALSE	4
19	CaseManager	1	100	200	FALSE	4
20	MedicalRecords	1	400	800	FALSE	4
21	RecordsOffice	1	200	400	FALSE	4
22	StaffLounge	1	200	400	FALSE	4
23	FamilyConference	1	200	400	FALSE	4
24	PhysicianOffice	1	150	300	TRUE	4
25	ExamRoom	1	100	200	TRUE	4
26	WaitingRoom	1	150	300	FALSE	4
27	Reception	1	100	200	FALSE	4
28	HumanResources	1	100	200	FALSE	4
29	HRFiles	1	100	200	FALSE	4
30	Records	1	400	800	FALSE	4
31	Conference	1	250	500	FALSE	4
32	Conference	1	250	500	FALSE	4
33	Offices	2	800	1600	TRUE	4
34	Dining	2	1000	2000	TRUE	4
35	Kitchen	1	600	1200	FALSE	4
36	Freezer	1	200	400	FALSE	4
37	TherapyGym	2	3000	6000	TRUE	4
38	ADL	1	600	1200	FALSE	4
39	TherapyRec	1	400	800	FALSE	4
40	ActivityArea	1	400	800	FALSE	4
41	Storage	1	200	400	FALSE	4
42	Storage	1	200	400	FALSE	4
43	RestRoom	1	400	800	FALSE	4
44	CleanSoiled	1	150	300	FALSE	4
45	Maintenance	1	1000	2000	FALSE	4
46	Lobby	4	150	300	TRUE	4

47	Desk	1	100	200	FALSE	4

22-bed Problem Instance

Id	Department Type	Max I/Os	Min Area	Max Area	Contact with Outside	Aspect Ratio
1	PatientRoom	1	150	300	TRUE	2
2	PatientRoom	1	150	300	TRUE	2
3	PatientRoom	1	150	300	TRUE	2
4	PatientRoom	1	150	300	TRUE	2
5	PatientRoom	1	150	300	TRUE	2
6	PatientRoom	1	150	300	TRUE	2
7	PatientRoom	1	150	300	TRUE	2
8	PatientRoom	1	150	300	TRUE	2
9	PatientRoom	1	150	300	TRUE	2
10	PatientRoom	1	150	300	TRUE	2
11	PatientRoom	1	150	300	TRUE	2
12	PatientRoom	1	150	300	TRUE	2
13	PatientRoom	1	150	300	TRUE	2
14	PatientRoom	1	150	300	TRUE	2
15	PatientRoom	1	150	300	TRUE	2
16	PatientRoom	1	150	300	TRUE	2
17	PatientRoom	1	150	300	TRUE	2
18	PatientRoom	1	150	300	TRUE	2
19	PatientRoom	1	150	300	TRUE	2
20	PatientRoom	1	150	300	TRUE	2
21	PatientRoom	1	150	300	TRUE	2
22	PatientRoom	1	150	300	TRUE	2
23	NurseStation	4	450	900	FALSE	4
24	NurseManager	1	50	100	FALSE	4
25	MedPrep	1	100	200	FALSE	4
26	Pharmacy	1	300	600	FALSE	4
27	PharmacyAdmin	1	50	100	FALSE	4
28	Charting	1	300	600	FALSE	4
29	SpeechTherapy	1	100	200	FALSE	4
30	SpeechTherapy	1	100	200	FALSE	4
31	CaseManager	2	150	300	FALSE	4
32	MedicalRecords	1	300	600	FALSE	4
33	RecordsOffice	1	100	200	FALSE	4
34	StaffLounge	1	150	300	FALSE	4
35	FamilyConference	1	300	600	FALSE	4
36	PhysicianOffice	1	150	300	FALSE	4
37	ExamRoom	1	150	300	FALSE	4
38	WaitingRoom	1	200	400	FALSE	4
39	Reception	1	100	200	FALSE	4
40	HumanResources	1	100	200	FALSE	4
41	HRFiles	1	50	100	FALSE	4
42	Records	1	300	600	FALSE	4
43	Conference	1	200	400	FALSE	4
44	Conference	1	200	400	FALSE	4
45	Offices	2	600	1200	FALSE	5

46	Dining	2	1200	2400	FALSE	4
47	Kitchen	1	400	800	FALSE	4
48	Freezer	1	200	400	FALSE	4
49	TherapyGym	2	1200	2400	TRUE	4
50	ADL	1	400	800	FALSE	4
51	TherapyRec	1	300	600	FALSE	4
52	ActivityArea	1	300	600	FALSE	4
53	Storage	2	150	300	FALSE	4
54	Storage	2	150	300	FALSE	4
55	Storage	2	150	300	FALSE	4
56	RestRoom	1	300	600	FALSE	4
57	CleanSoiled	1	150	300	FALSE	4
58	CleanSoiled	1	150	300	FALSE	4
59	Maintenance	1	500	1000	FALSE	4
60	Lobby	4	200	400	TRUE	4
61	Desk	1	100	200	FALSE	4

30-bed Problem Instance

Id	Department Type	Max I/Os	Min Area	Max Area	Contact with Outside	Aspect Ratio
1	PatientRoom	1	200	400	TRUE	4
2	PatientRoom	1	200	400	TRUE	4
3	PatientRoom	1	200	400	TRUE	4
4	PatientRoom	1	200	400	TRUE	4
5	PatientRoom	1	200	400	TRUE	4
6	PatientRoom	1	200	400	TRUE	4
7	PatientRoom	1	200	400	TRUE	4
8	PatientRoom	1	200	400	TRUE	4
9	PatientRoom	1	200	400	TRUE	4
10	PatientRoom	1	200	400	TRUE	4
11	PatientRoom	1	200	400	TRUE	4
12	PatientRoom	1	200	400	TRUE	4
13	PatientRoom	1	200	400	TRUE	4
14	PatientRoom	1	200	400	TRUE	4
15	PatientRoom	1	200	400	TRUE	4
16	PatientRoom	1	200	400	TRUE	4
17	PatientRoom	1	200	400	TRUE	4
18	PatientRoom	1	200	400	TRUE	4
19	PatientRoom	1	200	400	TRUE	4
20	PatientRoom	1	200	400	TRUE	4
21	PatientRoom	1	200	400	TRUE	4
22	PatientRoom	1	200	400	TRUE	4
23	PatientRoom	1	200	400	TRUE	4
24	PatientRoom	1	200	400	TRUE	4
25	PatientRoom	1	200	400	TRUE	4
26	PatientRoom	1	200	400	TRUE	4
27	PatientRoom	1	200	400	TRUE	4
28	PatientRoom	1	200	400	TRUE	4
29	PatientRoom	1	200	400	TRUE	4
30	PatientRoom	1	200	400	TRUE	4

31	NurseStation	4	600	1000	FALSE	4
32	NurseManager	1	100	200	FALSE	4
33	MedPrep	1	100	200	FALSE	4
34	Pharmacy	1	400	600	FALSE	4
35	PharmacyAdmin	1	100	200	FALSE	4
36	Charting	1	400	800	FALSE	4
37	SpeechTherapy	1	100	150	FALSE	4
38	SpeechTherapy	1	100	150	FALSE	4
39	SpeechTherapy	1	100	150	FALSE	4
40	SpeechTherapy	1	100	150	FALSE	4
41	CaseManager	2	100	200	FALSE	4
42	MedicalRecords	1	300	600	FALSE	4
43	RecordsOffice	1	100	200	FALSE	4
44	StaffLounge	1	200	300	FALSE	4
45	FamilyConference	1	200	400	FALSE	4
46	PhysicianOffice	1	100	200	TRUE	4
47	PhysicianOffice	1	100	200	TRUE	4
48	ExamRoom	1	100	200	TRUE	4
49	ExamRoom	1	100	200	TRUE	4
50	WaitingRoom	1	150	300	FALSE	4
51	Reception	1	50	200	FALSE	4
52	HumanResources	1	100	100	FALSE	4
53	HRFiles	1	50	100	FALSE	4
54	Records	1	400	800	FALSE	4
55	Conference	1	200	500	FALSE	4
56	Conference	1	200	500	FALSE	4
57	Offices	2	800	2000	TRUE	4
58	Dining	2	1500	2500	TRUE	4
59	Kitchen	1	500	2000	FALSE	4
60	Freezer	1	200	600	FALSE	4
61	TherapyGym	2	3500	5000	TRUE	4
62	ADL	1	800	1500	FALSE	4
63	TherapyRec	1	500	1000	FALSE	4
64	ActivityArea	1	300	600	FALSE	4
65	Storage	2	100	300	FALSE	4
66	Storage	2	100	300	FALSE	4
67	Storage	2	100	300	FALSE	4
68	Storage	2	100	300	FALSE	4
69	RestRoom	1	400	1000	FALSE	4
70	CleanSoiled	1	100	300	FALSE	4
71	CleanSoiled	1	100	300	FALSE	4
72	CleanSoiled	1	100	300	FALSE	4
73	Maintenance	1	800	1600	FALSE	4
74	Lobby	4	200	500	TRUE	4
75	Desk	1	100	200	FALSE	4

64-bed Problem Instance

Id	Department Type	Max I/Os	Min Area	Max Area	Contact with Outside	Aspect Ratio
1	PatientRoom	1	200	500	TRUE	4
2	PatientRoom	1	200	500	TRUE	4
3	PatientRoom	1	200	500	TRUE	4
4	PatientRoom	1	200	500	TRUE	4
5	PatientRoom	1	200	500	TRUE	4
6	PatientRoom	1	200	500	TRUE	4
7	PatientRoom	1	200	500	TRUE	4
8	PatientRoom	1	200	500	TRUE	4
9	PatientRoom	1	200	500	TRUE	4
10	PatientRoom	1	200	500	TRUE	4
11	PatientRoom	1	200	500	TRUE	4
12	PatientRoom	1	200	500	TRUE	4
13	PatientRoom	1	200	500	TRUE	4
14	PatientRoom	1	200	500	TRUE	4
15	PatientRoom	1	200	500	TRUE	4
16	PatientRoom	1	200	500	TRUE	4
17	PatientRoom	1	200	500	TRUE	4
18	PatientRoom	1	200	500	TRUE	4
19	PatientRoom	1	200	500	TRUE	4
20	PatientRoom	1	200	500	TRUE	4
21	PatientRoom	1	200	500	TRUE	4
22	PatientRoom	1	200	500	TRUE	4
23	PatientRoom	1	200	500	TRUE	4
24	PatientRoom	1	200	500	TRUE	4
25	PatientRoom	1	200	500	TRUE	4
26	PatientRoom	1	200	500	TRUE	4
27	PatientRoom	1	200	500	TRUE	4
28	PatientRoom	1	200	500	TRUE	4
29	PatientRoom	1	200	500	TRUE	4
30	PatientRoom	1	200	500	TRUE	4
31	PatientRoom	1	200	500	TRUE	4
32	PatientRoom	1	200	500	TRUE	4
33	PatientRoom	1	200	500	TRUE	4
34	PatientRoom	1	200	500	TRUE	4
35	PatientRoom	1	200	500	TRUE	4
36	PatientRoom	1	200	500	TRUE	4
37	PatientRoom	1	200	500	TRUE	4
38	PatientRoom	1	200	500	TRUE	4
39	PatientRoom	1	200	500	TRUE	4
40	PatientRoom	1	200	500	TRUE	4
41	PatientRoom	1	200	500	TRUE	4
42	PatientRoom	1	200	500	TRUE	4
43	PatientRoom	1	200	500	TRUE	4
44	PatientRoom	1	200	500	TRUE	4
45	PatientRoom	1	200	500	TRUE	4
46	PatientRoom	1	200	500	TRUE	4
47	PatientRoom	1	200	500	TRUE	4
48	PatientRoom	1	200	500	TRUE	4
49	PatientRoom	1	200	500	TRUE	4

50	PatientRoom	1	200	500	TRUE	4
51	PatientRoom	1	200	500	TRUE	4
52	PatientRoom	1	200	500	TRUE	4
53	PatientRoom	1	200	500	TRUE	4
54	PatientRoom	1	200	500	TRUE	4
55	PatientRoom	1	200	500	TRUE	4
56	PatientRoom	1	200	500	TRUE	4
57	PatientRoom	1	200	500	TRUE	4
58	PatientRoom	1	200	500	TRUE	4
59	PatientRoom	1	200	500	TRUE	4
60	PatientRoom	1	200	500	TRUE	4
61	PatientRoom	1	200	500	TRUE	4
62	PatientRoom	1	200	500	TRUE	4
63	PatientRoom	1	200	500	TRUE	4
64	PatientRoom	1	200	500	TRUE	4
65	NurseStation	4	400	1000	FALSE	4
66	NurseStation	4	400	1000	FALSE	4
67	NurseManager	1	50	200	FALSE	4
68	MedPrep	1	50	200	FALSE	4
69	Pharmacy	1	400	1000	FALSE	4
70	PharmacyAdmin	1	100	200	FALSE	4
71	Charting	1	200	500	FALSE	4
72	SpeechTherapy	1	50	200	FALSE	4
73	SpeechTherapy	1	50	200	FALSE	4
74	SpeechTherapy	1	50	200	FALSE	4
75	SpeechTherapy	1	50	200	FALSE	4
76	CaseManager	2	50	200	FALSE	4
77	MedicalRecords	1	400	1000	FALSE	4
78	RecordsOffice	1	100	200	FALSE	4
79	StaffLounge	1	150	500	FALSE	4
80	FamilyConference	1	200	500	FALSE	4
81	PhysicianOffice	1	50	200	FALSE	4
82	PhysicianOffice	1	50	200	FALSE	4
83	ExamRoom	1	50	200	FALSE	4
84	ExamRoom	1	50	200	FALSE	4
85	ExamRoom	1	50	200	FALSE	4
86	WaitingRoom	1	150	500	FALSE	4
87	Reception	1	50	200	FALSE	4
88	HumanResources	1	100	200	FALSE	4
89	HRFiles	1	100	200	FALSE	4
90	Records	1	500	1000	FALSE	4
91	Conference	1	200	500	FALSE	4
92	Conference	1	200	500	FALSE	4
93	Offices	2	800	1200	FALSE	5
94	Dining	1	1000	5000	FALSE	4
95	Kitchen	1	600	1200	FALSE	4
96	Freezer	1	200	500	FALSE	4
97	TherapyGym	1	2500	6000	TRUE	4
98	ADL	1	600	1000	FALSE	4
99	TherapyRec	1	400	1000	FALSE	4
100	ActivityArea	1	400	1000	FALSE	4
101	ActivityArea	1	400	1000	FALSE	4

102	Storage	2	100	250	FALSE	4
103	Storage	2	100	250	FALSE	4
104	Storage	2	100	250	FALSE	4
105	Storage	2	100	250	FALSE	4
106	Storage	2	100	250	FALSE	4
107	Storage	2	100	250	FALSE	4
108	RestRoom	1	300	1000	FALSE	4
109	CleanSoiled	1	100	250	FALSE	4
110	CleanSoiled	1	100	250	FALSE	4
111	CleanSoiled	1	100	250	FALSE	4
112	CleanSoiled	1	100	250	FALSE	4
113	Maintenance	1	800	1500	FALSE	4
114	Lobby	1	150	500	TRUE	4
115	Desk	1	50	200	FALSE	4