MANAGEMENT OF DISTANCING AND ADJACENCY SPECIFICATIONS IN FACILITY LAYOUT PROBLEMS

by

Richard Arthur Alaimo

A dissertation submitted to the faculty of
The University of North Carolina at Charlotte
in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in
Infrastructure and Environmental Systems

Charlotte

2022

Approved by:

_____
Dr. Churlzu Lim

_____
Dr. Simon Hsiang

_____
Dr. Linquan Bai

_____
Dr. Kyoung Hee Kim

ABSTRACT

RICHARD ARTHUR ALAIMO. MANAGEMENT OF DISTANCING AND ADJACENCY SPECIFICATIONS IN FACILITY LAYOUT PROBLEMS. (Under the direction of DR. CHURLZU LIM)

Facility layout planning is an intriguing and challenging problem that can be addressed from several perspectives, namely from an operational excellence point of view or in consideration of the well-being of occupants and their relative experiences within the facility. These two stances are known to be conflicting in nature since an improvement for one outlook is likely to have a negative impact on the other. At the same time, it might be necessary to address certain requirements and conditions when designing new or renovating existing facilities that can either have a positive or negative effect from an efficiency and/or human factors standpoint, thus revealing the underlying complexity of the facility layout problem (FLP). These inherent challenges make it difficult to apply exact methods for optimizing the layout of a facility, resulting in practitioners to resort to other techniques instead where an optimal design is failed to be guaranteed. There are three avenues that drive this dissertation research that are influenced by the aforementioned issues, including (1) coping with computational complexity and intractability, (2) consideration of infectious diseases in relevance to facility layout planning, and (3) increasing the applicability of exact methods for layout design practitioners (i.e., architects).

For the first research avenue, a special variant of FLP known as the double-row layout problem is considered, where departments are placed along two rows that are separated by a central corridor. By modifying an already existing formulation in the

literature and introducing additional symmetry-breaking constraints, it was found that solution times and optimality gaps were reduced for the most occurrences across 100 randomly generated problem instances of varying size (with respect to the number of departments). This is made possible by incorporating the minimum clearance requirements between departments in the pairwise distance constraints rather than the non-overlapping constraints. Doing so results in the number of binary variables to be reduced by 25% to 50%, respectively, compared to the existing model, thus lessening the overall computational complexity. These findings are key for the other topics in this dissertation.

The second research avenue is in the context of layout design problems during pandemic-induced circumstances, where social distancing and reduced capacity constraints are enforced to reduce the spread of infection. A restaurant layout problem is the primary focus for this avenue in the dissertation (although it can easily be extended to other facility types) to assist restaurant owners in maximizing the number of seats that can be placed inside of the facility when operating under pandemic conditions. Similar to the first research avenue, social distancing requirements are embedded in the pairwise distance constraints. Note that optimizing the layout solely based on the number of seats in the restaurant (aka "naïve approach") may yield inferior performance in practice for a variety of metrics, such as generated revenue, table utilization, rejection rates, etc. (i.e., poor table utilization, parties not being seated due to an unsatisfactory table arrangement). To circumvent this issue, the naïve model is transformed into a two-stage stochastic program with recourse that incorporates scenarios generated from the probability distribution of variously sized parties to maximize the expected revenue. The

assortment and arrangement of seats are determined during the first stage, and the assignment of parties to seats occurs in the second stage. When solving this problem with social distancing and reduced capacity constraints enforced, it was found that the stochastic program produced more revenue for up to 35.71% additional occurrences compared to the naïve approach, as well as improved rejection rates (in 10.5% additional occurrences) and table utilization under a simulated restaurant environment.

The final research avenue is intended to assist architects in the layout design process by generating layout alternatives in consideration of adjacency specifications, the flow of occupants within the facility interior, and building code requirements that are set forth by relevant governing bodies, such as the International Building Code. Existing studies only address a subset of the adjacency specifications, while ignoring proximity and separation requirements that can be enforced to specify maximum and minimum tolerable separation distances between relevant pairs of departments, respectively. In addition, the placement of facility/department accessways have been ignored thus far in the literature, which are subject to the building code requirements that architects must abide by. As a result of these shortcomings, a two-phase optimization framework is proposed for generating block layouts and aisle network configurations for further refining the quality of potential layout design alternatives early in the pre-design phase. The proposed optimization model has a multi-objective function, which is represented as a weighted sum of several objectives. Multiple layouts are generated from various combinations of weights, and are evaluated by an architect to estimate their preferences. The resulting estimated preference can then be analyzed in an effort to find the optimal weights.

DEDICATION

I dedicate this dissertation to the entire Alaimo family, for all past, current, and future generations.

ACKNOWLEDGEMENTS

I would like to thank the Department of Systems Engineering and Engineering Management at the University of North Carolina at Charlotte for granting me the privilege to perform research and teach undergraduate students in the department. I am also thankful for the financial support that has been provided to me so that I could embark on this journey. This experience has enabled me to grow in ways that would have otherwise not been possible, and for that I am eternally grateful.

I would like to extend my thanks to Dr. Churlzu Lim, my advisor and chairman of my dissertation committee, for his on-going guidance and support throughout the duration of this program. I can confidently say that he has given me everything that I could have possibly needed (and so much more) to succeed as his student. Ever since my senior year as an undergraduate student, when we collaborated on our first research endeavor together, he has shown me what it means to be a responsible and hardworking academician. I will miss the days of listening to your lectures and sitting in your office for our research meetings.

I would also like to thank Dr. Simon Hsiang, Professor and Department Chair, for offering his insights over the years to help drive this dissertation research. Collaborating with him has granted me the skill to think about a single problem from multiple perspectives, which I believe is essential for succeeding in all aspects of life. I also wish to thank Dr. Kyoung Hee Kim, Associate Professor and Director of the Integrated Design Research Lab of the School of Architecture, for her contribution in this dissertation to help further bridge the fields of systems engineering and architecture together. I would

also like to thank Dr. Linquan Bai, Assistant Professor, for agreeing to serve as a committee member.

Finally, I would like to thank my friends and family for giving me the support that I needed to thrive in this program. I would not be where I am today if it were not for every single one of you.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

ACO	ant colony optimization
AHP	analytic hierarchy process
ANP	analytic network process
BOFV	best objective function value
CAP	corridor allocation problem
CDC	Center for Disease Control and Prevention
COVID-19	Coronavirus Disease 2019
DEA	data envelopment analysis
DM	decision maker
DRLP	double-row layout problem
DRLP-C	double-row layout problem with clearance
FLP	facility layout problem
GA	genetic algorithm
LP	linear program
MILP	mixed-integer linear program
MIQP	mixed-integer quadratic program
MRLP	multiple-row layout problem
NC DHHS	NC Department of Health and Human Services
NLP	nonlinear programming model
OG	optimality gap
PMP	parties mix problem
PSO	particle swarm optimization

QAP	quadratic assignment problem
SA	simulated annealing
SBC	symmetry-breaking constraint
SDLP	socially distanced layout problem
SRLP	single-row layout problem
TMP	table mix problem
TS	tabu search
TU	totally unimodular

CHAPTER 1: INTRODUCTION

Consider the facility layout problem (FLP) where it is of interest to optimize the utilization of available space inside of a facility. To be more precise, FLP can be described as the optimal placement of differently- or same-sized departments/objects inside of a facility while optimizing one or more objectives. The most popular performance measure for FLP is the minimization of material flow cost, which is calculated by multiplying the flow cost per unit distance by the total distance between departments for each department pair, and then summing all of these terms together. The flow cost can be approximated by referencing the process sequence data from the operation process charts that are available at the facility (Vollman and Buffa, 1966). Other performance measures include robustness, flexibility, space utilization, etc. (Lin and Sharp, 1999). Examples of where this problem is frequently encountered include offices (Chen et al., 2020), hospitals (Halawa et al., 2020), manufacturing plants (Kusiak and Heragu, 1987), etc.

The geometry of a facility can be modelled in two ways, namely discrete or continuous representations. In the discrete setting, the facility is divided into a set of equally- or variously-shaped grids, and departments are assigned to grids within the facility, whereas departments can be placed anywhere in the continuous setting. In general, FLP is known to be NP-hard due to the fact that it is essentially a two-dimensional packing problem, which is also NP-hard since it is an extension to the bin packing problem (Coffman et al., 1997). Dynamic variants of FLP can be considered where each time period in the planning horizon requires a different layout design. This is applicable for the case when the facility needs to be rearranged based on changing demands for certain products or services to operate more efficiently.

Research has been ongoing for FLP since the 1950s and continues to attract researchers from various disciplines to this day. Koopmans and Beckmann (1957) are considered pioneers in the field since they presented one of the first mathematical formulations for FLP, namely the quadratic assignment problem (QAP). Montreuil (1990) proposed another variant of FLP that allows for departments to be placed anywhere within the facility, which is represented as a mixed-integer linear program (MILP) model. Graph theory is also used for solving FLP, but it can yield lower-performing layouts since it is a heuristic approach (Foulds and Robinson, 1978; Montreuil and Ratliff, 1989; Hassan and Hogg, 1991). When applying the graph theory, departments and adjacencies are typically represented as the vertices and edges, respectively, and the associated graph is constructed iteratively by inserting the department with the largest benefit (or smallest cost) into the graph. It is common for the graph to be initialized by strategically selecting a subset of departments (based on some heuristic measure) and creating a planar graph with three departments along its boundary and one department placed in between the others. After the final iteration, the dual of the planar graph can be constructed, and is then referenced for generating the layout design (Foulds and Robinson, 1978).

From these works, additional variants of and model enhancements for FLP have been proposed to increase its applicability and improve the overall computational tractability. Other variants of FLP that stemmed from these initial efforts include the single-row layout problem (SRLP), double-row layout problem (DRLP), multiple-row layout problem (MRLP), unequal-areas FLP, etc. The work done by Meller et al. (1999) and Sherali et al. (2003) are noteworthy examples of efforts that have been made to improve the computational tractability of FLP.

Although FLP was created in consideration of manufacturing facilities, it has been applied to other problem domains since it can be beneficial for other layout problems. Examples include (but not limited to) chemical plants (Park et al., 2011; Latifi et al., 2017; Vázquez-Román et al., 2019), architecture (Wu et al., 2018; Chaillou, 2019; Hu et al., 2020), and construction site planning (Zouein et al., 2002; Kumar and Cheng, 2015; Hammad et al., 2016). The architecture problem domain is one of the primary interests for this dissertation. Methods that are used for assisting the architects during the layout design process are commonly referred to as automated layout generation techniques, which render realistic layouts from conceptual designs.

From an efficiency standpoint, it is intuitive to configure the facility layout in a way that makes sense to the occupants with respect to facility operations to reduce costs and time spent navigating. From a design perspective, this requires the balancing of multiple, and conflicting, objectives that can be measured both quantitatively and qualitatively. The same objectives are not universally applied to all facility types; instead, it ultimately depends on the intended purpose of the facility, as well as the key decision makers (DMs) who are involved in the layout design process. Bate and Robert (2007) suggest that a good facility design consists of three factors, namely the facility performance, engineering, and usability. The facility design is an iterative process that requires input from the DMs to help guide the designers to a solution that is most desired.

1.1 Motivation

The research outlined in this dissertation stems from a variety of shortcomings in the FLP literature. These shortcomings include computational tractability as the problem size increases, consideration of the occupant health during the layout design/redesign process, as well as the need for more practical automated layout

generation for architectural layout design problems. The first shortcoming has been an issue since the introduction of QAP by Koopmans and Beckmann (1957) and is unfortunately still prevalent to this day. As a result, it is worth investigating alternative ways to reduce the computational complexity of FLP by reformulating existing models and generating valid inequality constraints that are intended to improve the quality of the linear programming (LP) relaxation when applying exact methods. The second shortcoming has been ignored for the most part up until the realization of the recent global pandemic, namely Coronavirus Disease 2019 (COVID-19). This situation has required facility designers to rethink how interior spaces are arranged to minimize the spread of infection while maximizing the number of occupants that can be engaged in the system simultaneously in a safe manner. The third shortcoming is an issue in the architectural design community. Since the existing automated layout generation tools reflect only limited aspects of the real facilities, it is likely to generate suboptimal layout design alternatives incorporating only some architectural requirements. Thus, there is a sense of necessity to present an alternative method that can increase the applicability of automated layout generation for architects.

The first research effort to be addressed is in the context of DRLP, which seeks to find an optimal arrangement of departments along both sides of a central corridor to minimize the total material flow cost or another performance metric of interest. In particular, the case where minimum clearance requirements between departments are enforced when they are assigned to the same side of the corridor is considered. It is observed that the existing models (Chung and Tanchoco, 2010; Zhang and Murray, 2012) contain an excess number of binary decision variables, which may increase the overall computational effort to solve the problem. To address

this issue, two MILP formulations are proposed with the motivation that using fewer binary variables compared to existing formulations in the literature helps reduce the solution time. This is accomplished by modifying the representation of the clearance/separation requirements in the model constraints. Noting the NP-hardness of the problem, symmetry-breaking constraints are also introduced to further alleviate the computational burden.

The second research effort considers the impact of facility layout planning in the context of COVID-19. Businesses and corporations across a wide variety of industries initially struggled with the impact of COVID-19 during 2020 since they had to rearrange their facilities due to various policies that were enforced by state and local governments to reduce the risk of infection. Because of these difficulties and the lack of prior experiences, as well as applicable studies in the FLP literature that can rearrange the facility in consideration of COVID-19, many businesses were forced to declare bankruptcy since they could not modify their facilities accordingly. For example, in the restaurant industry, 110,000 restaurants were reported to be temporarily or permanently closed during 2020 because of the pandemic, which resulted in restaurant industry sales to plummet by $240 billion (National Restaurant Association, 2021). Furthermore, most restaurant owners had to lay off 80% of their staff to reduce operational costs since average facilities were operating at between 10%-20% of the maximum potential (Dube et al., 2021). This research focuses mainly on restaurant layout planning, where it is desired to maximize the number of seats that can be occupied inside of the facility when enforcing social distancing and reduced capacity constraints. The novelty of this work in the FLP community is two-fold; first, social distancing, which is to keep the minimum separation distance that is required between individuals or groups of individuals during a pandemic event, is addressed in

designing the layout of the facility. In addition, reduced building capacity constraints are included where it is necessary for not only the placement of seating areas within a given space to be addressed during the optimization, but also which ones should be made available for customers to use. Unlike conventional FLP, this problem poses a variant of the two-dimensional knapsack problem as not all objects may fit in the facility due to the social distancing and reduced capacity constraints.

Note that solely maximizing the number of seats inside the restaurant without any consideration of customer demand (in the context of party sizes) may result in parties to be rejected upon their arrival due to an insufficient number of seating areas, or inefficient utilization rates of the available seating areas (i.e., large sized seating areas being assigned to smaller sized parties). To address this issue, a two-stage stochastic program with recourse is proposed, where the number of variously sized seating areas and their respective placement in the restaurant is considered in the first stage, and the assignment of parties to seating areas with respect to customer arrival rates and party size probabilities is addressed in the second stage. Both problem settings are important for restaurant owners for maximizing the revenue, and are commonly referred to as table mix and parties mix problems (TMP and PMP, respectively). Existing restaurant revenue management related studies typically address TMP and PMP separately from one another, so it is of interest to propose a modelling framework that solves both problems simultaneously. It is worth mentioning that this approach can be generalized for restaurant owners operating under normal conditions to optimize their existing layout, as well as for assisting new restaurant owners during the layout planning process.

The third research effort is intended to assist architects in the layout design process by generating layout alternatives in consideration of adjacency specifications

in an automated fashion. As noted by Arnolds and Nickel (2015), layout planners typically apply manual and rule of thumb approximations for developing and recommending layout designs. When developing layout designs, one important factor considered is the adjacency of each pair of departments. Adjacency specifications consist of three components, namely (1) adjacency, (2) proximity, and (3) separation requirements, where each requirement may reflect mechanical restrictions, logical constraints, and DM's preferences. A large portion of the literature in automated layout generation only consider the adjacency requirements and ignore the proximity and separation requirements. In addition, existing MILP formulations require a large number of binary variables for enforcing the adjacency requirements, which negatively affects the solution time. To address the shortcomings of the existing automated layout generation techniques, an MILP model that simultaneously accommodates adjacency, proximity and separation requirements is proposed for producing block layouts. The proposed model does not require binary variables for representing the adjacency requirements between departments, and hence, the increase of computational efforts can be insignificant.

As an extension to generating block layouts for the architect, it is also of interest to address the flow of occupants inside of the facility after a feasible layout is generated. This is accomplished by representing the boundaries of departments as potential aisle segments, and the objectives are to (1) select which boundaries to convert into aisle segments and (2) where to configure the department/facility doorways with respect to multiple criteria, such as the length of the aisle network, travel distances between departments and facility accessways, etc. Addressing multiple criteria simultaneously in this setting results in a complex multi-objective

optimization problem that requires tradeoff and compromise for finding high-quality solutions.

Some of the existing approaches attempt to accommodate building code requirements when generating the aisle network for new facilities (Peng et al., 2016; Li and Hua, 2019; Gao et al., 2020). What makes this approach distinct from what is in the literature is that the facility entrance and exit locations are represented as decision variables, and their configuration along the facility boundary are optimized with respect to requirements set forth by International Building Code (2021). These requirements include a minimum separation distance between exit points, maximum allowable travel distances from the departments to accessways, etc. In addition, an aisle configuration is introduced that can potentially improve the facility resilience by enforcing unidirectional flows along the aisle segments during pandemic events (such as COVID-19). Enforcing unidirectional flows along designated walking paths was found to be helpful in reducing the spread of infection between healthcare workers in previous studies (Lenaghan and Schwedhelm, 2015; Zimring et al., 2018; Wong, 2019). The combination of these two models (block layout and aisle generation optimization) yields a two-phase framework for further refining the quality of potential layout design alternatives early in the pre-design phase for new facilities.

The underlying theme of this dissertation is the concept of clearances and how they can be applied across different problem settings in consideration of the shortcomings in the FLP literature that were discussed earlier. In summary, the cumulation of these research efforts contributes to the FLP domain in three ways. First, the idea of building compact MILP formulations for DRLP presents a way to represent general types of FLPs with clearance requirements as more computationally efficient formulations. Second, the generation of socially distanced layouts can lay the

foundation for a new set of investigations that seek for measures to cope with a pandemic in existing facilities as well as design schemes to make facilities more resilient to a similar event in the future. Also, a unified modelling framework that optimizes the seating configuration inside of a restaurant with respect to customer demand is invaluable to restaurant owners for improving business operations. Third, incorporating the refined adjacency requirements can benefit the architecture practitioners by providing more realistic layout design options compared to conventional approaches with rather abstract representations cannot offer. Furthermore, the resulting layouts from the two-phase approach establish the norm for functional performance and serve as benchmark designs when architects attempt to create their own designs.

1.2 Organization of the Dissertation

The remainder of this dissertation is structured as follows. Chapter 2 provides a literature review on general FLP-related problems with emphasis on recent development in the literature. Generalized problem formulations are presented for FLP under discrete and continuous representations, and model variants to FLP are also discussed. Note that more thorough literature reviews will be provided for each specific research effort in its own chapter.

Chapter 3 addresses DRLP where compact MILP formulations consisting of fewer binary decision variables are presented and compared with existing models by performing a computational study. Chapter 4 includes the socially distanced variant of FLP in the context of restaurant planning in consideration of pandemic events and the two-stage stochastic programming model for solving TMP and PMP simultaneously. A few instances of restaurant layout problems were found online and used as input for the proposed models to demonstrate their respective efficacies. Following this, an

experimental design was created for assessing the quality of layout solutions that were generated under a simulated restaurant environment. Chapter 5 considers automated layout generation in the architectural community, and two MILP models are proposed. A case study is presented to show the effectiveness of the two-phase method for generating layout designs of new facilities. Finally, Chapter 6 will briefly describe the contents of this dissertation and suggest potential avenues for future research.

CHAPTER 2: LITERATURE REVIEW

The conventional FLP is to find an optimal arrangement of non-overlapping departments within the interior of a facility. Depending on its intended use, the arrangement of departments can be restricted by certain shapes, or flexible as long as departments are placed within the facility (as noted in Section 2.1). Earlier FLP models considered a finite number of candidate locations when placing departments. Such restrictions were later relaxed so that departments can be placed anywhere in the facility (Section 2.2). For certain problem instances, it may be required for departments to be separated by a minimum distance due to operational constraints. The identification of such relationships is critical before finding a solution to FLP since it might have a significant impact on the overall efficiency of the facility (Section 2.3).

Numerous FLP studies have investigated various solution approaches over more than six decades under diverse assumptions including exact and heuristic methods in conjunction with innovative modelling strategies (Section 2.4). The majority of the FLP literature minimizes the material flow cost. However, incorporating other quantitative, as well as qualitative, criteria has gained increasing traction recently (Section 2.5). New criteria have emerged from the recent global pandemic and play an essential role for improving facility resilience and reducing the vulnerability of the layout configuration to infectious diseases (Section 2.5.1). The FLP has been applied to other research disciplines, but its presence in the architecture community has shown to be a promising research direction for this dissertation (Section 2.6).

Before proceeding to the reviews, it would be useful to distinguish FLPs under the static and dynamic settings as they are frequently referred to in what follows. The

former assumes that the same layout is used regardless of the time-period, whereas the latter requires a new layout for each time-period over the scope of the planning-horizon. Static layout problems are more common in the literature since dynamic layout problems become more computationally intractable as the planning horizon length increases. In addition, it can be costly to modify the layout design in each period if rearrangement costs are included. As a result, researchers have proposed the so-called robust layout design where it is intended to minimize the material flow cost across all time-periods with respect to a single layout design alternative (Pillai, Hunagund, and Krishnan, 2011; Tosun, Dokeroglu, and Cosar, 2013; Peng et al., 2018).

2.1 Material handling configurations

The performance of the facility depends on how the material handling configuration is specified beforehand, and it is likely that certain configurations result in better performance than others for a given problem instance. The most common material handling configurations in the literature can be categorized as *row layouts* (single, double, or multiple), *loop layouts*, and *open-field layouts*.

2.1.1 Row and loop layouts

Row layouts require departments to be arranged in one or more rows, and are commonly used in modern flexible manufacturing systems (Tubaileh and Siam, 2017). Simmons (1969) introduced the SRLP where departments are placed across a horizontal line, which is very similar to the well-known linear arrangement problem in the literature (Adolphson and Hu, 1973). It is noteworthy that, despite its simplicity, SRLP is known to be NP-hard (Garey, Johnson and Stockmeyer, 1976). The loop layout was introduced by Afentakis (1989) and considers the placement of departments to a set of candidate locations such that materials are transported in a

single direction. Kaku and Rachamadugu (1992) modelled the loop layout based on the QAP formulation of Koopmans and Beckmann (1957), but other modelling techniques and extensions have been applied in more recent research efforts (Asef-Vaziri, Jahandideh, and Modarres, 2017; Asef-Vaziri and Kazemi, 2018; Kang, Kim, and Chae, 2018; Kim and Chae, 2019, Ahmadi-Javid and Ardestani-Jaafari, 2020).

Heragu and Kusiak (1988) introduced the DRLP for placing departments along two rows that are separated by a central corridor, which can be used for the transportation of materials or the traffic of occupants. Chung and Tanchoco (2010) provided the first MILP formulation for DRLP, which was found to be incorrect and modified accordingly in the work by Zhang and Murray (2012). Additional constraints were introduced by extending the application of DRLP to other problem settings, such as requiring no empty space between departments in each row (corridor allocation problem (CAP); Amaral, 2012), and assigning departments to either row a priori and optimizing their relative placement in the row they belong to (parallel-row ordering problem; Amaral, 2013a). Heragu and Kusiak (1988) also introduced the MRLP where departments are assigned to three or more rows that are all parallel to each other. It is noted by Anjos and Vieira (2017) that the model of Zhang and Murray (2012) can easily be extended to multiple rows if desired.

2.1.2 Open-field layouts

Unlike row and loop layout problems, where the placement of departments is restricted by the underlying structure, open-field layout problems allow departments to be placed anywhere within the boundaries of the facility. This characteristic of open-field layouts makes it the most flexible compared to other layout types. Montreuil (1990) gave the first MILP formulation of the open-field layout, and researchers since then have been addressing this problem from a variety of different

perspectives to improve its applicability. One disadvantage of the open-field layout is that it can be difficult to address the aisle structure and department placement simultaneously when modelling the problem. This has forced researchers to approximate the material flow cost by measuring the center-to-center distance between departments and multiplying the cost factor to these components. Recent efforts have been made to consider the aisle structure for the open-field layout using MILP, but the problem size grows very quickly thus making it less applicable for larger-sized problem instances (Lausnitzer and Lasch, 2019; Pourvaziri, Pierrerval, and Marian, 2021).

2.2 Problem formulation

As mentioned in the previous chapter, Koopmans and Beckmann (1957) and Montreuil (1990) were the first ones to formulate FLP using discrete and continuous representations of FLP, respectively. The discrete representation of FLP considers the assignment of departments to a set of candidate locations, whereas the continuous representation considers the placement of rectangular-shaped departments anywhere within the facility. Although they are modelled differently, both representations share the same objective of minimizing the material flow cost. The generalized problem formulations for both representations of FLP are presented in the next two subsections.

2.2.1 Discrete representation of FLP

Suppose that there are n departments and m candidate locations in the facility. It is assumed that $n \leq m$ to guarantee a feasible assignment of departments to locations. Let N and M denote the set of departments and locations, respectively. For simplicity (and without loss of generality), consider the case where $n = m$. The flow cost per unit distance between departments i and k and the distance between locations

j and l are represented as f_{ik} and d_{jl}, respectively. Let x_{ij} be a binary decision variable equal to 1 if department i is assigned to location j, and 0 otherwise.

Constraints for the discrete representation of FLP include: (1) each department is assigned to one location, and (2) each location is assigned to one department. From this problem description, the following nonlinear programming model (NLP) can be formulated for the discrete version of FLP.

Minimize $$\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n}\sum_{l=1}^{n} f_{ik}d_{jl}x_{ik}x_{jl}$$

subject to

$$\sum_{j=1}^{n} x_{ij} = 1 \quad \forall i \in N$$

$$\sum_{i=1}^{n} x_{ij} = 1 \quad \forall j \in M$$

$$x_{ij} \in \{0,1\} \quad \forall i \in N, j \in M.$$

Note that the nonlinear term in the objective function (which is the product of two binary variables) can be linearized by applying a standard linearization technique (Glover and Woolsey, 1974).

2.2.2 Continuous representation of FLP

Suppose that there are n departments to be placed inside of the facility. Let c_i^s and d_{ij}^s represent the center point of department i along direction s and the center-to-center distance between departments i and j along direction s, respectively. Note that the continuous representation of FLP can be addressed in one-, two-, or three-dimensions. In a two-dimensional case, the length of the facility along direction $s \in \{x, y\}$ is denoted by L^s, and the length and the width of department i are denoted by a pair (l_i^x, l_i^y). Some authors prefer to let l_i^x and l_i^y represent the half-length and the

half-width, respectively, resulting in a similar model formulation (Sherali et al., 2003). It is required for each department with variable dimensions to specify lower- and upper-bounds on l_i^x and l_i^y in consideration of the desired aspect ratio, which is evaluated for each department by taking the maximum of the length and width components and dividing it by the minimum of the same two components. In consideration of this, let lb_i^s and ub_i^s represent the lower- and upper-bounds of department i along direction s, respectively, and let $\alpha_i \geq 1$ represent the desired aspect ratio of department i. For the case where the dimensions of any department are known a priori, this results in $lb_i^s = ub_i^s$ for $s \in \{x, y\}$. Using a similar notation for the flow cost per unit distance as in the discrete representation of FLP, let f_{ij} denote this cost between departments i and j.

Constraints for the continuous representation of FLP include: (1) departmental area constraints (for departments with variable dimensions), (2) overlap prevention (with optional department separation conditions), (3) calculation of the pairwise distance, (4) placing departments within the boundaries of the facility, and (5) lower- and upper-bounds for the department dimensions. From this problem description, the following NLP can be formulated for the continuous version of FLP.

Minimize $\sum_{i=1}^{n-1} \sum_{j=i+1}^{n} f_{ij}(d_{ij}^x + d_{ij}^y)$

subject to

departmental area constraints,

overlap prevention,

$d_{ij}^s = \left|c_i^s - c_j^s\right| \qquad \forall i, j \in \{1, \dots, n\}, s \in \{x, y\}, i < j$

$\frac{l_i^s}{2} \leq c_i^s \leq L^s - \frac{l_i^s}{2} \quad \forall i \in \{1, \dots, n\}, s \in \{x, y\}$

$lb_i^s \leq l_i^s \leq ub_i^s \qquad \forall i \in \{1, \dots, n\}, s \in \{x, y\}.$

It is straightforward to convert the nonlinear pairwise-distance constraints into a series of inequalities to represent the model as an MILP instead. Although the model itself is not overly complex, it tends to perform poorly from a computational efficiency standpoint as the number of departments increases due to the number of binary variables and disjunctions that are required for the overlap prevention constraints.

2.3 Clearance requirements

For some instances of FLP, it may be required for certain pairs of departments to be separated by a minimum distance because of some operational condition(s) at the facility (such as ventilation or vibration effects). This minimum distance is commonly referred to as a minimum clearance requirement. Clearance requirements are determined by the layout planner before applying the optimization with respect to several factors, such as production volume and routing of materials, processing times, type of material handling equipment, etc. (Solimanpur et al., 2005). Therefore, properly enforcing the clearances during the design phase has a direct impact on the facility operations.

Typically, clearance requirements are specified for all pairs of departments because it is possible that the minimum separation distance may vary for the problem instance (i.e., explicit clearances). If it is unnecessary for a pair of departments to be separated in a problem instance, then the minimum separation distance is simply set to zero. Authors have attempted to simplify the representation of clearances using what are known as implicit clearances, where the interaction effect between departments regarding a minimum separation is assumed to be negligible (Amaral, 2013b). The clearance is instead embedded in the department dimensions, and the optional separation conditions from the problem formulation in Section 2.2 are ignored. Zuo et al.(2016) proposed what are known as additional clearances to

consider the situation where extra space is required for activities like maintenance or storage.

2.4 Solution approaches

A variety of solution approaches have been applied for solving classic and extended variations of FLP. Some of the earliest approaches that were used in practice include methods like CRAFT, ALDEP, and Spiral, which were designed by Armour and Buffa (1963), Seehof et al. (1966), and Goetschalckx (1992), respectively. CRAFT is an improvement-based approach that iteratively improves the performance of a layout by rearranging departments throughout the facility. ALDEP selects departments for placement during each iteration based on a minimum closeness score based on the departments that were placed earlier on during the procedure. Spiral is a software package that applies different algorithms in the FLP literature for generating graph- or block-based layout designs.

For this literature review, research efforts from 2010 to 2021 were investigated to assess the state that FLP research is currently in and to obtain a better idea as to where it is heading. Metaheuristic algorithms were and still are the most prevalent solution approaches for FLP throughout this time frame, but some research efforts have been made for finding optimal solutions to FLP via exact methods.

2.4.1 Exact methods

Díaz-Ovalle et al. (2010) propose a technique to optimize the placement of new departments in a region where existing departments are already located. The new departments have a risk of toxic release associated with them, so it is desired to minimize the worst-case scenario of an accident occurring at the facility via the convex-hull method in mixed-integer NLP. Meller et al. (2010) present a bottom-up approach for FLP that requires the layout planner to specify the arrangement of units

within each work cell a priori, and the model will determine the optimal orientation of the work cells to minimize the material flow cost. Jankovits et al. (2011) apply semidefinite relaxation and convex relaxation techniques for solving FLP.

Park et al. (2011) address safety factors in a chemical plant to reduce the impact of a physical explosion within the facility. Safety distances are implemented using a consequence analysis in which the likelihood of a disastrous event occurring for each department is measured. Bernardi and Anjos (2013) propose a two-phase procedure for the multi-floor FLP where the first phase assigns departments to each floor and the second phase optimizes the layout of each floor independently from the others. Bukchin and Tzur (2014) present an MILP that allows rectangular and L/T shaped departments to be placed inside of the facility.

Neghabi et al. (2014) address facility robustness by incorporating the so-called departmental length and width coefficient deviations. This allows for departments with uncertain dimensions to be considered for FLP using pre-specified lower and upper bounds on the dimensions instead. Ghassemi and Neghabi (2015) propose an alternative method for addressing adjacency in FLP. Traditionally, adjacency between departments was represented using a binary index, where 0 and 1 values indicate not adjacent and adjacent, respectively. The authors define two departments being adjacent to one another if they are within a pre-determined distance by using their center points as the reference point. They use a continuous variable to measure the respective degree of adjacency (bounded between 0 and 1) between a pair of departments.

Ahmadi and Jokar (2016) present a multi-phase optimization framework that addresses single- and multi-floor variants of FLP using a similar approach as Bernardi and Anjos (2013), but apply exact methods for finding an optimal solution or near-

optimal solutions. Anjos and Vieira (2016) propose a two-phase optimization process for generating a facility layout design. In the first phase, a barrier penalty method is used to determine the relative orientation of departments, which are then used as inputs to the second phase. Since the objective function is non-convex, local optimal solutions are referenced for determining relative orientation. In the second phase, non-overlapping constraints are then generated based on the results computed from phase one. Chae and Regan (2016) address departments of two types, namely those with flexible and inflexible dimensions, and propose constraints to determine the orientation of the latter type.

Latifi et al. (2017) introduce an approach that takes economic and safety aspects into consideration simultaneously for a process plant layout problem. Toxic release and occurrence of domino event constraints, which consider blast waves, and pool, jet and flash fires, are incorporated into the model and their disastrous impacts are approximated using probability functions. Che et al. (2017) consider the multi-floor FLP where each floor consists of a set of rooms with a given area and the objective is to assign departments to one or more rooms while minimizing material floor costs and wasted space. The authors linearize the original model and apply an epsilon-constraint method to optimize both objectives simultaneously.

Ghassemi and Neghabi (2018) extend the work of Ghassemi and Neghabi (2015) to classify departments as being adjacent if they share a minimum common boundary length along any direction. The benefit of this formulation compared to their previous work is that fewer binary variables are required. In Xie et al. (2018), the authors were interested in linearizing the Euclidean distance between pairs of departments since there are cases where this distance metric is more desired than the rectilinear distance that is traditionally used. This linearization allows for a pre-

determined margin of error specified by the DM to be satisfied in measuring the Euclidean distance prior to the optimization. Wu et al. (2018) formulate a mixed-integer quadratic program (MIQP) that allows for irregularly shaped departments to be considered. The building is decomposed into a set of sub-domains consisting of multiple rectangles that serve a similar functional purpose. The layouts of the sub-domains are optimized iteratively to remove irregularities from the layout.

Vázquez-Román et al. (2019) propose an NLP model consisting of fewer overlapping constraints and binary variables for process layout problems. Lausnitzer and Lasch (2019) formulated an MILP that optimizes the placement of departments, input and output locations, as well as material flow paths using a network design approach. Travel distances are measured along the aisles that are generated in between departments to account for the situation where it is not feasible to travel directly through the departments. Pourvaziri et al. (2021) address FLP in consideration of department placement and aisle generation simultaneously. It is assumed that there is one entrance and exit to the facility, and departments are assigned to different levels with respect to department input/output points and aisle accessibility constraints.

2.4.2 Genetic algorithm

The genetic algorithm (GA) represents the evolution of a species over time with respect to the chromosomes that are passed down from parents to their children when producing offspring. A series of operations are applied to each child to diversify the population of individuals, including *selection*, *evaluation*, *crossover*, and *mutation*. This diversity is important because it allows for a wide variety of individuals to be produced, thus allowing for the species to evolve for future generations. Each individual belonging to the population is identified as a solution

and the objective of the GA is to iteratively evolve the initial population to find the best observed solution across all generations.

Caputo et al. (2015) consider risk assessment as an objective in FLP to minimize operational costs and economic loss from accidents that may occur. Economic losses include costs associated with damage to equipment and fatalities from faulty equipment. The GA is solved using a performance-based approach where probabilities associated with accidents are dependent on physical effects between departments, as well as their distances apart. Ficko and Palcic (2013) propose a procedure that converts the facility layout into a mesh of equilateral triangles where nodes represent the departments and edges represent material flow paths.

Jiang and Nee (2013) developed an FLP system that uses augmented reality and an analytical hierarchy process-based GA for improving existing layouts. Augmented reality is used to collect data from the existing layout and the departments that are placed within it. Paes et al. (2017) apply a greedy approach for placing departments within the facility depending on the flexibility of departmental requirements. Empty maximal spaces are created within the boundaries of the facility and departments are placed as close as possible to the center of the maximal empty space it belongs to. Rodrigues et al. (2013) formulate a GA for solving the multi-floor FLP. A stochastic hill climbing search is applied to randomly generate solutions during each iteration.

Sadrzadeh (2012) proposes a GA for the multi-line FLP. Initial solutions are generated and grouped together based on how strong the inter-departmental relationships are, which is measured as the product of material flow, total cost, and distance for pairs of departments. Aiello et al. (2012) optimize material flow costs, aspect ratio, adjacency, and distance for the unequal-areas FLP by applying a multi-

objective GA, where the unequal-areas FLP refers to an FLP consisting of departments with variable dimensions. The facility is divided into horizontal and vertical cuts, forming a series of blocks that are used to represent the departments to be placed. Aiello et al. (2013) use elite mechanisms to diversify the offspring to further explore the solution space. Recursive mutation guarantees that all individuals in the population are unique by changing each chromosome until this condition is satisfied.

Emami and Nookadabi (2013) consider the multi-objective FLP and applied a nondominated sorting GA. Chromosomes of the offspring are generated using a single point operator between parents along the boundary where two different consecutive time periods exist in the encoding. Departments belonging to the same time period are then randomly swapped as a mutation operator. Lenin et al. (2013) formulate a GA for the multi-objective SRLP. Leno et al. (2013) consider the multi-objective FLP such that the inter-cell layout and material handling system are considered simultaneously. An elitist strategy is applied for the GA to ensure that the set of best solutions are preserved during each generation.

Gonçalves and Resende (2015) formulate a biased random-key GA and LP procedure for the unequal-areas FLP. A hybrid approach is used where the GA determines the order in which departments are placed and their respective sizes, followed by a placement strategy to place the departments within the facility. Kulturel-Konak and Konak (2013) propose a GA and LP procedure for the unequal-areas FLP. The encoding scheme for each individual is structured as a matrix that contains the centroids along the x- and y-axis of each department and its aspect ratio.

Datta et al. (2011) formulate a GA that converts SRLP to an unconstrained optimization problem. Random and rule-based techniques, known as worst-pair

together, flow-based permutation, and length-based permutation, are used for initializing feasible solutions. Mazinani et al. (2013) formulate a GA for the dynamic FLP where integers in the encoding scheme are used to represent which bay the department is assigned to and the decimal represents the order in which it is placed for each time period. Yang et al. (2011) formulate a GA for the dynamic FLP. Sequences of orders, or jobs, are defined that require a particular sequence of departments within the facility to reduce material flow costs.

Izui et al. (2013) consider the facility layout of a robotic cellular manufacturing system where the total operation time, layout area, and overall feasibility for robotic units to be placed are optimized. A non-dominated sorting GA is used for selecting the non-dominated solutions, which are then used for producing offspring for the next generation. Khaksar-Haghani et al. (2013) consider the multi-floor FLP within a cellular manufacturing setting, where it is necessary to optimize cell formation, group layout, and scheduling. A parameter ratio representing how far the child is from the better parent is used for producing offspring, which is followed by random mutation. Kia et al. (2014) formulate the multi-floor FLP as an MILP using a cell-based structure and applied a GA to solve it. Krishnan et al. (2012) apply a group technology strategy using binary part-machine incidence matrices, which represent the relationship between groups of departments and parts for cellular manufacturing systems.

Palomo-Romero et al. (2017) parallelize the GA procedure by randomly generating and separating the initial population into smaller sized subpopulations. A tournament selection method is used for selecting parents to produce offspring by selecting random pairs of individuals and choosing the one that has a better objective function value. Tosun et al. (2013) use a master-slave node representation for

parallelizing the GA. Peng et. al (2018) propose a GA for stochastic dynamic FLP to find a robust layout that minimizes the flow cost across the entire planning horizon.

2.4.3 Simulated annealing

Simulated annealing (SA) resembles the process of annealing a material by alternatingly applying hot and cool temperatures. This process is repeated until the material reaches a stable state such that it can be used for its intended purpose. A temperature parameter is applied to guide the solution process based on changes between the objective function values that are recorded during the procedure. When it is observed that this difference is small, then it can be assumed that the solution has achieved its "ground state," resembling a solution to the problem.

Kulturel-Konak and Konak (2015) propose an SA and LP procedure for the cyclic FLP. LP is used for generating solutions to the problem, which is followed by SA to explore the neighborhood space. The percentage difference between the objective function values is used to scale the acceptance probability when determining whether to accept a neighborhood solution. Xiao et al. (2013) create an SA algorithm using an interconnected zoning algorithm for arranging departments within the facility. Wang et al. (2015) use a mixed encoding scheme where the sequence of departments and their locations for each period in the planning horizon are included. Mathematical programming is used to determine the exact location for all departments and find an optimal solution that is better than the one from the SA by fixing the sequence of departments along both rows.

Navidi et al. (2012) apply the game theory to represent the decision-making process between facility layout and material handling systems designers as a duopoly Bertand competition game. Since these objectives are conflicting, it is obvious that the facility layout and material handling system designers have different strategies to

optimize their goals. Matai (2015) formulate an SA algorithm for the multi-objective FLP. Neighborhood solutions are generated by assigning departments to locations that share the same index and evaluating each one in order.

Matai et al. (2013) normalize the objective function prior to the procedure for quantitative and qualitative measures to ensure that no objectives are more dominant than the others. Palubeckis (2015) consider a special case of SRLP where the departments are all separated by an equal distance. The SA algorithm allows for pairs of departments to be swapped or inserted in the encoding scheme. The author designs two procedures to reduce the computational complexity of comparing best-known and new solutions for the multi-start SA algorithm, as well as for exploring the neighborhood space of current solutions.

Pourvaziri and Pierreval (2017) apply the open queuing network theory for representing processing and arrival rates of products during the facility layout design process and use SA to solve the multi-objective dynamic FLP. Cloud theory is used for handling the multi-objective function by measuring the uncertainties of qualitative measures using their level of membership in the fuzzy set. Bozer and Wang (2012) transformed the unequal-areas FLP into two graphs and applied the SA algorithm for determining the relative orientation and location of departments.

Pillai et al. (2011) structure their SA algorithm to solve the dynamic FLP under two situations. It is assumed in the first situation that rearrangement costs and time are low when making changes to the facility layout. In the second situation, it is assumed that rearrangement costs and time are extremely high so there is less flexibility in making changes to the facility layout. Moslemipour and Lee (2012) incorporate the DM's confidence level into their SA algorithm for the dynamic FLP, which represents their level of uncertainty of total demand for each time within the

planning horizon. Sahin et al. (2010) incorporate budget constraints into their SA algorithm for the dynamic FLP.

2.4.4 Tabu search

Tabu search (TS) is a local neighborhood search procedure for analyzing other solutions that might exist nearby a recently found solution. Good solutions that are found are stored in memory to avoid encountering the same solution in a future iteration. Solutions are constantly compared and added/removed from the memory until the best possible solution is found.

Bozorgi et al. (2015) use data envelopment analysis for analyzing the most efficient layout designs for the dynamic FLP and apply TS for exploring the solution space. McKendall Jr. and Hakobyan (2016) apply TS to solve the dynamic FLP with unequal-areas departments. A boundary search heuristic generates initial layouts for the facility and are improved using TS by exchanging department locations to reduce material handling costs. McKendall Jr and Liu (2012) present two TS algorithms for the dynamic FLP. One of the algorithms incorporates a dynamic tenure length based on the percentage difference from the best-known solution found and recently obtained solutions, and the other algorithm is known as a probabilistic TS where probabilities of assigning departments to locations are computed and ranked.

Kothari and Ghosh (2013) formulate two TS algorithms for SRLP. Initial solutions are generated by sorting the departments in non-decreasing order based on their lengths. In the first TS algorithm, solutions are generated by swapping the positions of adjacent departments. In the second TS algorithm, solutions are generated by placing departments elsewhere in the facility. Palubeckis (2012) formulates a branch-and-bound algorithm for SRLP using the Gilmore-Lawler bounding technique, which is used for generating lower bounds on the optimal solution for QAP (Gilmore,

1962; Lawler, 1963). TS is applied to reduce the size of the solution space and for performing a dominance test to compare different layout designs. Samarghandi and Eshghi (2010) considered a special case of SRLP where the material flow cost between all departments is equal. The authors prove that an optimal solution can be found by sorting the departments in non-decreasing order and partitioning them into three different sets.

Kulturel-Konak (2012) proposes a probabilistic TS and LP procedure for the unequal-areas FLP. A linear program is used for determining the shape and location of departments in the facility. After finding a solution, the probabilistic TS is applied using insert and swap operators. Zuo et al. (2014) formulate a procedure that includes TS and LP for the multi-objective DRLP. TS is used for generating department sequences along each row and LP determines the location of each department. Scholz et al. (2010) use slicing trees and structures for placing departments and determining the directions of the guillotine cuts for dividing the facility into different areas. A slicing tree is a binary tree which shows how departments are partitioned with respect to the cuts that are recursively made in the slicing structure. Each internal node in the tree represents the way a rectangular cut is reflected in the layout design. Cuts are made either vertically or horizontally, and the leaf nodes contain the departmental indices to represent which cut partition each department is assigned to (Tam and Chan, 1998).

2.4.5 Ant colony optimization

Ant colony optimization (ACO) simulates the behavior of ants when they are looking for new food sources. Ants work collaboratively by releasing pheromones to inform the other ants about their location and where to continue to search. After some

time, the ants will no longer continue searching and gather to the best-found food source.

Komarudin (2010) considers the unequal-areas FLP and uses a slicing tree representation for dividing the facility into horizontal and vertical segments. Pheromone information from the ants in the system are used to determine the slicing sequences and orientation types, which then generate a dummy layout. An adaptive penalty function is applied to restrict the search process within the feasible region. Chen (2013) proposes a new representation of the encoding scheme for the ACO algorithm. Binary encoding is used to convert the sequence of locations for departments using the hexadecimal numbering system. Using this approach, at most 256 departments can be considered. Kulturel-Konak and Konak (2011b) apply ACO for the unequal-areas FLP. Solutions are constructed by randomly assigning departments to cells based on their level of desirability.

2.4.6 Particle swarm optimization

Particle swarm optimization (PSO) simulates the behavior of birds who are flying together to find new sources. When a potential food source is found, the bird adjusts its direction and speed so that it can investigate the area of interest. Since the birds are working collaboratively, they share information with regard to food sources so that the entire swarm can have food to eat.

Smarghandi et al. (2010) formulate a PSO algorithm for solving SRLP. They define a relationship between the factoradic numbering system and the permutation property of SRLP. Random initial solutions (or particles) and velocities are generated for each particle and are updated throughout the procedure to guide them towards the global optimal solution. Adrian et al. (2015) apply a craziness function that generates new particles when an existing particle might be stuck at a local optimal solution.

Derakhshan and Wong (2017) formulate a PSO algorithm for the dynamic FLP where shapes and dimensions of the departments are known a priori and can be placed anywhere within the facility. Centroids and orientations for the departments are represented as unit intervals during the procedure and are converted to their actual values with respect to the facility following each iteration.

Jolai et al. (2012) formulate a multi-objective PSO to solve the unequal-areas dynamic FLP. Particles within the swarm are compared using a dominance score to update their best-known solution throughout the procedure. Weights are applied to each objective and a roulette wheel technique is used for updating the best-known solution of the swarm by evaluating the overall dominance score for the non-dominated solutions.

2.4.7 Miscellaneous

Abedzadeh et al. (2014) formulate an MILP model with an objective function consisting of multiple fuzzy measures. To solve this model, initial solutions are generated, and their respective neighborhood spaces are explored using the parallel variable neighborhood search. Palubeckis (2017) proposes a variable neighborhood search algorithm for SRLP. Interchange- and insertion-based local search algorithms are presented and combined using the variable neighborhood search framework for efficiently solving large-scale instances.

Altuntas and Selim (2012) create three association rule-based learning approaches for solving the FLP. Departments are assigned to one or more clusters during the procedure based on the relationships that were found between them after analyzing the data. For the situation where more than one cluster exists at the end of the procedure, they are all merged together to produce the facility layout design. Kang and Chae (2017) formulate a harmony search algorithm for the unequal-areas FLP.

Ulutas and Islier (2015) formulate an artificial immune system algorithm for the dynamic FLP. A matrix is used to encode the antibodies where each element in the matrix corresponds to the department's location within the facility. Strings of departments are randomly created initially and concatenated together to represent the antibodies encoding scheme. Ulutas and Kulturel-Konak (2012) consider the unequal-areas FLP and formulate an artificial immune system-based clonal selection algorithm. The authors modify the traditional encoding scheme that is used in the literature by adding the number of departments and their respective dimensions to the antibodies.

Kothari and Ghosh (2014) propose a scatter search algorithm for SRLP. To ensure that a diverse set of solutions are generated, deviation distances between parent permutations are computed by taking the absolute valued sum of differences between the set of department indexes from one solution and the element where the department is located from the encoding scheme of another solution. Niroomand et al. (2015) formulate a migrating birds optimization algorithm for solving the closed loop FLP. The first-fit strategy is used for generating initial solutions where departments are randomly placed starting at the northwest corner of the facility. To reduce the overall amount of empty space, a set of departments are required to be located at the corners of the facility. Vitayasak et al. (2017) formulate a backtracking search algorithm for the stochastic dynamic FLP. Material flow and rearrangement costs are minimized using this algorithm while considering the uncertainty associated with product demand during the planning horizon.

Li et al. (2018) consider FLP in the dynamic setting and optimize the safety, sustainability, efficiency, and cost of the facility. Various human and management factors are integrated in the model to improve health and well-being of employees in

conjunction with achieving organizational goals. An artificial bee colony algorithm is applied for finding a solution. Herrán et al. (2021) apply the variable neighborhood search metaheuristic for the space-free MRLP. The authors modify the procedure so that it can be implemented using parallel-computing, and it is shown that it is beneficial for finding a solution more quickly.

2.4.8 Hybrid algorithms

Kulturel-Konak and Konak (2011a) propose a hybrid binary PSO algorithm with a local search method for the unequal-areas FLP. A logistic function calculates the probability that a department is placed in a given location and is used for generating the order of placement. An adjustment method is applied to the locations in the facility that are not sufficiently wide enough to contain the departments that are assigned to it. Ou-Yang and Utamima (2013) combine estimation of distribution, PSO, and TS algorithms for SRLP. Estimation of distribution and PSO are used in an alternating fashion for generating offspring during the procedure. The estimation of distribution algorithm measures the probability of assigning departments to a given location and then randomly placing them in the facility.

Dokeroglu (2015) proposes a hybrid teaching-learning based TS algorithm for the discrete representation of FLP. Hosseini and Khaled (2014) combine the imperialist competitive, the variable neighborhood search, and the SA algorithms for the dynamic FLP. The imperialist competitive algorithm represents each solution as a country where the best solutions are imperialists. The goal is for one empire to dominate all other empires. To simulate the occurrence of a revolution in each colony, variable neighborhood search is used consisting of three operators to find a better solution. Once the best solution is selected using these two algorithms, SA is used to navigate the neighborhood space once more to find a better solution.

Leno et al. (2016) create a hybrid genetic and SA algorithm. Input and output stations are arranged for each department in order to minimize the material flow cost. Tayal and Singh (2018) formulate a hybrid firefly and chaotic SA algorithm for the multi-objective dynamic FLP. Analytic hierarchy process (AHP) is applied for assigning weights to the objectives, which influence the evaluation of solutions that are generated. Ripon et al. (2013) formulate a hybrid genetic and variable neighborhood search algorithm for the multi-objective unequal-areas FLP. A slicing tree representation is used for the encoding scheme that includes the department placement sequence, slicing sequence and orientation, and the number of departments. Guan and Lin (2016) combine variable neighborhood search and ACO algorithms for SRLP.

García-Hernández et al. (2013) formulate a hybrid genetic and fuzzy system algorithm for the unequal-areas FLP. Initial solutions are randomly generated and grouped together using fuzzy c-means clustering. The best layouts from each cluster are displayed to the DM so they can provide a score. Niching techniques are applied to select the best solutions from each cluster to produce offspring for the next generation. Each solution is clustered again for the next generation, and the same evaluation procedure is repeated until the DM selects their favorite layout design.

Hosseini-Nasab and Emami (2013) propose a hybrid PSO and SA algorithm for the dynamic FLP. To efficiently navigate the solution space, the authors map the set of solutions that represent the facility layout during the planning horizon into a set of factoradic base numbers. Pourvaziri and Naderi (2014) propose a hybrid genetic and SA algorithm to solve the dynamic FLP. The first two populations are generated by relaxing the integer variables in the nonlinear objective function and then

measuring the distribution of chromosomes among these populations depending on the placement of departments within the facility.

Ku et al. (2011) create a hybrid procedure that initially parallelizes the GA into subpopulations. After the best individuals from the subpopulations are found, SA is applied for exploring their solution spaces to determine if there exist any better solutions. Tosun (2015) formulate a hybrid genetic and TS algorithm for the FLP. Normal and robust TS procedures are applied to diversify the population using adaptive and short-term memory, respectively. This allows for them to avoid already explored neighborhood spaces.

Kulturel-Konak (2017) formulates an MILP for a variant of the dynamic FLP where departments are assigned to different zones of the facility. Each zone is created from the intersection of vertical and horizontal bands spanning across the facility where the number of bands are specified a priori. Similar to the flexible-bay structure variant of FLP, an aisle structure is naturally created as a result dividing the facility into zones. A metaheuristic was developed that consists of SA, variable neighborhood search, and exact methods to find a solution.

2.5 Layout evaluation and emerging criteria

The selection of a layout design is an iterative process that requires input from the DM to guide the designers towards a solution that is most desired. Once the designer is finished gathering requirements and generating a set of layout alternatives, DMs are responsible for evaluating them so that a final layout can be selected. The literature is rich with different methodologies for handling the evaluation aspect of FLP, particularly when both quantitative and qualitative criteria are involved. These techniques include analytic hierarchy process (AHP) (Satty, 1994; Foulds et al., 1998; Singh and Singh, 2011; Hadi-Vencheh and Mohamadghasemi, 2013), data

envelopment analysis (DEA) (Yang and Kuo, 2003; Ertay et al., 2006; Tayal et. al, 2020), etc. The output from these and similar approaches is a ranking of each layout design alternative based on its calculated score, and it is common for the top ranked alternative to be selected as the final solution.

Bashiri and Deghan (2010) consider the dynamic FLP and apply AHP and DEA for optimizing cost, adjacency, shape ratio, and flexibility over the span of the planning horizon. Singh and Singh (2011) apply AHP in a three-phased heuristic for performing pairwise comparisons, normalizing the criteria measures, and computing the objective weights of the criteria. Mohamadghasemi and Hadi-Vencheh (2011) use fuzzy set theory and NLP to maximize the minimum score of a given set of layout alternatives during the evaluation stage where the criteria weights are represented as decision variables. Hasan et al. (2012) compare the performance of different facility configurations with respect to agile manufacturing using analytic network process (ANP). Hadi-Vencheh and Mohamadghasemi (2012) develop a framework consisting of the AHP and NLP for evaluating layout alternatives. The NLP model maximizes the score of each layout by optimizing the weight of the criterion. The layouts are considered disjointly during the NLP procedure, thus resulting in different criteria weights each time this step is executed.

Al-Hawari et al. (2014) apply ANP to account for interdependencies that exist between criteria. To demonstrate the efficacy of this approach, the authors perform a sensitivity analysis to display how the ranking of layout alternatives changes. Wang et al. (2016) combine a simple additive weighting scheme, a technique for order of preference by similarity to ideal solution, grey relational analysis, and design of experiments to construct a hybrid decision-making methodology for the layout design problem. The first three multi-criteria decision-making techniques construct an

approximation dataset for the layout evaluation, which are used for constructing a regression model in an effort to represent the DM's evaluation process using design of experiments. Ferjani et al. (2019) take into account the fatigue of workers within the facility during the evaluation process. Contributors to fatigue in their approach include noise, posture, heavy loads, vibrations, and task durations.

Sagnak et al. (2018) consider 112 criteria measures for the layout design problem by applying the fuzzy decision-making trial and evaluation laboratory technique. Lin and Wang (2019) apply systematic layout planning for generating layout alternatives, and then use the fuzzy variant of AHP. The evaluation addresses human reliability from four perspectives, including software, hardware, environment, and liveware. Tayal et al. (2020) consider the stochastic dynamic variant of FLP and create a hybrid algorithm consisting of DEA, machine learning, and SA to predict the efficiency scores of different layout design alternatives.

It is worth mentioning that the layout design and evaluation processes are typically considered disjointly in the literature. In other words, it is not clear how to incorporate feedback from the DMs using the evaluation methods that currently exist in the FLP literature to design layouts that are more appealing. This can be of concern particularly if the DMs are not satisfied with the layout alternatives that are presented to them. Several researchers have created interactive systems that allow for DMs to be involved during the generation of layout alternatives to fill the gap between these processes. García-Hernández et al. (2015) develop an interactive GA where quantitative and qualitative criteria are considered simultaneously for generating layout design alternatives. After each generation, a set of high-fitness layout designs are presented to the DMs, and they provide feedback for the qualitative criteria.

García-Hernandez et al. (2020) present a similar model but instead use the coral reefs optimization metaheuristic for generating layout alternatives.

2.5.1 Socially distanced layouts using FLP

Because of the recent global pandemic, additional factors have been introduced that influence how new/existing layout configurations should be designed/modified to reduce the spread of infection in the facility, as well as for evaluating the quality of a layout design under pandemic conditions. Examples of such factors include social distancing, where it is required for individuals to be separated by a minimum distance, and the traffic flows of occupants within the facility. Consideration of these factors simultaneously is currently scarce in the FLP literature, and few research efforts have been performed that incorporate social distancing in the context of FLP (Bortolete et al., 2021; Fischetti et al., 2021; Ugali et al. 2021). One of the goals for this dissertation is to propose a framework that accommodates social distancing and occupant flow for improving the resilience of facilities that are operating under pandemic conditions.

Under the traditional FLP setting, social distancing requirements go together with how clearances are applied, where it is assumed that all clearance requirements are equal to the minimum separation threshold for satisfying the social distancing constraints. Instead of departments being arranged inside of a building envelope, perhaps a problem instance of FLP in consideration of social distancing may consist of a variety of objects that humans can interact with, such as desks, chairs, tables, etc. Based on how these objects are arranged in the facility, occupants will need to be able to navigate through the system in a fashion that (1) makes them feel comfortable, and (2) reduces the risk of infection. One way to accommodate this is to enforce unidirectional flow along designated walking paths, where occupants must walk along

a single direction. This approach was shown to be helpful for reducing the spread of infection among healthcare workers (Lenaghan and Schwedhelm, 2015; Zimring et al., 2018; Wong, 2019).

2.6 Automated layout generation in architecture

As mentioned in the previous chapter, FLP is a robust modelling framework that can easily be extended to other settings. Architecture is one example that blends together seamlessly with FLP since both problem settings address the arrangement of departments inside of facilities. The difference between FLP in the manufacturing and architectural settings is the consideration of other facility factors in the latter, such as lighting, window placement, doorways, beam placement, etc. (Michalek et al., 2002; Space Planning Basics, 2016). These factors can be addressed in traditional FLP problem formulations, but there are other complex factors in architecture that cannot easily be expressed, including building material type, building aesthetics, etc. Solution approaches that have been applied in this domain are similar to the most common ones used for FLP, including mathematical optimization (Glover et al., 1985; Michalek et al., 2002; Michalek and Papalambros, 2002; Kamol and Krung, 2005; Wu et al., 2018), metaheuristics (Verma and Thakur, 2010; Guo and Li, 2017; Laignel et al. 2021), and graph theory (Grason, 1971; Ruch, 1978), as well as other techniques such as machine learning (Yeh, 2006; Chaillou, 2019; Hu et al., 2020) and Bayesian theory (Merrell et al., 2010).

2.7 Concluding remarks for the literature review

This literature review considers research efforts from 2010 to 2021 for FLP. It is clear that metaheuristic algorithms are more prevalent in the literature for FLP since they are capable of solving the larger-size problem instances compared to when applying exact methods. For this reason, research is still necessary for reducing the

computational complexity of FLP either by reformulating existing models and creating valid inequality constraints. DRLP is the material handling configuration system of interest for addressing this concern in the dissertation.

Facility resilience is another research direction in FLP that is still developing, particularly in the context of pandemic events such as COVID-19. Stochastic variants of FLP have been introduced for variable demand over a planning horizon, and consideration of disastrous events are common for FLP in the chemical plant domain. However, there is still a gap in the literature between FLP and pandemic events. As far as the application of FLP is concerned, ideas and concepts from FLP are being applied in the architecture community for automating the layout generating process. However, there exists some open avenues of research in this domain with respect to developing more compact models using mathematical optimization and integrating the architect's needs during the layout design process. These three topics form the basis of this dissertation, and more thorough literature reviews for each of them will be presented in the subsequent chapters.

CHAPTER 3: COMPACT MILP MODELS FOR DRLP

3.1 Introduction

Consider a variant of FLP known as the *double row layout problem* (DRLP), which assigns a given number of departments along two rows that are separated by a central corridor. Similar problem variants include the *single row layout problem* (SRLP) (Simmons, 1969; Keller and Buscher, 2015) and *multi row layout problem* (MRLP) (Heragu and Kusiak, 1992; Tubaileh and Siam, 2017), where the former considers the placement of departments along a single row, and the latter considers the placement of departments in multiple rows such that rows are separated by corridors. This type of design is well known and widely used in modern flexible manufacturing systems (Tubaileh and Siam, 2017), where automated guided vehicles can be used to efficiently transport materials between departments along the corridor to increase the flexibility of the system. This chapter considers a variant of DRLP where minimum clearances between departments are enforced while the objective is to minimize the total material flow cost (this problem is denoted by DRLP-C hereafter).

DRLP-C was first formulated by Chung and Tanchoco (2010) as a MILP. Assuming that the width of the corridor is sufficiently large or the clearance between departments in different rows is irrelevant, the minimum clearance requirements are applied only when both departments are placed in the same row. As in typical DRLPs (Heragu and Kusiak, 1988), the distance between two departments in DRLP-C is measured by the length of the corridor between center points of respective departments without accounting for the width of the corridor. Without loss of generality, it is assumed that the corridor runs horizontally from left to right as illustrated in Figure 3.1.

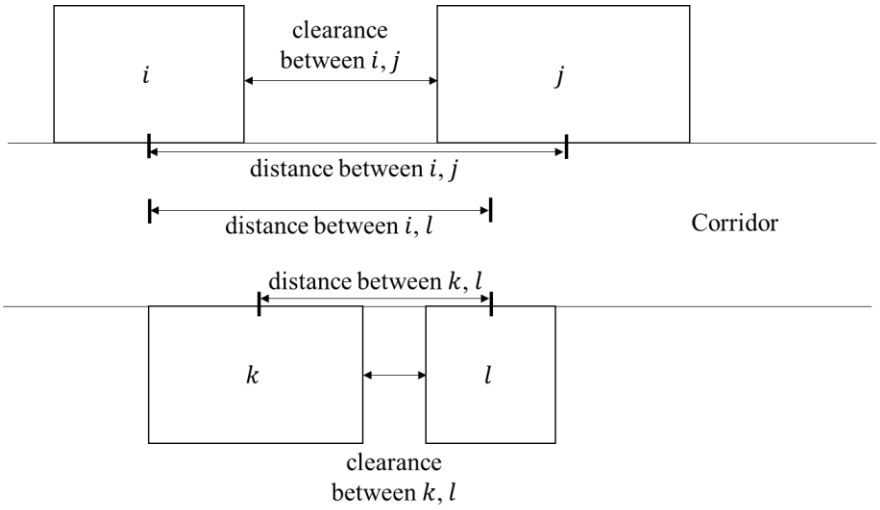


Figure 3.1: Illustration of DRLP-C and clearances between departments

The model of Chung and Tanchoco contains an error as the clearance is not observed for certain situations in the constraints that keep departments from overlapping. Zhang and Murray (2012) present a corrected model, where additional binary decision variables are introduced to explicitly account for the relative location of departments when they are assigned to the same row. Given the fact that DRLP-C is strongly NP-hard (Chung and Tanchco, 2010), this chapter proposes two new MILP formulations that consist of fewer binary decision variables than the model of Zhang and Murray in an effort to alleviate the computational burden to solve the problem. The key idea is to embed the clearance requirements into the constraints associated with distances, instead of directly incorporating them into the constraints enforcing the relative locations of departments. The remainder of this chapter is organized as follows. In the next section (Section 3.2), a literature review of relevant studies is provided. In Section 3.3, two proposed MILP formulations are presented and compared with the model of Zhang and Murray (2012). Also, three sets of symmetry-breaking constraints (SBCs) are introduced in Section 3.3. In Section 3.4, the results of the computational experiments that were performed on problem instances

generated in a similar fashion as in Chung and Tanchoco (2010) are reported. Section 3.5 concludes the chapter with a summary and a remark on the potential extension of this study.

3.2 Literature Review

Amaral (2013b) considers DRLP where clearance requirements are embedded in the department lengths (*implicit* clearances) and presents a MILP formulation consisting of fewer decision variables in comparison to the model of Chung and Tanchoco (2012), where clearance requirements are specified for all pairs of departments to indicate the minimum separation distance when they are assigned to the same row (*explicit clearances*). Amaral introduces valid inequalities for the convex hull of a set of points that represents a partition of departments into two segments using a set of binary variables representing the row membership and the relative location. In addition, a symmetry breaking constraint similar to the one by Sherali et al. (2003), known as the *position p-q method* is implemented. This constraint forces the center point of department p to be less than the center point of department q. It is suggested that departments p and q are selected as the pair of departments that have the least material flow.

Amaral (2019) presents an alternative formulation with fewer decision variables and more constraints in comparison to Amaral (2013b). However, the computational experiments do not display statistically significant differences in the performance between them. Secchin and Amaral (2019) introduce additional decision variables to the model of Amaral (2013b) to reflect relative positions of three departments in the same row and replace non-overlapping constraints with tighter constraints. They observe that the number of nodes explored by the branch-and-bound method is decreased after including additional valid inequalities in their proposed

formulation. Chae and Regan (2020) extend the model of Secchin and Amaral by introducing binary variables that indicate the row membership of each department and adding constraints that enforce the XNOR gate between these variables. Computational experiments of Chae and Regan show that their model consumes lesser time for a majority of the test problem instances when compared to the models of Amaral (2013b) and Secchin and Amaral (2019). Amaral (2020) formulated a MILP for DRLP based on a linear extension of a partial order. The formulation has the least number of binary variables in the literature for DRLP when implicit clearances are enforced.

Murray et al. (2012) formulate DRLP-C as a bi-objective MILP model to simultaneously minimize the total material flow cost and the area of the facility. They consider crossing the corridor as part of the transportation path of material flows, and hence, their model results in different layout designs from the solutions of Zhang and Murray (2012), even when the problem of Murray et al. is set to minimize only the material flow cost. Murray et al. also add constraints to the model of Zhang and Murray (2012) to reduce the problem symmetry by randomly selecting a department and fixing it to the first row, as well as restricting its respective center points to be less than or equal to a given threshold. Zuo et al. (2016) extend the formulation of Zhang and Murray (2012) to include so-called *additional clearances*, which are deemed to be necessary for departmental maintenance or storage of excess inventory/materials. The authors assume that adjacent departments can share their additional clearances by allowing these regions to overlap.

Murray et al. (2013) consider DRLP-C with asymmetric material flows, for which they propose constructive heuristics and local search methods to solve larger-sized problem instances. A LP is incorporated into these methods to further improve

the overall solution quality. Gülşen et al. (2019) extend DRLP-C of Murray et al. (2013) to consider the process flow (or sequence of departments that must be visited) of a set of products as part of the design criteria. The problem is formulated as a mixed-integer NLP and a two-phase heuristic procedure is proposed. Wang et al. (2015) applied a SA metaheuristic along with LP to solve a dynamic variant of DRLP-C for the situation where material flows between departments are subject to change over time. Under this setting, departments are assumed to be rearranged in each period, which makes it a difficult problem to solve using exact methods. Guan et al. (2020) applied a decomposition-based algorithm in conjunction with local search and PSO methods. The master problem is decomposed into two subproblems, where one ignores minimum clearance requirements between adjacent departments to optimize the sequences in both rows, and the other applies PSO to enforce the clearance requirements with a fixed sequence of departments.

Amaral (2012) presents a variant of DRLP known as the *corridor allocation problem* (CAP). CAP assumes that there is no empty space between adjacent departments. This restriction is important for certain applications, such as arranging rooms in a hospital or an office building to better utilize the available space. DRLP is different from CAP since empty spaces are allowed in-between departments. Kalita and Datta (2014) formulate a nonlinear bi-objective optimization model for CAP where the total material flow cost and the length of the corridor are minimized simultaneously. A permutation-based GA is implemented for solving an unconstrained representation of the model. Another variant of DRLP, called *parallel-row ordering problem*, is first introduced by Amaral (2013a) to consider the situation where a subset of departments share some common characteristic(s) and it is required for them to be assigned to the same row. Yang et al. (2019) modify some constraints

of the model in Amaral (2013a) and present their computational experiments where the modified formulation finds optimal solutions in lesser time in their study.

As mentioned earlier, DRLP-C considered in this chapter is originally formulated by Chung and Tanchoco (2010) and corrected by Zhang and Murray (2012). Those MILP formulations embed minimum clearances directly into the constraints that determine relative locations of center points of departments. As will be seen in the next section, the proposed formulations maintain separate constraints that relate the minimum clearance requirements to distances between departments.

3.3 Proposed models

3.3.1 Model formulations

In this subsection, two MILP models are presented for DRLP-C. In comparison to the model of Zhang and Murray (2012), the proposed two models have fewer decision variables (both binary and continuous) with the motivation that having fewer variables, especially binary variables, may be computationally advantageous in finding a solution using an off-the-shelf solver. The following notation of the problem parameters is used to present the model:

n	Number of departments
I	$\{1,2,\dots,n\}$: index set of departments
K	$\{1,2\}$: index set of rows
M	$\gg 1$: a large constant
l_i	Length of department $i \in I$
f_{ij}	Material flow cost per unit distance between departments i and j $(i < j)$
c_{ij}	Minimum clearance between departments i and j $(i < j)$.

As described in Section 3.1, the minimum clearance between departments is enforced only if they are placed in the same row. The distance between a pair of departments is measured by the length of the corridor between two center points of respective

departments, regardless of rows where those departments are placed. The following decision variables are used in the proposed models:

x_{ik}: center point of department i in row k if i is assigned to k; 0 if department i is not in row k

d_{ij}: distance between departments i and j $(i < j)$

y_{ik}: 1 if department i assigned to row k; 0 otherwise

z_{ij}: 1 if the center point of department i is to the left of the center point of department j regardless of their row membership; 0 otherwise $(i < j)$

δ_{kij}: auxiliary binary variable for disjunctive constraints $(i < j)$.

Using the notation defined above, the first MILP model (*M1*) can be stated as follows:

$$\text{Minimize} \quad \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} f_{ij} d_{ij} \tag{3.1}$$

subject to

$$x_{ik} \leq M y_{ik} \qquad \forall i \in I, k \in K \tag{3.2}$$

$$\sum_{k \in K} y_{ik} = 1 \qquad \forall i \in I \tag{3.3}$$

$$x_{ik} \geq x_{jk} + \frac{l_i y_{ik} + l_j y_{jk}}{2} - M\delta_{kij} \qquad \forall i, j \in I, i < j, k \in K \tag{3.4}$$

$$x_{jk} \geq x_{ik} + \frac{l_i y_{ik} + l_j y_{jk}}{2} - M(1 - \delta_{kij}) \qquad \forall i, j \in I, i < j, k \in K \tag{3.5}$$

$$d_{ij} \geq \sum_{k \in K} x_{jk} - \sum_{k \in K} x_{ik} \qquad \forall i, j \in I, i < j \tag{3.6}$$

$$d_{ij} \geq \sum_{k \in K} x_{ik} - \sum_{k \in K} x_{jk} \qquad \forall i, j \in I, i < j \tag{3.7}$$

$$d_{ij} \leq \sum_{k \in K} x_{jk} - \sum_{k \in K} x_{ik} + M(1 - z_{ij}) \qquad \forall i, j \in I, i < j \tag{3.8}$$

$$d_{ij} \leq \sum_{k \in K} x_{ik} - \sum_{k \in K} x_{jk} + M z_{ij} \qquad \forall i, j \in I, i < j \tag{3.9}$$

$$d_{ij} \geq \left(c_{ij} + \frac{l_i + l_j}{2} \right) (y_{ik} + y_{jk} - 1) \qquad \forall i, j \in I, i < j, k \in K \tag{3.10}$$

$$x_{ik} \geq 0 \qquad \forall i \in I, k \in K \tag{3.11}$$

$$d_{ij} \geq 0 \qquad \forall i, j \in I, i < j \tag{3.12}$$

$$y_{ik} \in \{0,1\} \qquad \forall i \in I, k \in K \qquad (3.13)$$

$$z_{ij} \in \{0,1\} \qquad \forall i, j \in I\ , i < j \qquad (3.14)$$

$$\delta_{kij} \in \{0,1\} \qquad \forall i, j \in I, i < j, k \in K. \qquad (3.15)$$

The objective function (3.1) minimizes the total material flow costs of the layout design. Constraint (3.2) ensures $x_{ik} = 0$ if department i is not assigned to row k (i.e., $y_{ik} = 0$). Constraint (3.3) assigns each department to one of the upper ($k = 1$) and lower rows ($k = 2$). Constraints (3.4)–(3.5) keep departments from overlapping when they are placed in the same row. In specific, there are four possible cases of the row membership: ($y_{ik} = y_{jk} = 1$), ($y_{ik} = 1$, $y_{jk} = 0$), ($y_{ik} = 0$, $y_{jk} = 1$), and ($y_{ik} = y_{jk} = 0$). If $y_{ik} = y_{jk} = 1$, δ_{kij} can either be 1 or 0 in (3.4) and (3.5). $\delta_{kij} = 1$ implies that j is to the right of i, and $\delta_{kij} = 0$ results in i to the right of j. Next, if $y_{ik} = 1$ and $y_{jk} = 0$, then only $\delta_{kij} = 0$ satisfies (3.4) and (3.5) since $x_{jk} = 0$ from (3.2). In result, $x_{ik} \geq l_i/2$ from (3.4). Similarly, if $y_{ik} = 0$ and $y_{jk} = 1$, then $\delta_{kij} = 1$, and in turn, $x_{jk} \geq l_j/2$ from (3.5). Finally, if $y_{ik} = y_{jk} = 0$, then (3.4) and (3.5) are satisfied regardless of the value of δ_{kij}. It is noteworthy that the role of δ_{kij} is similar to that of z_{kij} in the formulation of Chung and Tanchoco (2010). As pointed out by Zhang and Murray (2012), (4.10)–(4.11) of Chung and Tanchoco (2010) are incorrectly satisfied by solutions that violate the minimum clearance. However, since the clearance is indirectly enforced by (3.10) in the proposed formulation, employing δ_{kij} in (3.4)–(3.5) for the sole purpose of having non-overlapping departments is correct and valid.

Since $\sum_{k \in K} x_{ik}$ represents the center point of department i, constraints (3.6)–(3.9) assign the exact distance between departments i and j. In addition to (3.6)–(3.7) that describe the lower bounds on d_{ij}, (3.8)–(3.9) impose its upper bound. If $\sum_{k \in K} x_{ik} < \sum_{k \in K} x_{jk}$, (3.8) and (3.9) warrant z_{ij} = 1 since $d_{ij} \geq 0$. Conversely, if

$\sum_{k\in K} x_{ik} > \sum_{k\in K} x_{jk}$, then (3.8) and (3.9) give z_{ij} = 0. When $\sum_{k\in K} x_{ik} = \sum_{k\in K} x_{jk}$, d_{ij} = 0 and either z_{ij} = 0 or 1 trivially satisfies (3.8) and (3.9). Constraint (3.10) forces the center points of i and j to be separated by (at least) their minimum clearance plus respective half-lengths when both departments are assigned to the same row (i.e., $y_{ik} = y_{jk} = 1$). If y_{ik} or y_{jk} equals 0, then (3.10) becomes redundant. In $M1$, there are $3n(n-1)/2 + 2n$ binary variables, $n(n-1)/2 + 2n$ continuous variables, $5n(n-1) + 2n$ inequality constraints, and n equality constraints.

It is still possible to further reduce the number of variables for the same problem setting, thus leading us to the second MILP (*M2*). Suppose that constraints (3.4)–(3.5) are replaced by the following constraints.

$$x_{ik} \geq x_{jk} + \frac{l_i y_{ik} + l_j y_{jk}}{2} - M(2 + z_{ij} - y_{ik} - y_{jk}) \qquad \forall i, j \in I, i < j, k \in K \quad (3.16)$$

$$x_{jk} \geq x_{ik} + \frac{l_i y_{ik} + l_j y_{jk}}{2} - M(2 + (1 - z_{ij}) - y_{ik} - y_{jk}) \qquad \forall i, j \in I, i < j, k \in K \quad (3.17)$$

$$\sum_{k\in K} x_{ik} \geq \frac{l_i}{2} \qquad \forall i \in I. \quad (3.18)$$

Recall that, when i and j are in different rows (i.e., $y_{ik} + y_{jk} = 1$), (3.4) and (3.5) hold true, as long as the center points of i and j are greater than or equal to their respective half-lengths, in the respective rows with a suitably selected value of δ_{kij} (as explained above, $\delta_{kij} = 1$ if y_{ik} = 0 and y_{jk} = 1; $\delta_{kij} = 0$ if y_{ik} = 1 and y_{jk} = 0). Constraints (3.16) and (3.17) hold true for any positions of i and j when $y_{ik} + y_{jk} = 1$ since M is subtracted in the right-hand-side of both inequalities. To ensure the minimum possible value of the position of a department is its half-length, the lower bounding constraint (3.18) is explicitly added. Now, consider a case where both departments are in row k (i.e., y_{ik} = y_{jk} = 1). If i is to the left of j (i.e., z_{ij} = 1), then δ_{kij} = 1 makes (3.5) intact to keep departments from overlapping. Under the same

situation, z_{ij} = 1 along with $y_{ik} + y_{jk} = 2$ makes (3.17) intact. Using the same argument, (3.4) and (3.16) become intact when i is to the right of j. Finally, consider a case where both departments are not in row k (i.e., $y_{ik} = y_{jk}$ = 0). In this case, (3.4)–(3.5) are redundant regardless of the value of δ_{kij} since $x_{ik} = x_{jk} = 0$. Similarly, (3.16)–(3.17) become redundant regardless of the values of z_{ij}.

Replacing (3.4)–(3.5) by (3.16)–(3.18), the number of binary variables in $M2$ is reduced to $n(n-1)/2 + 2n$. The number of continuous variables is the same as that of *M1*, but the number of inequality constraints is increased by n because of (3.18). As noted by Chung and Tanchoco (2010), the value of a large number M for (3.2), (3.4)–(3.5), (3.8)–(3.9), and (3.16)–(3.17) can be set as

$$M = \sum_{i \in I} \left\{ l_i + \max_{j \in I}(c_{ij}) \right\}.$$

Before comparing the proposed MILP formulations with that of Zhang and Murray (2012), an intuitive property of an optimal solution of DRLP-C is presented to show the difference between formulations with respect to the lower bound on the location variable.

Proposition 3.1: Consider a DRLP-C problem with $n > 1$ and $f_{ij} > 0$ for $i, j \in I$, where $i < j$. Then, any optimal layout has at least one department in each row.

Proof: Suppose that there exists an optimal layout that has all departments in one row. Without loss of generality, let this be row 1. Furthermore, let (i) be the index of the department in the i-th position from the left in the optimal layout. Accordingly, $x_{(i)k}$ and $d_{(i)(j)}$ denote the location of the center point of department (i) in row k and the distance between departments (i) and (j) in this optimal layout, respectively. Note that the objective function value is $\sum_{i=1}^{n-1} \sum_{j=i+1}^{n} f_{(i)(j)} d_{(i)(j)}$. Let us construct a new layout as follows. First, move department (1) to row 2 while keeping the same position.

Then, shift the positions of the remaining departments in row 1 by $l_{(1)}/2$ to the left without altering the order (see Figure 3.2). Let $\bar{x}_{(i)k}$ and $\bar{d}_{(i)(j)}$ denote the position of the center point of department (i) in row k and the distance between departments (i) and (j) in the new layout, respectively. Observe that $\bar{x}_{(j)1} < \bar{x}_{(j+1)1}$ for $j = 2, 3, \ldots, n$ since their order remains the same. Furthermore, since $x_{(2)1} > x_{(1)1} + l_{(1)}/2$ and $\bar{x}_{(1)2} = x_{(1)(1)}$, we have $\bar{x}_{(2)1} = x_{(2)1} - l_{(1)}/2 > \bar{x}_{(1)2}$. Therefore,

$$\bar{d}_{(i)(j)} = \bar{x}_{(j)1} - \bar{x}_{(i)1} = \left(x_{(j)1} - l_{(1)}/2\right) - \left(x_{(i)1} - l_{(1)}/2\right) = d_{(i)(j)} \qquad \forall i, j \in I \setminus \{1\},\ i < j \qquad (3.19)$$

and

$$\bar{d}_{(1)(j)} = \bar{x}_{(j)1} - \bar{x}_{(1)2} = \left(x_{(j)1} - l_{(1)}/2\right) - \bar{x}_{(1)2} \qquad \forall j \in I \setminus \{1\}. \qquad (3.20)$$

$$= d_{(1)(j)} - l_{(1)}/2 < d_{(1)(j)}$$

The objective function value of the new layout is $\sum_{i=1}^{n-1}\sum_{j=i+1}^{n} f_{(i)(j)}\bar{d}_{(i)(j)} = \sum_{j=2}^{n} f_{(1)(j)}\bar{d}_{(1)(j)} + \sum_{i=2}^{n-1}\sum_{j=i+1}^{n} f_{(i)(j)}\bar{d}_{(i)(j)} < \sum_{j=2}^{n} f_{(1)(j)}d_{(1)(j)} + \sum_{i=2}^{n-1}\sum_{j=i+1}^{n} f_{(i)(j)}d_{(i)(j)}$, where the inequality holds from (3.19) and (3.20). This contradicts the fact that the first layout is optimal. This completes the proof. ■

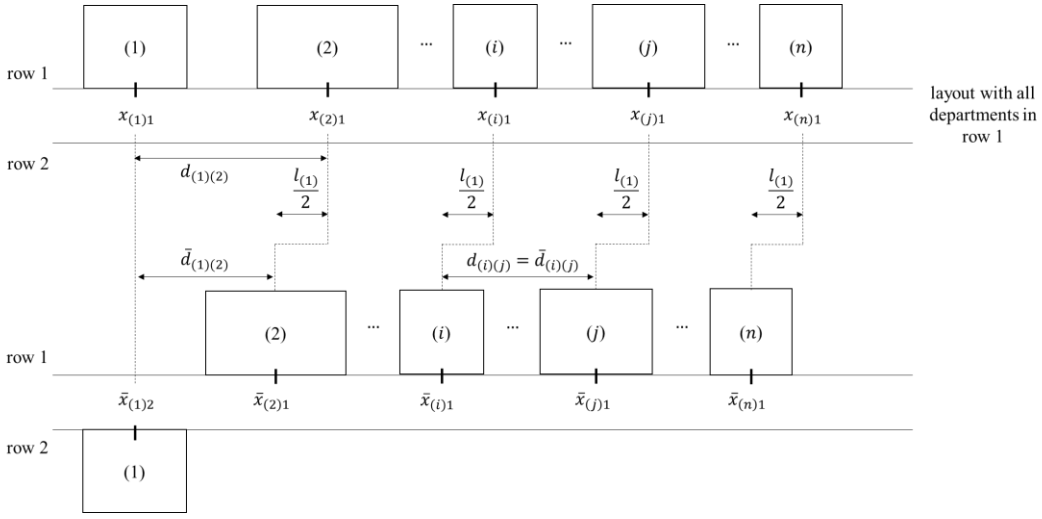


Figure 3.2: Graphical representation of Proposition 3.1

Thus, $\sum_{i \in I} y_{ik} \geq 1$ for $k \in \{1,2\}$ is deemed valid when $n > 1$ and $f_{ij} > 0$ for $i, j \in I$.

For the sake of completeness, *M0* is presented below, which appeared on pages 4221-4222 in Zhang and Murray (2012). In *M0*, the pairwise distance decision variables are separated into positive and negative components d_{ij}^{+} and d_{ij}^{-} to consider the cases where $\sum_{k\in K} x_{ik} < \sum_{k\in K} x_{jk}$ and $\sum_{k\in K} x_{ik} > \sum_{k\in K} x_{jk}$, respectively, and are measured using equality constraints.

$$\text{Minimize} \quad \sum_{i=1}^{n-1}\sum_{j=i+1}^{n} f_{ij}(d_{ij}^{+} + d_{ij}^{-}) \tag{3.21}$$

subject to

$$x_{ik} \le M y_{ik} \qquad \forall i \in I, k \in K \tag{3.22}$$

$$\sum_{k\in K} y_{ik} = 1 \qquad \forall i \in I \tag{3.23}$$

$$x_{ik} \ge x_{jk} + \frac{l_i y_{ik} + l_j y_{jk}}{2} + c_{ij} z_{kji} - M\left(1 - z_{kji}\right) \qquad \forall i, j \in I, i < j, k \in K \tag{3.24}$$

$$x_{jk} \ge x_{ik} + \frac{l_i y_{ik} + l_j y_{jk}}{2} + c_{ij} z_{kij} - M\left(1 - z_{kij}\right) \qquad \forall i, j \in I, i < j, k \in K \tag{3.25}$$

$$\sum_{k\in K} x_{jk} - \sum_{k\in K} x_{ik} = d_{ij}^{+} - d_{ij}^{-} \qquad \forall i, j \in I, i < j \tag{3.26}$$

$$z_{kij} + z_{kji} \le \frac{1}{2}\left(y_{ik} + y_{jk}\right) \qquad \forall i, j \in I, i < j, k \in K \tag{3.27}$$

$$z_{kij} + z_{kji} + 1 \ge y_{ik} + y_{jk} \qquad \forall i, j \in I, i < j, k \in K \tag{3.28}$$

$$x_{ik} \ge 0 \qquad \forall i \in I, k \in K \tag{3.29}$$

$$d_{ij}^{+}, d_{ij}^{-} \ge 0 \qquad \forall i, j \in I, i < j \tag{3.30}$$

$$y_{ik} \in \{0,1\} \qquad \forall i \in I, k \in K \tag{3.31}$$

$$z_{kij} \in \{0,1\} \qquad \forall i, j \in I\,, i < j, k \in K. \tag{3.32}$$

There are three major differences between the proposed models and *M0*. The most prominent difference is regarding how the minimum clearance requirements are expressed in the constraints. In specific, Zhang and Murray (2012) incorporate the minimum clearance into the non-overlapping constraints (3.24)–(3.25), which is same as the approach taken by Chung and Tanchoco (2010). Observe that incorporating the

minimum clearance is affected by the relative location between departments in order to maintain the feasibility in (3.19)–(3.20). In *M1* and *M2*, on the contrary, the minimum clearance is enforced directly to the distance between two departments as in (3.10), and as a result, fewer binary variables, δ_{kij} for $i < j$, are used to express disjunctive constraints (3.4)–(3.5).

Next, instead of using the upper bound d_{ij}^{+} and d_{ij}^{-} on the distance between the center points of departments as in (3.26) of *M0* (equivalent to $\sum_{k\in K} x_{jk} - \sum_{k\in K} x_{ik} \leq d_{ij}^{+}$ and $\sum_{k\in K} x_{ik} - \sum_{k\in K} x_{jk} \leq d_{ij}^{-}$), the proposed formulations *M1* and *M2* assign the exact distance to d_{ij} by (3.6)–(3.9) using lower and upper bounding constraints. This is necessary because the minimum clearance is incorporated by a lower bounding constraint (3.10) on the distance instead of non-overlapping constraints (3.4)–(3.5). As a result, d_{ij} of *M1* and *M2* is the exact distance between departments i and j without the need for the objective function to be minimized. On the other hand, $d_{ij}^{+} + d_{ij}^{-}$ of *M0* is an upper bound on the distance, which becomes tight when the objective function is minimized.

Third, the lower bounding constraints on the center points of departments as in (3.18) are missing in the original formulation of Zhang and Murray. Therefore, it is possible that the left-most department has its center point at zero. For example, consider a situation where department i^* is in row 1 while all other departments are in row 2. Then, the non-overlapping constraints (3.24)–(3.25) between i^* and $j^* \neq i^*$ become redundant since $z_{ki^*j} = z_{kji^*} = 0$. This results in x_{i^*} being restricted only by the nonnegativity constraints, and hence, x_{i^*} is allowed to be zero. To this end, the lower bounding constraints (3.18) are also added to *M*0 in the computational study presented in the next section. As a result, *M0* has n additional inequality constraints

from the original formulation of Zhang and Murray. It should be noted that *M1* does not need such lower bounding constraints since (3.4)–(3.5) are disjunctive constraints, which enable the half-length lower bound to the department in row k when $y_{ik} + y_{jk} = 1$. One may argue that the lower bounding constraint is necessary for the case where all departments are in one row. That is, if all departments are in one row, (3.4)–(3.5) will not enforce the half-length lower bound to the left-most department since $y_{ik} + y_{jk} \neq 1$ for all i, j. However, from Proposition 3.1, there exists at least one department in each row, which guarantees that the half-length lower bound is enforced to all departments. Note that the model of Chung and Tanchoco (2010) did not need the half-length lower bounding constraint for the same reason, although their model is incorrect.

Table 3.1: Model size comparison

	Number of variables		Number of constraints	
Model	Binary	Continuous	Inequality	Equality
M0	$2n(n-1)+2n$	$n(n-1)+2n$	$4n(n-1)+3n$	$n(n-1)/2+n$
M1	$3n(n-1)/2+2n$	$n(n-1)/2+2n$	$5n(n-1)+2n$	n
M2	$n(n-1)/2+2n$	$n(n-1)/2+2n$	$5n(n-1)+3n$	n

The numbers of decision variables and constraints in *M0*, *M1* and *M2* are compared in Table 3.1. The numbers of both binary and continuous variables are in the order of n^2 in all three models. However, when comparing coefficients of n^2, *M1* has 25% and 50% smaller values than *M0* for respective variable types. Furthermore, the coefficient becomes 1/3 in *M2* when compared to that in *M1* for the number of binary variables. The number of inequality constraints in all three models is in the order of n^2. The number of equality constraints in *M1* and *M2* is in the order of n whereas *M0* has the order of n^2.

3.3.2 Symmetry-breaking constraints

As commonly observed in combinatorial optimization problems, DRLP-C suffers from a problem symmetry since the permutations of departments between rows, as well as between the directions of the corridor, make it possible to produce alternative solutions that have the same objective function value (Margot, 2010). In an effort to alleviate the burden to unnecessarily explore alternative solutions, two sets of symmetry-breaking constraints (SBCs) that can be applied to all three models are introduced next.

The first SBC is to eliminate the symmetry with respect to the direction of the corridor. For this SBC, a pair of departments p and q are selected, and their relative locations are fixed by letting the center point of p be less than or equal to the center point of q as follows.

$$\sum_{k \in K} x_{pk} \leq \sum_{k \in K} x_{qk}. \tag{3.33}$$

When this SBC is added, the MILP solver avoids exploring the symmetric solutions where the direction of the corridor is flipped, and hence, the relative locations of p and q are reversed. While any pair of departments can be selected for (3.33), two departments having the lowest flow cost between them are chosen in the computational implementation in Section 3.4.

Another set of SBCs is to break the symmetry between rows by pre-assigning some department $i^* \in I$ to one of the two rows. This department can be selected randomly or strategically. For example, Murray et al. (2012) select the department that has the longest length. Assuming that i^* is assigned to the upper row (i.e., row 1), the following constraints are added.

$$y_{i^*1} = 1 \qquad i^* \in I \tag{3.34}$$

$$x_{i^*2} = 0 \qquad i^* \in I. \tag{3.35}$$

Constraint (3.34) assigns department i^* to the upper row and constraint (3.35) forces the center point of department i^* in the lower row to equal 0. Note that (3.18) is still satisfied when enforcing (3.34)-(3.35) for department i^* since $x_{i^*1} \geq \frac{l_i}{2}$ when $y_{i^*1} = 1$.

Note that the above SBCs can be applied to all three models. In addition, another SBCs that can be applied to *M1* is introduced next. Recall that δ_{kij} is an auxiliary binary variable for the disjunctive non-overlapping constraints (3.4)–(3.5). Suppose that department i is in row $k \in K$ and j is in row $\bar{k} \in K \setminus \{k\}$ (i.e., in the opposite row). Since $x_{i\bar{k}} = 0$ and $x_{jk} = 0$ while $x_{ik} > 0$ and $x_{j\bar{k}} > 0$, δ_{kij} must be zero and $\delta_{\bar{k}ij}$ must be one from (3.4) and (3.5). On the other hand, suppose that departments i and j are placed in the same row k. Then, regardless of the value of δ_{kij}, $\delta_{\bar{k}ij}$ can take on either 0 or 1 since $x_{i\bar{k}} = x_{j\bar{k}} = 0$, while the objective function value is not affected. To eliminate the symmetry due to this situation, the following SBC is valid for *M1*.

$$\sum_{k \in K} \delta_{kij} = 1 \qquad \forall i, j \in I, \ \ i < j. \tag{3.36}$$

Incorporating (3.36) to *M1* results in $n(n-1)/2$ additional equality constraints. In the next section, the computational efficacies of three formulations in conjunction with aforementioned SBCs are compared via a computational study.

3.4 Computational experiments

Computational experiments conducted to compare the performances of three MILP formulations for solving DRLP-C are reported in this section. A total of 60 problem instances were generated, in a similar fashion as in Chung and Tanchoco

(2010), using the testing factors specified in Table 3.2. The problem instances can be downloaded from Alaimo and Lim (2020).

The number of departments varies from 6 to 16 with increments of 2. For each number of departments, 10 instances were created by randomly generating problem parameters as displayed in Table 3.2. The MILP models were coded in Python (3.7.3) and solved using a commercial optimization solver, Gurobi (8.1.1). All runs were executed on a server equipped with four processors of Intel Xeon CPU E7-8890 v3 (3.60 GHz) and the operating system of Windows Server 2012 R2. A 1-hour time limit was imposed to account for the larger-sized problem instances which might not be solved to optimality in a reasonable amount of time. Three formulations (*M0, M1,* and *M2*) were solved by the Gurobi MILP solver while all solver parameters were set to their default values, and these results are reported in Section 3.4.1. The SBCs described in Section 3.2 were implemented in separate runs to investigate their effectiveness, and the results are reported in Section 3,4.2. When the default setting is used for the Gurobi solver, its presolver is enabled. This presolver is intended to make an MILP model smaller and easier to solve (Gurobi Optimization, Inc. 2018). Note that the efficacy of the presolver of Gurobi can be dependent on how the MILP model is formulated. In order to ascertain such dependency, additional experiments were conducted with the presolver disabled for *M0, M1,* and *M2* along with SBCs that displayed the most promising performance when the presolver was enabled. These results are reported in Section 3.4.3.

Table 3.2: Testing factors for generating random test problems

Test factors	Levels
Number of departments	6–16 (increments of 2)
Number of replications	10
Material flow matrix	Uniform (0,50) for each pair and rounded to nearest integer
Clearance matrix	Uniform (1,2) for each pair and rounded to one decimal point

Department length	Uniform (1,20) and rounded to nearest integer

3.4.1 Comparison of M0, M1, and M2

Among the 60 problem instances, 41 of them were solved to optimality by at least one of *M0, M1,* and *M2* within the 1-hour time limit. All 40 problem instances with 6–12 departments were solved to optimality by *M1* and *M2,* whereas *M0* reached optimality for 39 problem instances (30 with 6–10 departments and 9 with 12 departments). Also, one problem instance with 14 departments (instance 14_6) was solved to optimality by *M2*. None of the problem instances with 16 departments were solved to optimality. Out of those 41 problem instances, *M0, M1,* and *M2* consumed the least amount of solution time in 3, 2, and 36 of them, respectively. The instances for which *M0* outperformed *M1* and *M2* are relatively smaller-sized problems with n = 8 and 10. Average solution times (in seconds) consumed by *M0, M1,* and *M2* are displayed in Table 3.3 for each group with n = 6, 8, 10, and 12. In the table, the least average solution time is highlighted in boldface.

Table 3.3: Comparison of average solution times of *M0, M1,* and *M2*

	Average solution time (sec.)			Percentage improvement from *M0*	
n	*M0*	*M1*	*M2*	*M1*	*M2*
6	1.0	0.8	**0.6**	-20.00%	**-40.00%**
8	2.4	2.6	**2.2**	8.11%	**-8.52%**
10	32.1	18.1	**12.5**	-43.61%	**-61.06%**
12†	1466.5	373.7	**227.6**	-74.52%	**-84.48%**

† For n = 12, the average is taken from nine problem instances that were solved to optimality by all three MILPs.

M2 exhibits the least average solution times for n = 6, 8, 10 and 12. The percentage improvement in average solution time over *M0* ranges from 8.52% (10 departments) and 84.48% (12 departments). As displayed in Table 3.3, *M1* and *M2* clearly outperformed *M0* when n = 12 since the average solution time is improved by at least

74.52%. To compare the performance of the MILPs for the problem instances with n = 14 and 16, best objective function values (BOFVs) and optimality gaps (OGs) that were found after the 1-hour time limit are reported in Table 3.4. The minimum BOFVs and OG for each problem instance in the table is highlighted in boldface.

Out of the 20 instances, *M0, M1,* and *M2* produced the least BOFVs for 2, 6, and 16 cases, respectively. The three MILPs produced the same BOFV for 1 instance (instance 14_1). In addition, *M1* and *M2* resulted in the same BOFVs for 2 instances (14_6 and 14_10). As far as OGs are concerned, *M0, M1,* and *M2* display the least OGs for 6, 5, and 9 test instances, respectively. For n = 14, the average OGs are 31.85% (*M0*), 28.31% (*M1*), and 23.28% (*M2*). For n = 16, the average OGs are 63.09% (*M0*), 64.81% (*M1*), and 65.48% (*M2*). Note that *M0* resulted in the least OGs for a majority of the instances when n = 16 (6 in total), whereas *M2* resulted in the least OGs for 8 cases when n = 14. Overall, *M2* displayed the most promising performance among *M0, M1,* and *M2*.

Table 3.4: Comparison of OFVs and OGs for n = 14 and 16

		OFV			OG		
n	Inst.	*M0*	*M1*	*M2*	*M0*	*M1*	*M2*
14	14_1	**36,412.9**	**36,412.9**	**36,412.9**	23.84%	20.19%	**17.71%**
	14_2	37,759	37,759	**37,714.1**	32.77%	35.13%	**25.63%**
	14_3	42,044.5	41,913.5	**41,613.7**	38.42%	**18.95%**	19.20%
	14_4	45,915.6	46,215.7	**45,895**	25.04%	27.40%	**22.75%**
	14_5	41,413.1	41,445.7	**41,024**	38.20%	33.38%	**30.37%**
	14_6	46,888.4	**46,830.4**	**46,830.4**	21.60%	21.09%	**0.00%**
	14_7	40,042.5	39,734	**39,671.2**	49.44%	**37.82%**	53.80%
	14_8	**43,153.3**	43,637.9	43,496.4	33.80%	38.46%	**31.66%**
	14_9	37,481.7	37,537.7	**37,321.7**	30.24%	28.44%	**11.90%**
	14_10	46,549.5	**46,121.1**	**46,121.1**	25.10%	22.27%	**19.81%**
16	16_1	100,685.3	103,086.9	**100,147.4**	**71.34%**	80.19%	78.63%
	16_2	63,427	**62,046.1**	62,583.3	**55.81%**	57.33%	58.39%
	16_3	74,054	74,961.6	**73,741.8**	60.48%	60.03%	**53.81%**
	16_4	93,035.8	92,149.3	**91,174.9**	**70.00%**	78.15%	75.18%
	16_5	92,519.4	91,605.1	**90,762.8**	64.46%	**52.76%**	67.71%
	16_6	64,665.8	**63,883.6**	64,878.6	61.11%	**58.92%**	61.72%
	16_7	65,488.8	66,392	**64,819.9**	**56.18%**	57.60%	59.30%
	16_8	81,914.3	82,687.1	**80,989.7**	**67.05%**	76.58%	73.29%
	16_9	86,065.1	**85,358.3**	86,345.6	**66.63%**	75.73%	71.29%

	16_10	57,409.5	57,025.9	**56,757.4**	57.87%	**50.78%**	55.44%

3.4.2 Efficacy of symmetry-breaking constraints

To analyze the impact of the SBCs presented in Section 3.2, the group of problem instances with n = 12 was tested and the average solution times were compared. First, two sets of SBCs, (3.33) and (3.34)–(3.35), that can be applied to all three MILPs, were added individually, as well as combined together. Table 3.5 displays solution times consumed to achieve optimality, where columns with headings, Base, SBC1, SBC2, and SBC3, denote MILPs with no SBCs added, only (3.33) added, only (3.34)–(3.35) added, and both (3.33) and (3.34)–(3.35) added, respectively. The penultimate row in Table 3.5 represents the average solution time of 10 instances and the last row represents the average solution time of the first 9 instances. The latter is to account for only those that were solved to optimality by all MILPs. For SBC2, following the same strategy by Murray et al. (2012), the longest department was selected as mentioned in Section 3.2.

Table 3.5: Individual impact of symmetry-breaking constraints

	M0				*M1*				*M2*			
Inst.	Base	SBC1	SBC2	SBC3	Base	SBC1	SBC2	SBC3	Base	SBC1	SBC2	SBC3
12_1	1070.7	801.2	**416.7**	439.1	192.9	**137.7**	246.5	138.9	148.6	**105.7**	191.7	128.9
12_2	583.9	422.1	255.2	**35.5**	84.3	61.8	112.2	**57.8**	51.4	45.1	64.2	**42.9**
12_3	2189.9	1473.7	1860.4	**958.3**	605.3	350	414.1	**282.0**	334.6	204.8	294.6	**197.2**
12_4	1347.4	2227	**1858.9**	1888.1	805.5	**293.9**	558.7	386.0	536	**259.2**	525.2	340.9
12_5	715.1	297.3	**181.4**	240.3	81.4	58.5	85.7	**56.7**	53.5	38.1	101.8	**37.3**
12_6	3154.2	3194.2	**777.5**	1131.3	978.8	361.0	673.9	**260.0**	468.4	239.6	329.6	**228**
12_7	2386.7	1511.3	1410.1	**458.1**	360.8	**158.5**	404.7	325.0	249.7	**152.3**	295.3	202.8
12_8	996.7	352.2	457.6	**166.2**	159.6	103.5	117.3	**83.0**	132.2	60.7	129.9	**55.4**
12_9	753.8	**108**	330.6	207.2	94.5	93.1	116.4	**56.8**	73.8	**43.4**	80.4	54.5
12_10	*	*	**1483.6**	*	2223	**1477.2**	1494.6	2942.9	2475.8	1964	**1249.3**	1308.4
Avg	1679.9	1398.7	**903.2**	912.4	558.6	**309.5**	422.4	458.9	452.4	311.3	326.2	**259.6**
Avg†	1466.5	1154.1	838.7	**613.8**	373.7	**179.8**	303.3	182.9	227.6	**127.7**	223.6	143.1

† For n = 12, the average is taken from nine problem instances that were solved to optimality by all three MILPs.

* Gurobi solver terminated after 1-hour

When individual SBCs are applied to *M0*, SBC2 outperformed SBC1 in seven cases. This suggests that SBC2 is more effective in reducing the solution time for *M0*. When compared to the combination of these constraints (i.e., SBC3), SBC2 still consumed the least solution times in the majority of the instances (5 in total), while SBC3 outperformed SBC1 and SBC2 in four cases. It is noteworthy that, when *M0* is implemented, only SBC2 was able to optimally solve instance 12_10. When considering nine instances that were solved to optimality by *M0*, SBC3 resulted in the least average solution time (613.8 seconds), which was followed by SBC2 (838.7 seconds). When compared to the average solution time of the base *M0*, these average solutions times are improvements of 58.15% and 42.81%, respectively.

M1 in conjunction with SBCs solved all instances to optimality as its base model did. SBC1 and SBC3 respectively consumed the least solution times among the implemented formulations in 4 and 6 instances. However, the impact of SBC2 was not significant since it consumed longer time than the base model in five cases, not to mention that SBC1 and SBC3 outperformed SBC2 in all cases. When the average solution times of 10 instances are compared, SBC1 displays the best average solution time of 309.5 seconds, and SBC2 shows the second-best average solution time of 422.4 seconds, which are improvements of 44.59% and 24.38% over the base model, respectively.

Based on the observation above, SBC3 was selected as a viable option and applied to all three formulations to solve the entire 60 test instances. A '+' is appended to the name of each model to represent the addition of SBC3 to its base formulation. As their base models did, *M1+* and *M2+* found optimal solutions for all 40 problem instances

with 6–12 departments, while *M0+* could do so only for 39 problem instances (all but instance 12_10). For n = 14, all models along with SBC3 found an optimal solution for 1 instance (instance 14_6). However, *M0+* consumed the entire hour to achieve optimality right before the time limit was reached, whereas *M1+* and *M2+* found an optimal solution in 1,556.6 and 1,917.6 seconds, respectively. Recall, from Table 3.4, that the base model of *M2* was the only one that reached optimality before the time limit and the solution time was 3,121.3 seconds. None of the problem instances with 16 departments were solved to optimality. Out of the 41 problem instances, *M0+*, *M1+*, and *M2+* consumed the least amount of times in 7, 9, and 25 cases, respectively.

M0+ consumed an average solution time of 9.6 seconds for n = 10, which is slightly less than those of *M1+* and *M2+* by 1.9 and 2 seconds, respectively (see Table 3.6). However, *M2+* appears the most promising since it resulted in the least solution times for a majority of the test instances, as well as consumed the least solution time on average when n = 6, 8, and 12. *M1+* exhibits a similar performance as *M2+* when n = 6 and 8, but it is obvious that *M2+* outperformed *M1+* for n = 12.

Table 3.6: Comparison of average solution times with SBC3

	Average solution time (sec.)			Percentage improvement from base formulation			Percentage improvement from *M0+*	
n	*M0+*	*M1+*	*M2+*	*M0+*	*M1+*	*M2+*	*M1+*	*M2+*
6	0.9	0.9	**0.6**	-10.00%	12.50%	0.00%	0.00%	-33.33%
8	2.2	**1.4**	**1.4**	-8.33%	-46.15%	-36.36%	-36.36%	-36.36%
10	**9.6**	11.5	11.7	-70.09%	-36.46%	-6.40%	19.79%	21.88%
12†	613.8	182.9	**143.1**	-58.15%	-51.06%	-37.13%	-70.20%	-76.69%

† For n = 12, the average is taken from nine problem instances that were solved to optimality by all three MILPs.

From columns 5-7 in Table 3.6, observe that the average solution times were significantly improved when SBC3 was added in most cases except for n = 6, for which the average

solution times of all models were less than one second. The most drastic improvement was achieved for *M0+* when n = 10 (70.09%). For the larger-sized instances with n = 14 and 16, the BOFVs and OGs recorded after the 1-hour time limit are compared in Table 3.7. Out of the 20 larger-sized instances, *M0+*, *M1+*, and *M2+* produced the least BOFVs in 7, 6, and 11 cases, respectively, while there were two 2-way ties and one 3-way tie. It is worth mentioning that improved BOFVs were found in 11, 14, and 9 instances when SBC3 was added to *M0, M1,* and *M2*, respectively. The average OGs when n = 14 were 24.17% (*M0+*), 19.59% (*M1+*), and 21.05% (*M2+*). For n = 16, the average OGs are 54.85% (*M0+*), 60.96% (*M1+*), and 62.15% (*M2+*). Note that the average OGs were reduced by 20.1%, 19.9% and 8.1%, on average, as a result of applying SBC3 to *M0*, *M1*, and *M2*, respectively.

Recall that the SBC in (3.36) is applicable only for *M1* and *M1+*. In the rest of this subsection, the results from the implementation of (3.36), denoted by SBC4, in conjunction with *M1* and *M1+* to solve all 60 problem instances are reported. Let *M1'* and *M1'+* denote *M1* and *M1+* formulations with SBC4, respectively. Among the 60 problem instances, *M1'* and *M1'+* were able to optimally solve 41 and 42 instances, respectively. All instances with 6–12 departments were solved to optimality by both *M1'* and *M1'+*. In addition, *M1'* was able to solve one instance with n = 14, and this is comparable to the base model *M1* that could not solve any instance with n = 14 (see Table 3.4). Similarly, *M1'+* optimally solved two instances with n = 14, and this exhibits an improvement from *M1+* that could find an optimal solution for only one instance. Again, no problem instances with n = 16 were solved to optimality. The average solution times for *M1'*, *M1'+*, *M1*, and *M1+* are displayed in Table 3.8 for each group of the

smaller-sized problem instances. Note that, in this table, the average solution time was taken over all 10 instances for n = 12 unlike that in Table 3.3.

Table 3.7: Comparison of OFVs and OGs of larger-sized instances with SBC3

		OFV			OG		
n	Inst.	*M0+*	*M1+*	*M2+*	*M0+*	*M1+*	*M2+*
14	14_1	36,587.4	36,412.9	**36,356.6**	16.20%	**13.68%**	14.60%
	14_2	38,064.5	37,755.2	**37,583.2**	29.96%	18.47%	18.11%
	14_3	42,127.7	**41,613.7**	**41,613.7**	26.65%	**16.63%**	18.51%
	14_4	45,942.1	45,927.6	**45,895**	22.76%	18.64%	**18.60%**
	14_5	41,496.5	**41,009.8**	41,153.8	36.63%	28.34%	**25.99%**
	14_6	**46,830.4**	**46,830.4**	**46,830.4**	**0.00%**	**0.00%**	**0.00%**
	14_7	39,734.6	39,886.	**39,718.9**	36.59%	**34.43%**	51.77%
	14_8	**43,441.6**	**43,441.6**	43,774.2	38.97%	**29.17%**	37.01%
	14_9	37,344.4	37,321.7	**37,321.7**	13.95%	18.04%	**8.38%**
	14_10	**46,417.9**	46,673.5	46,561.6	20.03%	18.53%	**17.56%**
16	16_1	**100,115**	100,806.8	100,768.1	**65.66%**	77.47%	76.23%
	16_2	**62,338**	63,373.3	62,408.7	54.60%	53.62%	**51.08%**
	16_3	74,053	73,707.8	**73,509.2**	**46.98%**	65.13%	54.78%
	16_4	**94,184.4**	91,938.6	91,271.4	**64.99%**	76.35%	70.78%
	16_5	91,400	**90,562.5**	90,596.6	**48.61%**	60.88%	57.61%
	16_6	64,872.5	64,850.7	**63,905.9**	54.51%	**52.83%**	56.38%
	16_7	**64,871.3**	65,569.1	65,225	**42.78%**	50.33%	56.32%
	16_8	82,129.9	81,001.8	**80,830.4**	**61.03%**	66.53%	72.70%
	16_9	85,602.7	**84,325.8**	85,427.7	60.65%	**58.08%**	71.85%
	16_10	56,828.7	56,484	**56,465.8**	48.67%	**48.39%**	53.77%

Table 3.8: Comparison of average solution times for variants of *M1* with SBC4

	Average solution time (sec.)			
n	*M1'*	*M1'+*	*M1*	*M1+*
6	**0.6**	1.2	0.8	0.9
8	2.5	4.1	2.6	**1.4**
10	14.9	18.4	18.1	**11.5**
12	**233.2**	299.7	558.6	458.9

Table 3.9: Comparison of OFVs and OGs for larger-sized instances for variants of *M1* with SBC4

		OFV		OG	
n	Inst.	*M1'*	*M1'+*	*M1'*	*M1'+*
14	14_1	36,426.4	**36,356.6**	10.51%	**6.83%**
	14_2	37,583.2	**37,567.1**	18.52%	**11.14%**
	14_3	41,888.7	**41,613.7**	19.29%	**12.64%**

	14_4	45,895.0	**45,700.1**	**13.38%**	14.35%
	14_5	**41,009.8**	**41,009.8**	22.54%	**21.96%**
	14_6	**46,830.4**	**46,830.4**	**0.00%**	**0.00%**
	14_7	**39,718.9**	39,734.6	28.24%	**22.54%**
	14_8	**43,153.3**	**43,153.3**	25.30%	**23.47%**
	14_9	**37,321.7**	**37,321.7**	2.05%	**0.00%**
	14_10	**46,121.1**	**46,121.1**	15.91%	**12.89%**
16	16_1	**100,287.5**	100,450.2	71.88%	**64.36%**
	16_2	**63,671.9**	63,889.5	**48.42%**	51.26%
	16_3	74,129.8	**73,973.6**	40.39%	**37.10%**
	16_4	**91,236.4**	91,719.4	**56.54%**	67.95%
	16_5	91,418.5	**91,320.6**	**48.93%**	51.24%
	16_6	64,413.2	**63,538.2**	61.56%	**45.65%**
	16_7	65,710.6	**65,225.0**	46.12%	**43.48%**
	16_8	**81,844.1**	82,608.3	69.82%	**67.24%**
	16_9	**84,798.0**	85,080.3	**56.46%**	67.78%
	16_10	**56,484.0**	57,083.8	45.78%	**41.39%**

Note that *M1'* resulted in smaller average solution times for all problem instance groups compared to *M1'*+. When comparing *M1* and *M1'*, the latter outperformed the former for all groups by at least 3.85% (n = 8) and at most 58.25% (n = 12). On the other hand, *M1'*+ outperformed *M1*+ only for n = 12 with 46.35% improvement in the average solution time. The performance of *M1'* and *M1'*+ for the larger-sized instances are reported in Table 3.9. Out of the 20 larger-sized instances, *M1'* and *M1'*+ produced the least BOFVs in 12 and 13 cases, respectively, with five ties. When compared to *M1* and *M1*+, *M1'* and *M1'*+ yielded the same or better solutions for 17 and 14 instances out of 20, respectively. Also, an optimal solution of instance 14_9 was found by *M1'*+ whereas all other formulations were unable to reach optimality within the time limit. The average OGs when n = 14 are 15.57% (*M1'*) and 12.58% (*M1'*+). For n = 16, the average OGs are 54.59% (*M1*') and 52.57% (*M1'*+). Observing the percentage improvement in the average solution times for the smaller-sized instances and the number of cases of improved BOFVs after the 1-hour limit for the larger-sized instances, adding SBC4

appears to be promising in reducing the computational burden for solving DRLP-C via *M1*. In particular, *M1'+* is recommended for solving relatively larger-sized problems based on the observation of the computational results.

3.4.3 Gurobi presolver disabled with symmetry-breaking constraints

Next, *M0+*, *M1+*, and *M2+* formulations of all 60 problem instances were solved with the Gurobi presolver disabled. Although disabling the presolver is not advised in practice, this test is to observe the performance of different MILP formulations without the intervention of the presolver. As mentioned earlier, enabling the presolver may reduce the problem size with the intention of eliminating redundancy and the overall complexity while maintaining the equivalency between the original formulation and the presolved problem. Hence, the effectiveness of the presolver may depend on the structure of the original formulation.

Among those 60 problem instances, 40 were solved to optimality by at least one of the MILPs within the 1-hour time limit. All 40 problem instances with 6–12 departments were solved to optimality by *M1+*. On the other hand, *M0+* failed to find an optimal solution for one instance, 12_10. Table 3.10 compares the performance of three formulations with respect to the Gurobi presolver being enabled and disabled. It is noteworthy that disabling the presolver did not necessarily result in a worse performance.

Table 3.10: Comparison of average solution times with respect to the presolver

	Average solution time (sec.) – presolver enabled			Average solution time (sec.) – presolver disabled			Percentage difference		
n	*M0+*	*M1+*	*M2+*	*M0+*	*M1+*	*M2+*	*M0+*	*M1+*	*M2+*
6	0.9	0.9	**0.6**	1.1	0.9	**0.6**	22.22%	0.00%	0.00%
8	2.2	**1.4**	**1.4**	3.5	**1.6**	**1.6**	59.09%	14.29%	14.29%
10	9.6	**11.5**	11.7	14.9	10.1	**8.5**	55.21%	-12.17%	-27.35%
12†	613.8	182.9	**143.1**	302.2	**155**	165.4	-50.77%	-15.25%	15.58%

The average solution times of *M1+* and *M2+* did not change after disabling the presolver when n = 6, while that of *M0+* was slightly increased. For problem instances with n = 8, the average solution times were increased by 59.09% (*M0+*), 14.29% for (*M1+*), and 14.29% (*M2+*). When n = 10, the average solution time of *M0+* was increased by 55.21%. However, the average solution time of *M1+* and *M2+* were decreased by 12.17% and 27.53%, respectively. When n = 12, only *M1+* was able to solve all 10 instances to optimality. Interestingly, *M0+* experienced the most drastic reduction in the average solution time (50.77%) when n = 12, which was followed by *M1+* (15.25%). *M1+* and *M2+* outperformed *M0+* and displayed a similar performance for these problem instances with the difference of average solution times between them being relatively small (10.4 seconds).

The BOFVs and OGs for n = 14 and 16 in Table 3.11. Out of 20 larger-sized problem instances, *M0+*, *M1+*, and *M2+* produced the least BOFVs for 2, 13, and 10 instances, respectively. All three formulations produced the same BOFV for one instance (14_9). *M0+* and *M1+* found an optimal solution for one instance with 14 departments (instance 14_6) in 2,718.3 and 2,850.2 seconds, respectively, whereas *M2+* failed to do so within the 1-hour limit. Besides, *M1+* and *M2+* resulted in the same BOFV for two instances (instances 14_8 and 16_2). Excluding instance 14_6 for which *M0+* and *M1+* achieved optimality, *M0+*, *M1+*, and *M2+* displayed the least OGs in 6, 9 and 4 instances, respectively. For n = 14, the average OGs are 27.47% (*M0+*), 24.47% (*M1+*), and 28.62% (*M2+*). The OGs for *M0+*, *M1+*, and *M2+* were increased by 13.65%, 24.91%, and 35.96%, respectively, after disabling the presolver. For n = 16, the average OGs are 62.70% (*M0+*), 60.23% (*M1+*), and 66.00% (*M2+*). The average OG for *M1+*

was decreased by 1.20% while it was increased by 14.31% and 6.19% for *M0+* and *M2+*, respectively, by disabling the presolver.

Table 3.11: Comparison of OFVs and OGS of larger-sized instances with presolver disabled

		OFV			OG		
n	Inst.	*M0+*	*M1+*	*M2+*	*M0+*	*M1+*	*M2+*
14	14_1	36,412.9	**36,356.6**	36,426.4	**12.24%**	15.01%	18.51%
	14_2	37,728.2	**37,567.1**	37,871.8	27.23%	**19.71%**	31.17%
	14_3	42,176.7	**41,848.3**	41,888.7	27.97%	26.68%	**23.61%**
	14_4	46,675.3	46,032.8	**45,895.0**	26.40%	**20.12%**	24.97%
	14_5	41,633.6	**41,043.3**	41,326.2	43.14%	**35.48%**	46.19%
	14_6	**46,830.4**	**46,830.4**	46,843.7	**0.00%**	**0.00%**	11.97%
	14_7	40,226.2	40,567.6	**40,401.5**	61.42%	55.84%	**52.13%**
	14_8	43,536.6	**43,496.4**	**43,496.4**	49.90%	**31.73%**	36.67%
	14_9	**37,321.7**	**37,321.7**	**37,321.7**	**6.93%**	20.38%	18.80%
	14_10	46,417.9	46,491.3	**46,220.1**	**19.50%**	19.76%	22.20%
16	16_1	100,399.7	100,487.7	**99,110.2**	79.24%	76.62%	**76.09%**
	16_2	62,671.5	**62,231.1**	**62,231.1**	69.46%	**55.96%**	65.82%
	16_3	74,226.6	74,399.2	**73,731.9**	**42.26%**	42.91%	61.92%
	16_4	92,101.2	**91,435.5**	93,414.9	**67.02%**	75.76%	72.12%
	16_5	91,980.0	92,154.8	**90,508.9**	**54.17%**	72.08%	64.04%
	16_6	64,340.1	**63,905.9**	64,093.8	56.14%	**53.26%**	63.09%
	16_7	66,037.3	**64,773.9**	65,571.0	55.34%	**48.32%**	56.34%
	16_8	82,822.5	82,375.8	**81,692.0**	74.98%	**63.99%**	71.43%
	16_9	85,322.6	**85,283.3**	85,588.4	74.33%	67.01%	**65.86%**
	16_10	57,203.2	**56,465.8**	56,617.1	54.04%	**46.42%**	63.32%

Table 3.12: Comparison of average solution times for *M1'+* with and without the presolver

	Average solution time (sec.) – presolver enabled	Average solution time (sec.) – presolver disabled	Percentage difference
n	*M1'+*	*M1'+*	*M1'+*
6	1.2	**0.6**	-50.00%
8	4.1	**2.0**	-51.22%
10	18.4	**7.9**	-57.07%
12	299.7	**247.3**	-17.48%

In addition, *M1'+* was tested on the 60 problem instances with the Gurobi presolver disabled. Recall that *M1'+* solved 42 instances with the presolver enabled. When the presolver was disabled, 41 instances were solved to optimality, including all 40

instances with 6–12 departments and one with 14 departments. Table 3.12 reports the performance of *M1'*+ with respect to the setting of the Gurobi presolver. Note that disabling the presolver further reduced the average solution time of *M1'*+ for all problem instances with 6–10 departments. The OFVs and OGs for n = 14 and 16 are reported in Table 3.13.

Table 3.13: Comparison of OFVs and OGs of larger-sized instances for *M1'*+ with and without the presolver

		Presolver enabled		Presolver disabled	
		OFV	OG	OFV	OG
n	Inst.	*M1'*+	*M1'*+	*M1'*+	*M1'*+
14	14_1	36,356.6	**6.83%**	**36,426.4**	14.39%
	14_2	**37,567.1**	**11.14%**	37,823.6	19.22%
	14_3	**41,613.7**	**12.64%**	**41,613.7**	25.63%
	14_4	**45,700.1**	**14.35%**	45,927.6	19.40%
	14_5	**41,009.8**	**21.96%**	**41,009.8**	24.43%
	14_6	**46,830.4**	**0.00%**	**46,830.4**	**0.00%**
	14_7	39,734.6	**22.54%**	**39,671.2**	51.81%
	14_8	**43,153.3**	**23.47%**	43,362.6	30.43%
	14_9	**37,321.7**	**0.00%**	37,344.4	9.44%
	14_10	**46,121.1**	**12.89%**	**46,121.1**	18.02%
16	16_1	**100,450.2**	**64.36%**	102,966.6	75.97%
	16_2	63,889.5	**51.26%**	**63,504.3**	55.45%
	16_3	**73,973.6**	**37.10%**	75,190.8	41.90%
	16_4	**91,719.4**	67.95%	93,649.6	**64.78%**
	16_5	91,320.6	51.24%	**90,355.4**	**44.04%**
	16_6	63,538.2	**45.65%**	**63,649.2**	47.78%
	16_7	65,225.0	43.48%	**64,819.9**	**43.02%**
	16_8	82,608.3	**67.24%**	**81,419.7**	69.26%
	16_9	85,080.3	67.78%	**83,897.1**	**51.59%**
	16_10	57,083.8	**41.39%**	**56,551.0**	42.63%

For most instances with n = 14 (9 out of 10), disabling the presolver did not improve BOFVs. On the other hand, disabling the presolver helped improve the BOFVs in six instances with n = 16. All OGs were increased slightly when n = 14, while only five out of 10 instances with n = 16 displayed an increase in OGs. Overall, there was no strong evidence that disabling the presolver would improve the computational time.

However, the mixed results in the BOFVs of problem instances with n = 16 suggest that disabling the presolver may be worth trying when an approximated solution is sought for larger-sized problems in practice.

3.5 Conclusion

This study considers DRLP-C, which is the problem of assigning a given number of departments along two rows that are separated by a central corridor in order to minimize the total material flow cost, while enforcing minimum clearances between departments. Two new MILP formulations are proposed and thoroughly compared with the model of Zhang and Murray (2012). In addition, three SBCs are introduced in an effort to improve the computational performance. The results from a series of experiments suggest that the proposed MILP formulations along with SBCs are promising in alleviating the computational burden of solving DRLP-C when compared to the model of Zhang and Murray. Extending the problem to multiple rows as well as to two-dimensional layout problems can be addressed as future directions in this line of research.

CHAPTER 4: RECONFIGURATION OF RECTANGULAR RESTAURANT LAYOUT UNDER SOCIAL DISTANCING REQUIREMENTS

4.1 Introduction

Many restaurant owners across the United States are recovering from the recent global pandemic, namely COVID-19, that forced many facilities to temporarily close or operate at a lower scale. In an effort to mitigate the impact of the pandemic, the Centers for Disease Control and Prevention (CDC) prepared a list of guidelines to assist restaurant owners in making decisions with regard to restaurant operations (CDC, 2020). In addition to these guidelines, state and local governments are enforcing policies that limit the occupancy of restaurants to practice social distancing, which is found to be one of the most effective ways to reduce a risk of infection (Chu et al., 2020).

In particular, on May 20th, 2020, Governor Roy Cooper of North Carolina issued Executive Order No. 141 that permits restaurants to accommodate customers for on-site food and beverage consumption (Cooper, 2020). Similar to the guidelines of CDC, the executive order requires restaurants to abide by the following rules with respect to social distancing: (1) limit the number of customers in indoor and outdoor seating areas to no more than 50% of the stated fire capacities, (2) limit the number of people so that groups are at least six feet apart, (3) limit the number of customers at tables so that no more than ten people are seated together unless they belong to the same household, (4) employees are strongly encouraged to wear facemasks when they are within six feet of another person, and (5) mark six feet of spacing in high-traffic areas for customers. In order for restaurant owners to comply with these requirements, it will be necessary for them to rearrange the layout of the dining area, preferably in a way that optimizes the utilization of the available space. A decision is made on the assortment and the arrangement of the

objects of a restaurant (i.e., tables, booths, bar, etc.). The assortment decision is to determine which objects are placed in the restaurant, while the arrangement decision is to determine where those objects are placed within the restaurant. Note that this problem can be considered as a variant of FLP that arranges a set of non-overlapping departments within the facility boundaries while optimizing some relevant objectives such as material flow costs and adjacency preferences (e.g., Montreuil, 1990; Meller, Narayanan and Vance, 1998; Sherali and Fraticelli, 2003; Singh and Sharma, 2005).

In this chapter, a mathematical optimization model is presented to help rearrange tables in the dining area of a restaurant so that the reduced capacity and social distancing constraints are satisfied. The solution to the optimization model will provide the restaurant owner a layout design that complies to the distancing requirement while the performance measure is maximized. Initially, the number of customers that can be seated in the dining area is considered as the performance measure. The concept of the clearance requirements between departments in DRLP from the previous chapter is extended to this work. Unlike the individual clearance between each pair of departments in DRLP, all of the pairwise clearances are equal in the restaurant setting since social distancing remains constant regardless of the pair of objects being addressed. In addition, social distancing between a pair of objects is enforced only if both objects are placed in the restaurant. This is similar to how clearances are represented in DRLP where they are applied only when both departments are placed in the same row.

Noting that the utilization of a table is not necessarily 100% (e.g., a party with three customers is seated in a table with four chairs), the concept of revenue is introduced as a performance measure toward the end of the chapter. Accordingly, following the

presentation of the initial base model, the formulation is then transformed into a two-stage stochastic programming problem for optimizing the restaurant layout design under *static* customer demand scenarios (i.e., parties arrive at the restaurant at the same time for a given time interval) and *dynamic* customer demand scenarios (i.e., parties arrive at the restaurant sequentially over the time interval).

A literature review of relevant restaurant layout studies and efforts that have been made thus far for layout planning in consideration of COVID-19 is presented in Section 4.2. The socially distanced FLP optimization model is described in detail in Section 4.3, and examples with various layouts are presented in Section 4.4 to demonstrate how the optimization generates an optimal assortment and arrangement of tables. Following this, the two-stage stochastic program with recourse model is presented in Section 4.5 in an effort to incorporate the probability distribution of party sizes when optimizing the assortment of seating areas in the restaurant. Numerical results are presented in Section 4.6 to assess the quality of layout solutions that are generated under a simulated restaurant environment. Lastly, final concluding remarks are discussed in Section 4.7.

4.2 Literature review

The restaurant design problem has received attention from researchers across a variety of operational and modeling perspectives. Thompson (2002) considered the mean party size of the restaurant and probability distribution of various party sizes to determine the ideal number of tables inside of a restaurant, with respect to the stated fire capacity, using the Naïve Ideal Table Mix model. Using this model, the author determines the mix of dedicated and flexible tables to be allocated within the restaurant to improve the total revenue by measuring the revenue per available seat hour. Thompson (2003) extended

this work to evaluate the table-mix that yields better profitability by measuring the contribution margin per available seat hour. Kimes and Thompson (2005) applied goal programming to produce feasible integer solutions to the Naïve Ideal Table Mix model (both duration and non-duration based) of Thompson (2002) with respect to restaurant capacity and demand constraints. The base model is extended to a dynamic setting where a table-mix is determined daily and the total revenue over the planning horizon is maximized.

Bertsimas and Shioda (2003) formulate an integer programming model to determine an optimal table-mix by maximizing the total expected revenue and minimizing the waiting time of customers. Fairness constraints are applied to represent a first-come first-serve queuing system that seats customers within the same party size in the order in which they arrive to the restaurant. Their model is formulated and solved each time a customer arrives to determine when (and if) they should be seated. Hwang (2008) developed a simulation model to determine where customers should be seated in consideration of table placement to reduce the overall waiting time. They did not consider the types of tables in their model, which makes it challenging to determine which table placement policy is ideal based on the existing table-mix inside of the restaurant. Hwang et al. (2010) used a queuing-based optimization model where the restaurant congestion and waiting time of customers are represented as a state-dependent quasi birth-and-death process. Four competitive strategies, namely operations, marketing, joint, and fixed price, are proposed to determine the optimal pricing-capacity policy that balances the customer demand and the restaurant capacity. Their computational experiments indicate that the joint competitive strategy is ideal from the perspective of profit-maximization.

Urgulu et al. (2015) presented a GA for generating restaurant layouts to maximize the total revenue and minimize the investment cost. The location of the kitchen, tables, and windows are optimized simultaneously in consideration of staffing and energy costs. The authors suggest that it is advantageous to place most tables closer to the windows to improve customer satisfaction and reduce the lighting cost. Raman and Roy (2015) proposed a queuing theory model for determining the optimal table-mix for restaurants to minimize the customer waiting time and under-utilization of tables. A multi-server queueing system is used where tables with identical capacities belong to the same server, and groups of customers arrive in batches of different sizes. The authors applied a policy in their model where tables are assigned to the largest batch of customers that satisfy its seating capacity.

Bortolete et al. (2021) apply NLP for positioning seats inside of a classroom in consideration of social distancing requirements. A quadratic penalty method is applied to avoid producing infeasible layouts. The model is formulated in a similar fashion as the classic circular-packing problem, where it is of interest to fit a given set of circles with different- or similar-radii into a given region. Several model variants are proposed, including the fixed-seat problem, free-position arrangement problem, and maximizing the number of seats that can be placed using a strategy that is inspired by the work in Birgin and Lobato (2010). Fischetti et al. (2021) propose a similar model as Bortolete et al. and approximate how a virus may spread inside of a space, which is represented as being proportional to distance, modeled as a Gaussian function, and as a steady-state function. Metaheuristics are used for finding a solution to the model. Ugali et al. (2021) formulate an unconstrained NLP using the circular packing representation and propose a heuristic

that initially places seats randomly in the space and iteratively improves the quality of the packing in consideration of social distancing constraints.

To the authors' best knowledge, formulating an optimization model in consideration of reduced capacity and social distancing within the context of layout design and infectious diseases is novel. Furthermore, this work proposes a two-stage stochastic programming model to handle queueing scenarios for optimizing the average revenues as detailed later in this chapter.

4.3 Mathematical optimization model

In this section, a mathematical optimization model as a MILP is presented. The model aims at finding an optimal arrangement of tables that maximizes the total number of seats while satisfying the reduced capacity and social distancing constraints. Later in Section 4.5, this model will be extended to stochastic optimization models to incorporate the revenue of the restaurant under stochastic scenarios. The layout considered in this study is assumed to be a rectangular shape, which is commonly adopted in facility designs. Let the width and length of a restaurant be X and Y, respectively. Furthermore, assume that all objects of the restaurant, including dining tables and fixtures, can be represented or approximated by rectangular shapes with respective widths and lengths.

For mathematical representation of the problem, assume that the restaurant is placed on a two-dimensional Cartesian coordinate system, where the bottom-left corner point of the restaurant is the origin. Suppose that the dimensions of each table include the space of chairs assigned to the table such that the customers can sit comfortably while dining at the table. In conventional facility layout problems, the distance between rooms/spaces is typically defined as the rectilinear distance between the respective center

points when rectangular shape objects are arranged in the facility. However, the rectilinear distance is not adequate for enforcing the social distance since using the rectilinear distance between two edge points could result in violating the social distance requirement. For example, observe two tables i and j in Figure 4.1, which shows the Euclidean distance $\overline{AB}$ shorter than the rectilinear distance $\overline{AC} + \overline{BC}$. In result, although $\overline{AC} + \overline{BC}$ would be longer than the social distance, the actual distance $\overline{AB}$ can be smaller than the required distance. In this study, therefore, the distance between tables is approximated by the Chebyshev distance of two closest points of two objects as defined below.

Definition 4.1 (Chebyshev distance between two rectangles): Let $S_1 \subset R^2$ and $S_2 \subset R^2$ be disjoint rectangles in R^2. The distance between S_1 and S_2 is defined as

$$\min\left\{\max_{s\in\{x,y\}} |u_s - v_s| : (u_x, u_y) \in S_1, (v_x, v_y) \in S_2\right\}.$$

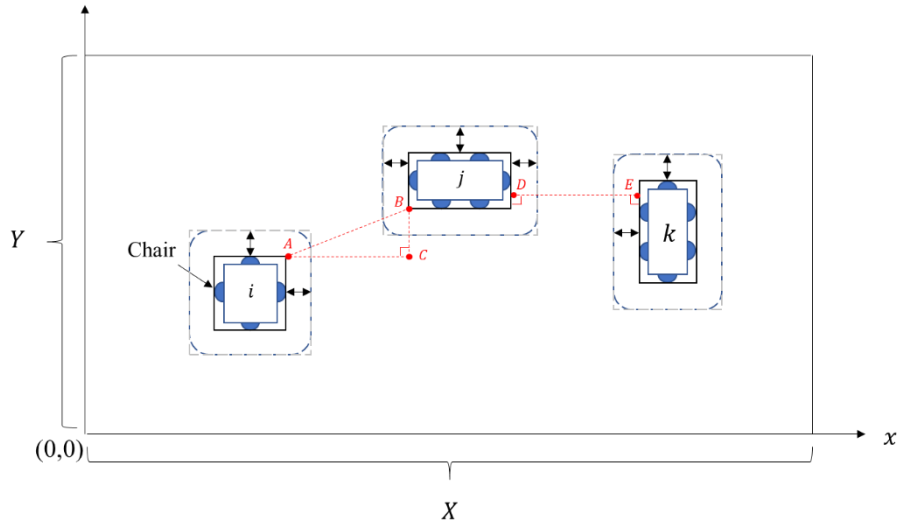


Figure 4.1: Representation of the layout and distance of tables on the Cartesian coordinate system

In Figure 4.1, the Euclidean distance between table i and table j is the length of $\overline{AB}$ while the Chebyshev distance is the length of $\overline{AC}$. Between tables j and k, both Euclidean and

Chebyshev distances are the same as the length of $\overline{DE}$. Not only is this distance metric advantageous for improving the computational tractability, but it also prevents irregular walking paths from being generated relative to the Euclidean distance, which might result in tables being placed in sequences that are not aligned straight. This can be inconvenient for occupants to navigate around, so this is another motivation for not applying the Euclidean distance.

To set up the problem, consider three types of objects of the restaurant: tables, booths, and fixed areas. First, there are n_T tables that would have been used in the restaurant under normal circumstances. Due to the pandemic, only a subset of these tables will be selected to be placed in the dining area. While these tables are freely movable, it is assumed that they can be rotated by only 90 degrees in the proposed problem. Also, suppose that there are n_B booths that are immovable, and hence, some or all of them may not be used for dining to comply with the distancing requirements even if they stay in the restaurant. Note that, if a particular booth is not selected to be used, then distancing requirements are not applicable from this booth to other tables/booths. In addition, assume that there are n_F fixed areas where tables cannot be placed. Examples of fixed areas include the ingress/egress, designated walkways, and immovable fixtures in the restaurant such as reception, kitchen appliances, bar area, etc.

Let $I = \{1,2, \dots, n_T, n_T + 1, n_T + 2, \dots, n_T + n_B, n_T + n_B + 1, n_T + n_B + 2, \dots, n_T + n_B + n_F\}$ denote the index set of three types of objects. For the sake of convenience in presenting formulations, partition I into three subsets: $T = \{1, \dots, n_T\}$, $B = \{n_T + 1, \dots, n_T + n_B\}$, and $F = \{n_T + n_B + 1, \dots, n_T + n_B + n_F\}$, which represent index sets for tables, booths, and fixed areas, respectively. Each object in I may vary in

width and length, and each table/booth in $T \cup B$ can accommodate a certain number of chairs (i.e., customers). Although six feet is the current guideline for social distancing, different distances may be implemented if necessary, depending on other factors that influence the chance of transmission (Chu et al., 2020). In general, assume that tables are required to be separated by at least d_{min} in this problem. Furthermore, suppose that the restaurant has a stated fire capacity of c, and the capacity of the restaurant during a pandemic event is set to be a fraction of c (i.e., αc, where $\alpha \in (0,1]$). The parameters and decision variables that are used for the MILP formulation are summarized as follows.

I	Index set of objects ($I = T \cup B \cup F$)
T	Index set of table objects ($T = \{1, \dots, n_T\}$)
B	Index set of booth objects ($B = \{n_T + 1, \dots, n_T + n_B\}$)
F	Index set of fixed area objects ($F = \{n_T + n_B + 1, \dots, n_T + n_B + n_F\}$)
c	Stated fire capacity of the restaurant
p_i	Number of customers that table/booth $i \in T \cup B$ can accommodate
w_i / l_i	Width/length of object $i \in I$
X/Y	Dimensions of the restaurant along the x-/y-axis
d_{min}	Minimum separation for social distancing requirements
b_{ij}	A binary parameter equal to 1 if the distance between booths i and j is greater than or equal to d_{min} for $i, j \in B, i < j$
(x_i, y_i)	Variables representing the coordinates of the center point of table $i \in T$ (x_i and y_i for $i \in B \cup F$ are constants since object i is immovable)
d_{ij}^s	Variable related to enforcing social distancing requirements between table i and table/booth j along the s-dimension for $s \in \{x, y\}$, $i \in T, j \in T \cup B, i < j$
t_i	Binary variable equal to 1 if table $i \in T$ is placed in the restaurant; 0 otherwise ($t_i = 1$ for $i \in B \cup F$ since object i is immovable)
b_i	Binary variable equal to 1 if seating in booth $i \in B$ is allowed; 0 otherwise

o_i Binary variable equal to 1 if the width face of table $i \in T$ is parallel to the x-axis; 0 otherwise
($o_i = 1$ for $i \in B \cup F$ since object i is immovable)

z_{ij}^s (z_{ji}^s) Binary variable equal to 1 if table i (object j) is forced to precede object j (table i) along the s-dimension for $s \in \{x, y\}$, $i \in T, j \in I, i < j$; 0 otherwise.

When table $i \in T$ is not placed in the restaurant, its location and orientation will have default values $(x_i, y_i) = (0,0)$ and $o_i = 0$. Using the notation introduced above, the proposed MILP, namely social distancing layout problem (SDLP) can be written as follows.

SDLP:

Maximize $\sum_{i \in T} p_i t_i + \sum_{i \in B} p_i b_i$ (4.1)

subject to

$$\sum_{i \in T} p_i t_i + \sum_{i \in B} p_i b_i \leq \alpha c \quad (4.2)$$

$$o_i \leq t_i \qquad \forall i \in T \quad (4.3)$$

$$x_i + \frac{l_i}{2} t_i + \left(\frac{w_i - l_i}{2}\right) o_i \leq x_j - \frac{l_j}{2} t_j - \left(\frac{w_j - l_j}{2}\right) o_j + M\left(1 - z_{ij}^x\right) \qquad \forall i \in T, j \in I \quad (4.4)$$

$$x_j + \frac{l_j}{2} t_j + \left(\frac{w_j - l_j}{2}\right) o_j \leq x_i - \frac{l_i}{2} t_i - \left(\frac{w_i - l_i}{2}\right) o_i + M\left(1 - z_{ji}^x\right) \qquad \forall i \in T, j \in I \quad (4.5)$$

$$y_i + \frac{w_i}{2} t_i + \left(\frac{l_i - w_i}{2}\right) o_i \leq y_j - \frac{w_j}{2} t_j - \left(\frac{l_j - w_j}{2}\right) o_j + M(1 - z_{ij}^y) \qquad \forall i \in T, j \in I \quad (4.6)$$

$$y_j + \frac{w_j}{2} t_j + \left(\frac{l_j - w_j}{2}\right) o_j \leq y_i - \frac{w_i}{2} t_i - \left(\frac{l_i - w_i}{2}\right) o_i + M\left(1 - z_{ji}^y\right) \qquad \forall i \in T, j \in I \quad (4.7)$$

$$x_i + \left(\frac{w_i - l_i}{2}\right) o_i \leq \left(X - \frac{l_i}{2}\right) t_i \qquad \forall i \in T \quad (4.8)$$

$$y_i + \left(\frac{l_i - w_i}{2}\right) o_i \leq \left(Y - \frac{w_i}{2}\right) t_i \qquad \forall i \in T \quad (4.9)$$

$$x_i \geq \frac{l_i}{2} t_i + \left(\frac{w_i - l_i}{2}\right) o_i \qquad \forall i \in T \quad (4.10)$$

$$y_i \geq \frac{w_i}{2} t_i + \left(\frac{l_i - w_i}{2}\right) o_i \qquad \forall i \in T \qquad (4.11)$$

$$d_{ij}^x \geq \left(x_j - \frac{l_j}{2} t_j - \left(\frac{w_j - l_j}{2}\right) o_j\right) - \left(x_i + \frac{l_i}{2} t_i + \left(\frac{w_i - l_i}{2}\right) o_i\right) \qquad i \in T, j \in T \cup B, \; i < j \qquad (4.12)$$

$$d_{ij}^x \geq \left(x_i - \frac{l_i}{2} t_i - \left(\frac{w_i - l_i}{2}\right) o_i\right) - \left(x_j + \frac{l_j}{2} t_j + \left(\frac{w_j - l_j}{2}\right) o_j\right) \qquad i \in T, j \in T \cup B, \; i < j \qquad (4.13)$$

$$d_{ij}^y \geq \left(y_j - \frac{w_j}{2} t_j - \left(\frac{l_j - w_j}{2}\right) o_j\right) - \left(y_i + \frac{w_i}{2} t_i + \left(\frac{l_i - w_i}{2}\right) o_i\right) \qquad i \in T, j \in T \cup B, \; i < j \qquad (4.14)$$

$$d_{ij}^y \geq \left(y_i - \frac{w_i}{2} t_i - \left(\frac{l_i - w_i}{2}\right) o_i\right) - \left(y_j + \frac{w_j}{2} t_j + \left(\frac{l_j - w_j}{2}\right) o_j\right) \qquad i \in T, j \in T \cup B, \; i < j \qquad (4.15)$$

$$d_{ij}^x \leq \left(x_j - \frac{l_j}{2} t_j - \left(\frac{w_j - l_j}{2}\right) o_j\right) - \left(x_i + \frac{l_i}{2} t_i + \left(\frac{w_i - l_i}{2}\right) o_i\right) + M(1 - z_{ij}^x) \qquad i \in T, j \in T \cup B, \; i < j \qquad (4.16)$$

$$d_{ij}^x \leq \left(x_i - \frac{l_i}{2} t_i - \left(\frac{w_i - l_i}{2}\right) o_i\right) - \left(x_j + \frac{l_j}{2} t_j + \left(\frac{w_j - l_j}{2}\right) o_j\right) + M(1 - z_{ji}^x) \qquad i \in T, j \in T \cup B, \; i < j \qquad (4.17)$$

$$d_{ij}^y \leq \left(y_j - \frac{w_j}{2} t_j - \left(\frac{l_j - w_j}{2}\right) o_j\right) - \left(y_i + \frac{w_i}{2} t_i + \left(\frac{l_i - w_i}{2}\right) o_i\right) + M(1 - z_{ij}^y) \qquad i \in T, j \in T \cup B, \; i < j \qquad (4.18)$$

$$d_{ij}^y \leq \left(y_i - \frac{w_i}{2} t_i - \left(\frac{l_i - w_i}{2}\right) o_i\right) - \left(y_j + \frac{w_j}{2} t_j + \left(\frac{l_j - w_j}{2}\right) o_j\right) + M(1 - z_{ji}^y) \qquad i \in T, j \in T \cup B, \; i < j \qquad (4.19)$$

$$d_{ij}^s \geq d_{min}(z_{ij}^s + z_{ji}^s + t_i + t_j - 2) \qquad \forall i, j \in T, \; i < j, s \in \{x, y\} \qquad (4.20)$$

$$d_{ij}^s \geq d_{min}(z_{ij}^s + z_{ji}^s + t_i + b_j - 2) \qquad \forall i \in T, j \in B, \; s \in \{x, y\} \qquad (4.21)$$

$$b_i + b_j \leq b_{ij} + 1 \qquad \forall i, j \in B, \qquad (4.22)$$

	$i < j$	
$z_{ij}^{x} + z_{ji}^{x} + z_{ij}^{y} + z_{ji}^{y} = 1$	$\forall i \in T, j \in I,$	(4.23)
	$i < j$	
$x_i, y_i \geq 0$	$\forall i \in T$	(4.24)
$d_{ij}^{s} \geq 0$	$\forall i \in T, j$	(4.25)
	$\in T \cup B,$	
	$i < j, s \in \{x, y\}$	
$t_i, o_i \in \{0,1\}$	$\forall i \in T$	(4.26)
$b_i \in \{0,1\}$	$\forall i \in B$	(4.27)
$z_{ij}^{s}, z_{ji}^{s} \in \{0,1\}$	$\forall i \in T, j \in I,$	(4.28)
	$i < j, s$	
	$\in \{x, y\}.$	

The objective function (4.1) maximizes the number of seats inside the restaurant, which is the weighted sum of the binary variables representing tables and booths being used. Constraint (4.2) enforces the reduced seating capacity of the restaurant. Constraint (4.3) suppresses binary variables $o_i = 0$ when $t_i = 0$. By setting to zero, other constraints are not affected when table i is not placed in the restaurant. Constraints (4.4)-(4.7) describe the locational relationship between center points of two objects so that each pair of objects can avoid overlapping. Note that o_i and the corresponding terms that are multiplied by it can be excluded from (4.4)-(4.7) if table i is square-shaped (i.e., $w_i = l_i$). Constraints (4.8)-(4.11) warrant that the tables are placed within the boundaries of the restaurant. Note that, if t_i = 0 for some table i, we have $o_i = 0$ from (4.3) and in turn, $(x_i, y_i) = (0,0)$ from (4.8)-(4.9). This ensures that (4.4)-(4.7) hold true regardless of object j.

Constraints (4.12)-(4.15) set the lower bounds on d_{ij}^{s}, while (4.16)-(4.19) make one of d_{ij}^{x} and d_{ij}^{y} represent the exact distance along that dimension between the pair i and j. Constraint (4.20) guarantees that social distancing is satisfied by separating tables along the s-dimension if tables i and j are placed inside the restaurant (i.e., $t_i = t_j = 1$). Constraint (4.21) is identical to (4.20) with the exception that social distancing is enforced between a table and a booth. Constraint (4.22) prevents booths i and j from being used simultaneously if social distancing is not satisfied. Constraint (4.23) prevents tables from overlapping one another, as well as booths and fixed areas. Note that d_{ij}^{s} will take on the exact distance between two objects i and j along a single dimension from (4.12)-(4.19) when $z_{ij}^{s} = 1$ or $z_{ji}^{s} = 1$ in (4.23) (i.e., z_{ij}^{x}= 1 or z_{ji}^{x} = 1 provides the exact distance along the x-dimension but not along the y-dimension; this also enforces separation along the x-dimension between this pair of tables/booths). The correctness of the proposed formulation is described in what follows.

Lemma 4.1 (table selection): Consider a feasible solution to SDLP with respect to table object $i \in T$. Then, the following holds true:

1) $t_i = 0$ if and only if $(x_i, y_i) = (0, 0)$.

2) If $t_i = 1$, then the table object i fits in the restaurant.

Proof. For 1), if $t_i = 0$, $(x_i, y_i) = (0, 0)$ follows from (4.8)-(4.9) along with (4.3). Conversely, if $(x_i, y_i) = (0, 0)$, then $t_i = 0$ from (4.10)-(4.11) since $l_i, w_i > 0$. For 2), suppose that $o_i = 1$. From (4.8)-(4.11), $\frac{w_i}{2} \leq x_i \leq X - \frac{w_i}{2}$ and $\frac{l_i}{2} \leq y_i \leq Y - \frac{l_i}{2}$. Hence, the entire table i fits in the restaurant. Similarly, $\frac{l_i}{2} \leq x_i \leq X - \frac{l_i}{2}$ and $\frac{w_i}{2} \leq y_i \leq Y - \frac{w_i}{2}$ when $o_i = 0$. ■

Lemma 4.2 (social distance between tables): Consider a feasible solution to SDLP with respect to two objects $i, j \in T$ such that $t_i = t_j = 1$ and $i < j$. Then, table i and table j are socially distanced by at least d_{min}.

Proof. First, consider a case where $z_{ij}^x = 1$ and $z_{ji}^x = z_{ij}^y = z_{ji}^y = 0$ in (4.23). By (4.12), (4.16) and (4.4), we have $d_{ij}^x = \left(x_j - \frac{l_j}{2} t_j - \left(\frac{w_j - l_j}{2}\right) o_j\right) - \left(x_i + \frac{l_i}{2} t_i + \left(\frac{w_i - l_i}{2}\right) o_i\right) \geq 0$. Note that $d_{ij}^x \geq d_{min}$ from (4.20) with $s = x$. In turn, $d_{ij} \geq d_{ij}^x$, where d_{ij} denotes the distance between tables i and j as defined in Definition 4.1. Therefore, the distance between two tables is at least d_{min}. Similarly, the same argument can be used for other cases with $z_{ji}^x = 1$, $z_{ij}^y = 1$, or $z_{ji}^y = 1$ in (4.23). ■

Lemma 4.3 (social distance between table and booth): Consider a feasible solution to SDLP with respect to two objects $i \in T$ and $j \in B$ such that $t_i = b_j = 1$. Then, table i and booth j are socially distanced by at least d_{min}.

Proof. The same argument as in the proof of Proposition 4.2 (later on) can be used to have $d_{ij}^x = \left(x_j - \frac{l_j}{2} t_j - \left(\frac{w_j - l_j}{2}\right) o_j\right) - \left(x_i + \frac{l_i}{2} t_i + \left(\frac{w_i - l_i}{2}\right) o_i\right) \geq 0$ when $z_{ij}^x = 1$. From (4.21) with $s = x$, we have $d_{ij}^x \geq d_{min}$, and in turn, $d_{ij} \geq d_{ij}^x$, where d_{ij} denotes the distance between table i and booth j. Therefore, $d_{ij} \geq d_{ij}^x \geq d_{min}$. The same argument can be used for other cases with $z_{ji}^x = 1$, $z_{ij}^y = 1$, or $z_{ji}^y = 1$ in (4.23). ■

Lemma 4.4 (social distance between two selected booths): Consider a feasible solution to SDLP with respect to two objects $i, j \in B$ such that $b_i = b_j = 1$. Then, booths i and j are socially distanced by at least d_{min}.

Proof. Constraint (4.22) is satisfied only when $b_{ij} = 1$, which implies booths i and j are socially distanced. ■

Proposition 4.1 (social distance): Objects represented by a feasible solution to SDLP are socially distanced by at least d_{min}.

Proof. It directly follows Lemmas 4.1-4.4. ■

Proposition 4.2: Consider a restaurant layout that complies with the social distancing requirement, where the distance is defined by Definition 4.1. Then, there exists a feasible solution to SDLP that corresponds to the layout.

Proof. The proof is done by showing that a feasible solution can be identified from the given layout. First, consider the objects placed in the given layout and specific values will be assigned to corresponding variables. Denote the set of tables placed in the layout by $\bar{T} \subset T$, and the set of booths where seating is allowed in the layout by $\bar{B} \subset B$. Then, assign the values of x_i, y_i, t_i, o_i, and b_i for $i \in \bar{T} \cup B \cup F$ according to the definitions of the variables. Let a object $i \in \bar{T} \cup \bar{B} \cup F$ be represented as a rectangle $S_i = \left\{(u_x, u_y): x_i - \frac{l_i}{2} t_i - \left(\frac{w_i - l_i}{2}\right) o_i \le u_x \le x_i + \frac{l_i}{2} t_i + \left(\frac{w_i - l_i}{2}\right) o_i, y_i - \frac{w_i}{2} t_i - \left(\frac{l_i - w_i}{2}\right) o_i \le u_y \le y_i + \frac{w_i}{2} t_i + \left(\frac{l_i - w_i}{2}\right) o_i\right\}$. Suppose that the distance between two objects $i, j \in \bar{T} \cup \bar{B} \cup F, i < j$, in the given layout is $\hat{d}_{ij}$ which is yielded by $s^* = \underset{s \in \{x,y\}}{\operatorname{argmax}} |u_s - v_s|$ as defined in Definition 4.1. Then, assign $d_{ij}^x = d_{ij}^y = \hat{d}_{ij}$. Furthermore, if $s_i^* \le s_j^*$, assign $z_{ij}^{s^*} = 1$, $z_{ji}^{s^*} = 0$, and $z_{ij}^{\bar{s}} = z_{ji}^{\bar{s}} = 0$ where $\bar{s} \in \{x, y\} \setminus \{s^*\}$. On the other hand, if $s_i^* > s_j^*$, assign $z_{ji}^{s^*} = 1$, $z_{ij}^{s^*} = 0$, and $z_{ij}^{\bar{s}} = z_{ji}^{\bar{s}} = 0$ where $\bar{s} \in \{x, y\} \setminus \{s^*\}$. This assignment satisfies Constraint (4.23) for $i \in \bar{T}$ and $j \in \bar{T} \cup B \cup F$.

Since $t_i = 1$ for $i \in \bar{T}$ and o_i is set to represent its orientation, Constraints (4.3), (4.8)-(4.11) are satisfied. Also, for $i \in \bar{T}$ and $j \in \bar{T} \cup B$, Constraints (4.12)-(4.15) are satisfied from $d_{ij}^x = d_{ij}^y = \hat{d}_{ij}$ and Definition 4.1. Now, consider $i \in \bar{T}$ and $j \in \bar{T} \cup B \cup F$. Note that

$s^* = x$ and $x_i \leq x_j$ imply $d_{ij}^x = \hat{d}_{ij} = \left[x_j - \frac{l_j}{2} - \left(\frac{w_j - l_j}{2}\right) o_j\right] - \left[x_i + \frac{l_i}{2} + \left(\frac{w_i - l_i}{2}\right) o_i\right]$ and also $z_{ij}^x = 1$ from above. Hence, Constraints (4.4) is satisfied while Constraints (4.5)-(4.7) are redundant. Similarly, $s^* = x$ and $x_i > x_j$ imply $\hat{d}_{ij} = \left[x_i - \frac{l_i}{2} - \left(\frac{w_i - l_i}{2}\right)\right] - \left[x_j + \frac{l_j}{2} t_j + \left(\frac{w_j - l_j}{2}\right) o_j\right]$, hence Constraint (4.5) is satisfied while Constraints (4.4), (4.6)-(4.7) are redundant. The same argument can be applied when $s^* = y$, and thus, Constraints (4.4)-(4.7) are satisfied for all $i \in \bar{T}$ and $j \in \bar{T} \cup B \cup F$. Noting that $\bar{T} \cup B \subset \bar{T} \cup B \cup F$, a similar argument results in Constraints (4.16)-(4.19) being satisfied for $i \in \bar{T}$ and $j \in \bar{T} \cup B$. Constraints (4.20)-(4.21) are trivially satisfied since $d_{ij}^x = d_{ij}^y = \hat{d}_{ij} \geq d_{min}$. Finally, from the definition of b_i and b_{ij}, Constraint (4.22) is satisfied.

Next, consider table $i \in T \setminus \bar{T}$ and assign $x_i = y_i = t_i = o_i = 0$. These values trivially satisfy Constraints (4.3) and (4.8)-(4.11). For the rest of constraints, consider two cases, $j \in T \setminus \bar{T}$ and $j \in \bar{T}$. Case 1: Assign $z_{ij}^x = 1$ (hence, $z_{ij}^y = z_{ji}^x = z_{ji}^y = 0$) and $d_{ij}^x = d_{ij}^y = 0$. Then, Constraints (4.5)-(4.7) and (4.17)-(4.19) become redundant, while Constraints (4.4), (4.12)-(4.15), (4.16), (4.20)-(4.21), and (4.23) are trivially satisfied. Case 2: Assign $d_{ij}^x = x_j - \frac{l_j}{2} t_j - \left(\frac{w_j - l_j}{2}\right) o_j$, $d_{ij}^y = y_j - \frac{w_j}{2} t_j - \left(\frac{l_j - w_j}{2}\right) o_j$, and $z_{ij}^x = 1$ (hence, $z_{ij}^y = z_{ji}^x = z_{ji}^y = 0$). Then, (4.5)-(4.7) and (4.17)-(4.19) become redundant. Furthermore, (4.4) is satisfied since the right-hand-side is nonnegative. Constraints (4.12), (4.14), and (4.16) are also satisfied since $d_{ij}^x = x_j - \frac{l_j}{2} t_j - \left(\frac{w_j - l_j}{2}\right) o_j$, and $d_{ij}^y = y_j - \frac{w_j}{2} t_j - \left(\frac{l_j - w_j}{2}\right) o_j$. Constraints (4.13), (4.15) and (4.20)-(4.21) hold true as the right-hand-sides are nonpositive. In consequence, feasible solutions were identified for all cases, and this completes the proof. ■

A phenomenon that often appears in combinatorial optimization problems is the problem symmetry. The symmetry typically results in longer solution times during the solution process since alternative solutions that yield the same objective function value can be unnecessarily explored (Margot, 2010). For the proposed MILP, the problem symmetry is evident when there are multiple tables with identical dimensions being considered for placement within the restaurant. In an effort to alleviate this burden, SBCs are introduced. Suppose that there are K types of tables, where tables of type $k \in \{1, \dots, K\}$ have the same seating capacity and dimensions. Letting n_T^k represent the number of tables of type k, the index set for tables can be written as $T = \bigcup_{k=1}^{K} T_k$, where $T_k = \left\{\sum_{i=1}^{k-1} n_T^i + 1, \sum_{i=1}^{k-1} n_T^i + 2, \sum_{i=1}^{k} n_T^i\right\}$. Then, the following SBCs can effectively reduce the problem symmetry.

$$x_i \leq x_{i+1} \qquad \forall i \in T_k \setminus \left\{\sum_{i=1}^{k} n_T^i\right\}, k \in \{1, \dots, K\} \qquad (4.29)$$

$$t_i \leq t_{i+1} \qquad \forall i \in T_k \setminus \left\{\sum_{i=1}^{k} n_T^i\right\}, k \in \{1, \dots, K\}. \qquad (4.30)$$

Constraints (4.29)-(4.30) guarantee that identical tables are placed from left to right in ascending order of the table index within the restaurant. In addition to (4.29)-(4.30), the following SBC can be applied as well.

$$z_{ji}^x \leq t_i \qquad \forall i, j \in T, i \neq j \qquad (4.31)$$

$$z_{ji}^x + z_{ji}^y \leq t_i \qquad \forall i \in T, j \in B \cup F. \qquad (4.32)$$

Constraint (4.31) prevents z_{ji}^x from being 1 if table i is not placed within the restaurant (i.e., t_i = 0). That is, table j is not allowed to precede table i along the x-dimension when t_i = 0. Constraint (4.32) is similar to (4.31), with the exception that it also applies to the y-dimension. Due to (4.32), booth/fixed area j is not allowed to precede table i in any dimension if table i is not placed in the restaurant. However, it should be noted that z_{ji}^y

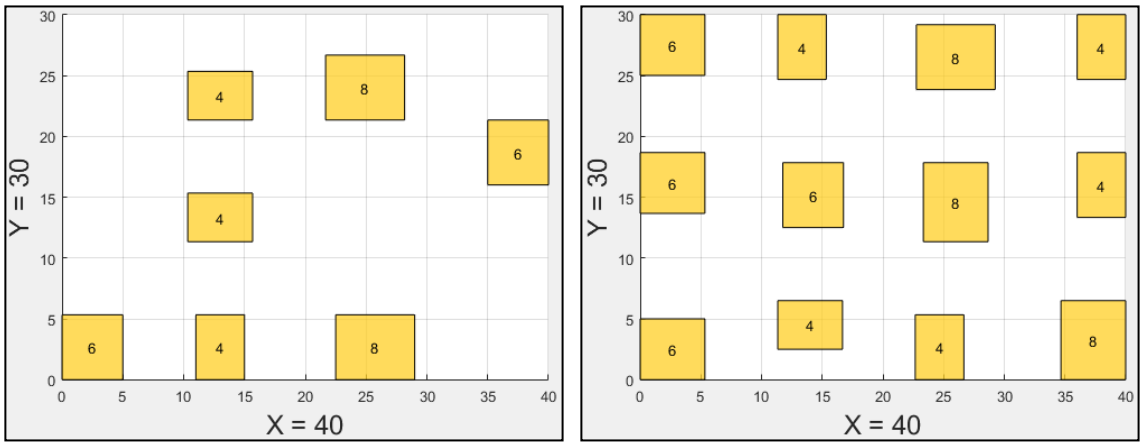
may not be added to the left-hand side of (4.31) unlike that of (4.32) because (4.23) is violated when $t_i = t_j = 0$. However, since $t_i = 1$ for $i \in B \cup F$, the feasibility of (4.23) is still maintained.

4.4 Empirical study

In this section, the proposed model is applied to several restaurants with difference sizes and layouts to demonstrate how an optimal arrangement of tables can be generated. For each case, two values of $\alpha \in \{0.5, 1\}$ are considered to illustrate how the layout design is impacted with and without the reduced capacity constraint. In addition, d_{min} is set equal to 6 feet, which is the social distancing requirement that has been enforced by Cooper (2020). The model is built using Python (3.7.3, 2018) and solved by a commercial optimization solver, Gurobi (8.1.1, 2018), which is run in a server equipped with four processors of Intel Xeon CPU E7-8890 v3 (3.60 GHz) and an OS of Windows Server 2012 R2. A 1-hour time limit is imposed in case an optimal solution could not be found in a reasonable amount of time. In each figure of the resulting restaurant layout that will be presented below, tables are displayed as yellow rectangles with their respective seating capacities in the center and immovable non-seating areas are displayed as black rectangles.

In the first case, a fictitious dining hall that has a width of 40 feet and a length of 30 feet is considered. In total, there are 15 tables, which consist of eight tables having a seating capacity of 4 people with dimensions of 5.333’ x 4’ (width x length; seating space included in the width dimensions), four tables having a seating capacity of 6 people with dimensions of 5.333’ x 5’, and three tables having a seating capacity of 8 people with dimensions of 5.333’ x 6.5’. No immovable fixtures are considered in this case.

According to the 2015 International Fire Code, the occupant load factor for unconcentrated assembly without fixed seats is 15 sq. ft. per person (International Fire Code, accessed 2020). Using this load factor, a stated fire capacity (c) is prescribed as 80 (=1200/15). Two layout designs with and without the capacity limit for this case are displayed in Figure 4.2. Let Z represent the number of seats that are placed inside of the restaurant after the optimization is executed.



(a) reduced capacity (α = 0.5) Z = 40

(b) original capacity (α = 1) Z = 68

Figure 4.2: Layout designs for case 1

Figure 4.2 (a) is an optimal solution when a reduced capacity (α = 0.5) was enforced. The solution was obtained in 2.38 seconds,. Figure 4.2 (b), on the other hand, displays the best solution until the one-hour time limit was reached when a full capacity (α = 1) was applied. Observe, from Figure 4.2 (b), that all tables accommodating 6 and 8 people were selected, and five (out of eight) tables that accommodate 4 people were selected. Note that alternative optimal solutions may exist, especially when enforcing the capacity limit (Figure 4.2 (a)) since some of the tables can be moved while keeping the distancing requirements. For example, three tables that are closest to the upper wall can be shifted upward or the three tables closest to the bottom wall to the right.

The second case is chosen from Fabregas (accessed 2020) and the size of this restaurant is 42.467’ x 23.833’ with c = 49. There are four immovable fixtures to reserve the spaces of the kitchen, reception area, walkway, and bathroom, as shown in Figure 4.3.

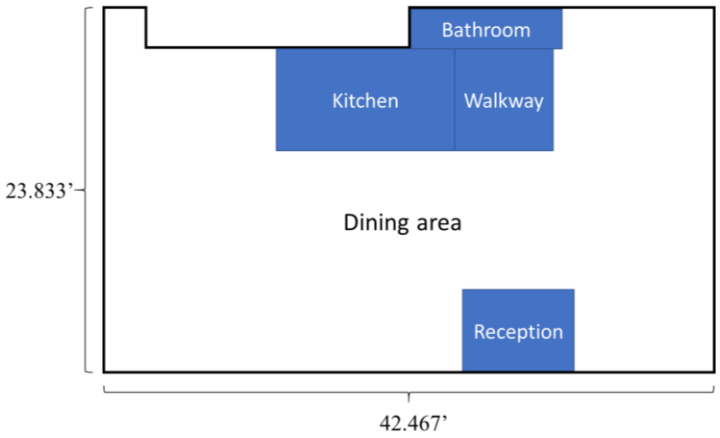
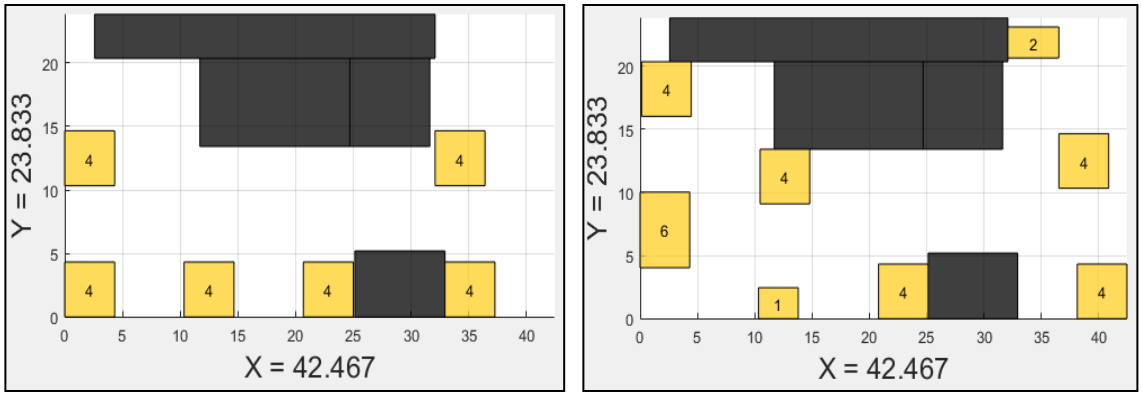


Figure 4.3**:** Dining area of case 2 with four fixed areas

Note that the restaurant is not in a rectangular shape. In order to accommodate the non-rectangularly shaped facility along the upper wall, the fixed area of the bathroom is extended towards the left wall, which is shown as the long black rectangle in the resulting layouts displayed in Figure 4.4. There are a total of 13 tables: one has a seating capacity of 1 person with dimensions of 3.458’ x 2.458’, one has a seating capacity of 2 people with dimensions of 4.583’ x 2.458’, ten have a seating capacity of 4 people with dimensions of 4.333’ x 4.333’, and one has a seating a capacity of 6 people with dimensions of 4.333’ x 6’. Two optimal layout designs with and without the capacity limit for this case are displayed in Figure 4.4.



(a) reduced capacity (α = 0.5) Z = 24

(b) original capacity (α = 1) Z = 29

Figure 4.4: Restaurant layouts of case 2

Optimal solutions were found in 0.25 seconds and 3.13 seconds for α = 0.5 and α = 1, respectively. Similar to the first case, it appears that there is some flexibility in the layout with α = 0.5 as moving the tables around the restaurant is possible while still satisfying the social distancing and reduced capacity constraints. This is also the case for Figure 4.4 (b), but the degree of flexibility is much less in comparison to the layout in Figure 4.4 (a). Note that the layout with α = 0.5 accommodates exactly 50% of c, whereas the layout without the capacity constraint could only achieve 59.18% of c due to social distancing requirements.

The restaurant in the third case is obtained from WebstaurantStore (accessed 2020) and has a size of 62.4’ x 58.133’. Using the aforementioned load factor, c is set to 105. Six fixed areas identified to reserve the spaces of the kitchen, reception area, wall, bar area, and bathroom (include its passage), as shown in Figure 4.5.

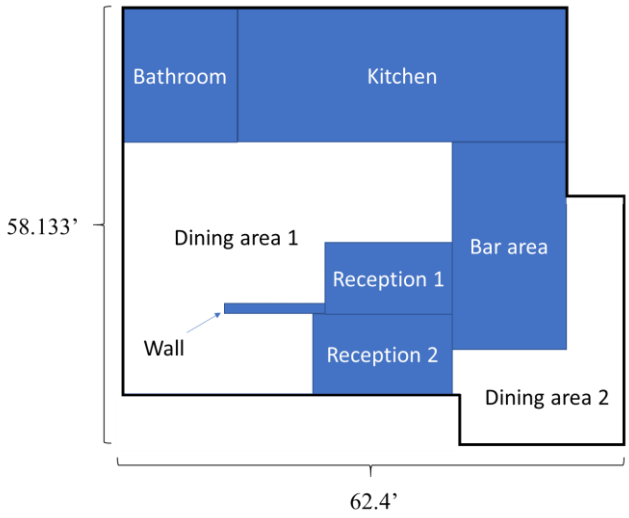
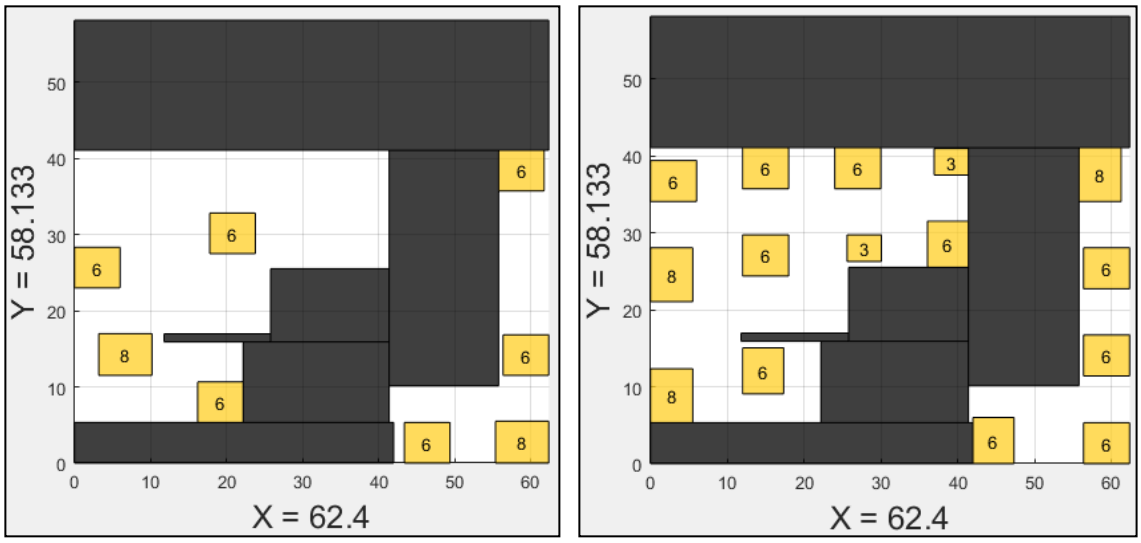


Figure 4.5: Dining area of case 3 with six fixed areas

In this case, it is assumed that customers cannot be seated at the bar area. Unlike the previous two cases, the dining area is split into two areas which are separated by a set of fixed areas. In total, 18 tables are considered for possible placement in the dining area. Three of them have a seating capacity of 3 people with dimensions of 4.458' x 3.458', twelve have a seating capacity of 6 people with dimensions of 5.333' x 6', and three have a seating capacity of 8 people with dimensions of 5.5' x 7'. The layout designs generated from the solutions are displayed in Figure 4.6.



(a) reduced capacity (α = 0.5) Z = 52

(b) original capacity (α = 1) Z = 90

Figure 4.6: Layout designs for case 3

For α = 0.5, an optimal solution was found in 0.34 seconds. When α = 1, however, the solution process was terminated after the one-hour limit when α = 1, and Figure 4.6 (b) is the layout from the best solution up to that point. When relaxing the reduced capacity constraint in the layout when α = 1, the tables are placed in an organized fashion, and the capacity achieves 85.71% of c. Again, the layout for α = 0.5 can be modified by shifting some of the tables into less-utilized regions of the dining areas.

For the fourth case, the layout of a 47.233’ x 29.466’ restaurant obtained from DaGue (accessed 2020) is considered. Similar to the way prescribing the fire capacity in the previous cases, c is set to 52. Five fixed areas are identified to represent the spaces of the kitchen, reception area, televisions, and bathroom, as shown in Figure 4.7.

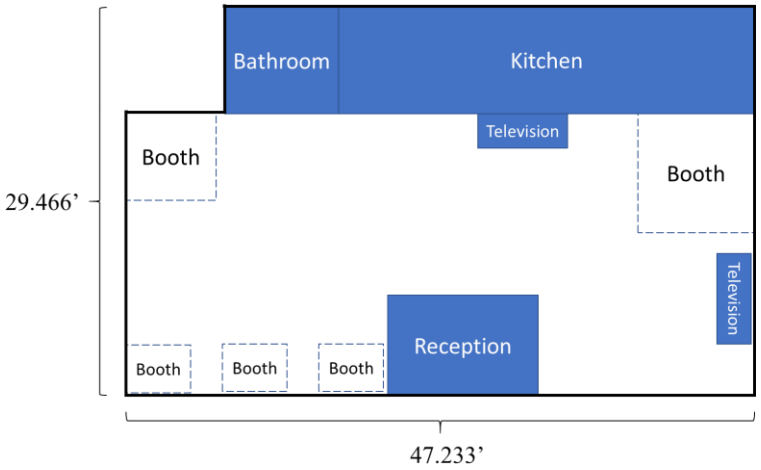
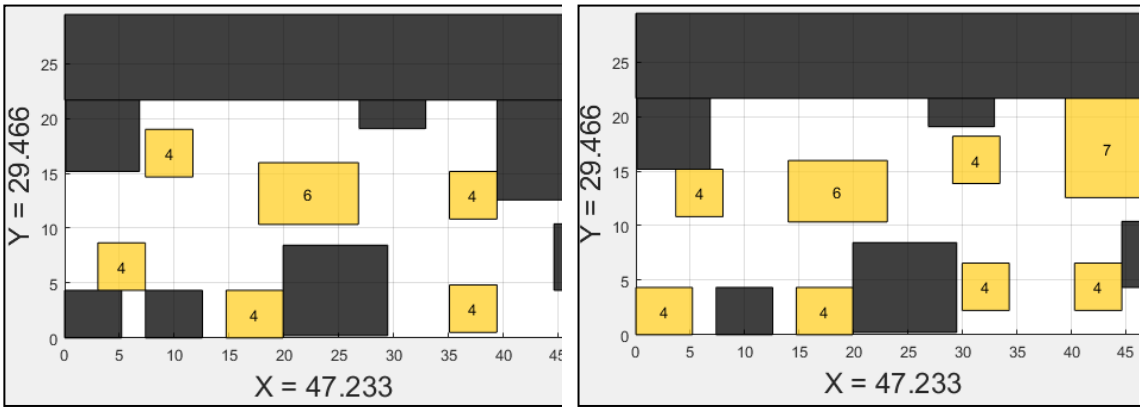


Figure 4.7: Dining area of case 4 with five fixed areas and five immovable booths

In addition, there are five immovable booths in the dining area. One booth has a seating capacity of 3 people with dimensions of 6.716’ x 6.5’, three have a seating capacity of 4 people with dimensions of 5.2’ x 4.333’, and one has a seating capacity of 7 people with dimensions of 7.8’ x 9.1’. Recall that, when a booth is not selected to be used, then social

distancing constraints are not applicable from that booth to the other tables and booths. Besides those immovable booths, nine movable tables are considered in this case. Four of them have a seating capacity of 2 people with dimensions of 2.6’ x 4.333’, four have a seating capacity of 4 people with dimensions of 4.333’ x 4.333’, and one has a seating capacity of 6 people with dimensions of 9.1’ x 5.633’. Figure 4.8 displays two layout designs generated from the solutions for case 4.



(a) reduced capacity (α = 0.5) Z = 26

(b) original capacity (α = 1) Z = 37

Figure 4.8: Layout designs for case 4

Optimal solutions were found in 0.57 seconds and 512 seconds for α = 0.5 and α = 1, respectively. Note that the booth at the upper-right corner of the restaurant is not used in Figure 4.8 (a), whereas it is in Figure 4.8 (b). This is also the case for the booth located at the bottom-left corner.

Recall that the proposed MILP is concerned with maximizing the seating capacity of the dining area. However, the maximum capacity does not necessarily lead to the maximum revenue. In the next section, therefore, alternative optimization models are proposed to take into account the expected revenue under stochastic customer demands.

4.5 Extension to table mix problem (TMP)

Referring to the previous section, observe how in cases 3 and 4 that some of the largest sized tables are placed inside of the restaurant. For example, notice in Figure 4.8 (a) that a table which can accommodate six customers is placed in the middle of the restaurant. Indeed, placing this table in the center allows for the restaurant to achieve the maximum number of seats when enforcing social distancing and reduced capacity constraints. However, it is possible that this table will not be well-utilized since the likelihood of parties with six people was not taken into account when optimizing the layout. The determination of the best configuration of seating areas in this problem setting is known as the *table mix problem* (TMP).

As mentioned earlier in Section 4.2, methods have been proposed to find a reasonable configuration of seating areas inside of a restaurant for maximizing the total revenue. One of these pioneering efforts is from Thompson (2002) where a Naïve Ideal Table Mix model is proposed for incorporating the probability of different sized parties to arrive at the restaurant. The model outputs a preferred number of tables of different sizes, but the arrangement of tables inside of the restaurant is not part of the output. Guerriro et al. (2014) extend the work from Kimes and Thompson (2005) by taking the arrangement of tables into consideration using a two-phase approach. In the first phase, a solution to TMP is found by discretizing the restaurant into a series of segments (with fixed length and width) to determine the optimal number of tables to place in each segment in consideration of average demand and sizing constraints. During the second phase, the solution from the first phase is referenced for solving the *party mixes problem* (PMP), which considers the assignment of parties to tables to maximize revenue.

To our knowledge, the work from Guerriro et al. is the only approach in the literature where the table assortment and (partial) arrangement are accounted for simultaneously in TMP. However, it is unclear how to segment the restaurant most effectively, as well as how to determine their respective dimensions. Because of this, having too few or many segments (or too narrow or wide) are likely to result in solutions that are sub-optimal. Also, in their TMP model, it is assumed that parties can be seated at any table with a capacity that is greater than or equal to the party size. Doing so places the restaurant at risk of under-utilizing the seating areas during operation. Intuitively speaking, it makes more sense to assign parties to seating areas that are most similar in size for improving the utilization rate. Lastly, TMP and PMP are optimized sequentially without a feedback loop from PMP to TMP. Hence, a globally optimal solution is not guaranteed.

In an effort to resolve these shortcomings, the model from Section 4.3 can be transformed into a two-stage stochastic program to improve the table mix inside of the restaurant in consideration of a large number of customer demand scenarios (with respect to the party sizes) for maximizing the expected revenue of the restaurant. The proposed model also addresses the table assortment and arrangement simultaneously, but relaxes the condition that seating areas must be placed in user-defined segments (discrete space). Hence, tables can be placed in any region where dining is permitted (continuous space). An alternative representation of the customer demand compared to Guerriro et al. is used for expressing the arrival rate of customers per hour (denoted by λ), and it is of interest to find the table mix where the greatest number of parties can be accommodated. It is assumed that all parties arrive at the beginning of each hour to consider the extreme case

in this setting where all customers arrive at the restaurant simultaneously. It is also assumed that tables cannot be combined when assigning parties to tables. Karmarkar and Dutta (2011) share these assumptions in their approach for TMP during each time-period over a planning horizon.

Representing the problem in this fashion allows for TMP and PMP to be optimized simultaneously in a unified modelling framework, which has not been considered in the literature before. Furthermore, social distancing and reduced capacity constraints are incorporated into the layout design problem, which are newly emerging factors that restaurant owners need to take into consideration during pandemic events. Note that this model can also be applied by practitioners who are either opening a new restaurant or are interested in modifying an existing one, regardless of whether a pandemic is happening or not. Subsection 4.5.1 includes an overview on stochastic programming, and is followed by the two-stage stochastic program for TMP and PMP in Section 4.5.2.

4.5.1 Stochastic programming

In a stochastic programming framework, uncertainty in model parameters is represented as a set of scenarios to account for various outcomes that might occur within the system. The uncertainty is modeled using discrete or continuous probability distributions, which are assumed to be known a priori (Sahinidis, 2004). Stochastic programming models are typically represented as two- or multi-stage decision making problems, where decisions from the previous stages influence the decisions that are made in future stages. For a two-stage stochastic programming problem, a top-down decision-making framework is applied, where stage-one decisions are made at the current moment,

and the stage-two decisions are made in response to the observed uncertainty in the second stage based on the decisions that were made in the first stage (commonly referred to as programming with recourse).

Discretizing the uncertainty into a set of scenarios and stages allows for a deterministic variant of the original problem to be formulated where the expected cost of all scenarios is minimized in conjunction with other costs (Louveaux and Birge, 1997). However, the tractability of stochastic programming models becomes less attractive as the number of stages and scenarios increase since the size of the model grows exponentially after converting it into a deterministic form. Hence, the Benders Decomposition arose as an effective solution strategy.

4.5.1.1 Modeling and solution approaches

Benders Decomposition is one of the pioneering efforts for solving stochastic programming problems, and is still being used and improved upon to this day. Benders Decomposition is an effective algorithm that can solve two-stage stochastic programming problems to optimality. It is frequently referred to as the L-shaped method in the literature since the constraint matrix exhibits an L-shaped structure when executing the algorithm. Typically, the master problem is defined using only the stage-one variables. Sub-problems are represented as LP models, where each LP is associated with a different scenario and solved to optimality while fixing the values of the stage-one decision variables. Valid inequalities are added iteratively to the master problem using the solutions from the sub-problems, and the process is repeated until the lower and upper

bounds converge. The sub-problems represent an approximation of the expected cost associated with the uncertain parameters, so adding the valid inequalities into the master problem allow for the expected cost to be considered during the optimization (Benders, 1962). Since the sub-problems are represented as LPs, it is required for the stage-two model to consist of only continuous variables when applying Benders Decomposition. This is because the dual representation of the problem with strong duality is essential when applying the procedure, which is straightforward to verify when the model is formulated as an LP.

4.5.2 Two-stage stochastic programming model

In this section, a two-stage stochastic programming model is presented for finding an optimal solution to TMP and PMP simultaneously. During the first stage, the assortment and arrangement of seating areas is optimized using constraints from the formulation in Section 4.3. After obtaining a solution for the first stage problem, the second stage recourse problem then considers the assignment of parties to tables. The objective function of the stochastic program is to maximize the expected revenue that is generated across S equally likely scenarios with varying party sizes.

When generating the scenario data, lower and upper limits on the party sizes and λ are specified. Furthermore, the probability distribution of party sizes is assumed to be known a priori (e.g., Thompson, 2002 and Guerriro et al.). Then, scenarios are generated from the multinomial distribution with the number of trials, λ, and party size probabilities. For example, when $\lambda = 6$ and the party size is ranged between 1 and 8, a sample scenario is an array of six values [2, 4, 1, 4, 7, 3]. Since λ denotes the expected number of parties that arrive to the restaurant per hour, the order of the values can be

interpreted as the order in which the parties arrive to the restaurant (i.e., party of 2 arrived first, party of 4 arrived second, etc.). Note the objective function from Section 4.3 (maximize the number of seats) has a much different interpretation than the objective function of this stochastic program (i.e., maximize the expected revenue), suggesting that these two modelling approaches can result in different layout designs.

Recall that t_i and b_j represent whether or not table i/booth j are used inside of the restaurant ($i \in T, j \in B$). t_i and b_j are binary decision variables in the first stage problem. Also, recall that λ denotes the arrival rate of parties that will arrive to the restaurant during a one-hour period. As stated before, it is assumed that all parties arrive simultaneously at the beginning of an hour for representing the most extreme case. Let $P = \{1,2, \dots, \lambda\}$ be an index set of the parties for the hour. Depending on the values assigned to t_i and b_j and the arrival rate λ, three cases can occur: (1) the number of seating areas is less than the number of parties ($\sum_{i \in T} t_i + \sum_{i \in B} b_i < |P|$), (2) the number of seating areas is greater than the number of parties ($\sum_{i \in T} t_i + \sum_{i \in B} b_i > |P|$), and (3) the number of seating areas is equal to the number of parties ($\sum_{i \in T} t_i + \sum_{i \in B} b_i = |P|$). Note that cases (1) and (2) result in what is known as an unbalanced assignment problem, where a one-to-one mapping of seating areas to parties is impossible since there are too few seating areas and parties in cases (1) and (2), respectively. Although the number of seating areas and parties are equal in case (3), it is still possible that not all parties can be seated due to the mismatch between the party sizes and table capacities, thus resulting in another instance of an unbalanced assignment problem. As a result, it is necessary to formulate the second stage model as an unbalanced assignment problem.

The proposed two-stage stochastic program is referred to as the static variant because the party size information is assumed to be known for the hour. The maximization of expected revenue in this static setting will have a preference to greedily assign parties to tables that have a higher dollar amount associated to them regardless of the order they are in for each scenario. This can be adjusted by introducing what are known as seniority constraints (Caron et al., 1999) for assigning parties to tables, resulting in what is referred to as the dynamic variant of the proposed two-stage stochastic program, where it is of interest to maximize the number of parties that can be accommodated based on the order they arrive to the restaurant. The following notation of problem parameters and decision variables are used to present the static variant:

$p_y^s =$ number of people in party $y \in P$ for scenario $s \in \{1, \dots, S\}$
$R_y =$ revenue that is generated from party $y \in P$
z_{i,p_y}: binary parameter equal to 1 if seating area $i \in T \cup B$ can accommodate parties of size p_y $(y \in P)$
$x_{i,y}^s$: binary variable equal to 1 if seating area $i \in T \cup B$ is assigned to party $y \in P$ in scenario $s \in \{1, \dots, S\}$.

Using the notation shown above, the deterministic equivalent of the two-stage stochastic programming model is as follows:

$$\text{Maximize } \frac{1}{S}\sum_{s=1}^{S}\left[\sum_{y\in P} R_y\, p_y^s \sum_{i\in T\cup B} z_{i,p_y^s} x_{i,y}^s\right] \tag{4.33}$$

subject to

(4.2)-(4.28)

$$\sum_{y\in P} x_{i,y}^s \le t_i * \max\left(z_{i,p_1^s}, \dots, z_{i,p_\lambda^s}\right) \qquad \forall i \in T, s \in \{1, \dots, S\} \tag{4.34}$$

$$\sum_{y\in P} x_{i,y}^s \le b_i * \max\left(z_{i,p_1^s}, \dots, z_{i,p_\lambda^s}\right) \qquad \forall i \in B, s \in \{1, \dots, S\} \tag{4.35}$$

$$\sum_{i \in T \cup B} x_{i,y}^{s} \leq 1 \qquad \forall y \in P, s \in \{1, \ldots, S\} \tag{4.36}$$

$$x_{i,y}^{s} \in \{0,1\} \qquad \forall i \in T \cup B, y \in P, s \in \{1, \ldots, S\}. \tag{4.37}$$

Objective function (4.33) maximizes the expected revenue across S equally likely scenarios. (4.2)–(4.28) are included for satisfying the reduced capacity, social distancing, nonoverlapping seating areas, and seating area placement constraints. It is assumed that the revenue from each party is linearly proportional to the party size. (4.34)-(4.35) restrict the tables and booths, respectively, from being assigned to no more than one party, whereas (4.36) restricts each party from being assigned to no more than one seating area. (4.37) define the binary variables for the second stage problem.

Binary parameter z_{i,p_y} is referenced in (4.33)-(4.35) for determining if seating area i is large enough to accommodate party y. If $t_i = \max\left(z_{i,p_1^s}, \ldots, z_{i,p_\lambda^s}\right) = 1$, this implies that table i can be considered for assignment since the table is being used within the restaurant (decision that is made during the first stage) and it can accommodate at least one party in scenario s. On the other hand, if $t_i = 0$, then table i cannot be assigned to any party since the table is not placed in the restaurant. When $\max\left(z_{i,p_1^s}, \ldots, z_{i,p_\lambda^s}\right) = 0$, then table i is also restricted from being assigned to a party regardless of the value assigned to t_i since it cannot accommodate any of the parties in scenario s. The same reasoning applies for the booth seating areas.

As mentioned earlier, one of the shortcomings of stochastic programming models is that they become more and more computationally complex as the number of scenarios S increases. For most cases, it is desired to generate a large number of scenarios to capture the underlying probability distribution. With this in mind, it is of interest to apply

Benders Decomposition to improve the tractability rather than directly solving the deterministic variant of the proposed model. Note that the only decision variable in the recourse problem is $x_{i,y}^{s}$, and is restricted to binary values in its initial representation. However, these binary variables can be relaxed to continuous ones in the formulation because the coefficient matrix A of the second stage model is totally unimodular (TU) (Hoffman and Gale in Hoffman and Kruskal, 1956), and the right-hand side vector b is restricted to integer values. In particular, after adding slack variables to the inequality constraints of the recourse problem, the coefficient matrix is equivalent to the node-arc incidence matrix of a bipartite graph, where one set of nodes corresponds to $T \cup B$ and the other set of nodes corresponds to P. Hence, the coefficient matrix is totally unimodular (Hoffman and Gale in Hoffman and Kruskal, 1956).

Maximize (4.33)

subject to

(4.2)-(4.28), (4.34)-(4.36)

$$0 \leq x_{i,y}^{s} \leq 1 \qquad \forall i \in T \cup B, y \in P, s \in \{1, \dots, S\}. \qquad (4.38)$$

Now that the second stage recourse problem is a linear program, Benders Decomposition can be applied for finding an optimal solution to the proposed two-stage stochastic programming problem.

4.5.2.1 Dynamic variant

Note that the static variant of the two-stage stochastic program is primarily focused on maximizing the expected revenue across all scenarios for a given problem instance. Because of this, larger-sized parties are given a higher preference for being seated in the recourse problem regardless of the order in which each party arrives. This

can result in an unsatisfactory customer experience for the earlier arriving parties who blatantly get skipped for the sake of the restaurant to earn more money. To address this issue, a dynamic variant of the stochastic program is proposed where it is of interest to maximize the number of parties that are seated in the restaurant based on the order in which they arrive. The dynamic variant can still be interpreted as a revenue maximization problem from this perspective because the restaurant will earn more revenue for each additional party (in consecutive order) that gets seated. At the same time, a restaurant owner would prefer to assign parties to the smallest sized seating area that can accommodate the party to maximize the utilization of the table. The work by Caron et al. (1999) where so-called seniority and priority constraints for assignment problems are introduced is referenced to incorporate these additional features into the optimization process.

For the assignment problem (in general), it is of interest to assign m persons to n jobs for optimizing a particular objective function. As observed before, there are instances of the assignment problem where $m \neq n$, resulting in an unbalanced assignment problem. The balanced assignment problem is given when $m = n$. Depending on the circumstances, it might be necessary to enforce additional requirements for the assignment problem (such as budget, time, etc.), and are typically referred to as side-constraints. Caron et al. (1999) propose seniority constraints to produce an optimal solution that considers the ranking of persons in the problem instance (perhaps based on skill level or experience) by granting them a higher preference for being assigned to jobs they are capable of doing. This is accomplished by modifying the objective function

weights based on the lexicographic order of all individuals (from largest to smallest). The same ordering can also apply for the jobs and is referred to as priority constraints.

To do so, the authors represent the problem as a bipartite graph $G_{P \cup Q} = (P \cup Q, E)$, where P and Q are the index sets of all people and jobs, respectively, and E is the set of edges that associate which jobs each person can perform. This mapping references binary parameter $a_{i,j}$, which is equal to 1 if person i can perform job j; otherwise, 0. Let $A_1, A_2, \ldots, A_M$ be a partition of $\{1,2,\ldots,m\}$ representing the seniority classes of each person, and assume that A_k has a higher seniority than A_l, if $k < l$, for $k, l = 1,2,\ldots, M$. Similarly, let $B_1, B_2, \ldots, B_N$ have a similar interpretation, with the only exception being that it is for the jobs and instead is a partition of $\{1,2,\ldots,n\}$. Given objective function coefficient c_{ij} ($\forall i = \{1,2, \ldots, m\}, j \in \{1,2, \ldots n\}$), the modified objective coefficient $\hat{c}_{ij}$ can be computed that incorporates seniority and priority levels of each person and job, respectively, by using the following equations.

$$L = 1 + 2 \min\{m, n\} \max_{(p_i, q_j) \in E} |c_{i,j}| \tag{4.39}$$

$$\hat{c}_{i,j} = \begin{cases} L(M + N - k(i) - l(j) + 1) + c_{ij}, & if\ a_{i,j} = 1, \\ 0 & \text{otherwise.} \end{cases} \tag{4.40}$$

where $k(i)$ is the class index of person i and $l(j)$ is the class index of job j. In the context of assigning parties to tables, sets $A_1, A_2, \ldots, A_M$ can be constructed in consideration of the order in which the parties arrive. For example, suppose that there are three 4-seat tables, four 6-seat tables, and two 8-seat tables (nine tables in total). Also, suppose that there are 10 parties with sizes [3, 7, 6, 3, 5, 4, 2, 5, 6, 1]. The first party in the array is of size 3, and is mapped to A_1 because it arrives before all other parties. Similarly, the second party is of size 7, and is mapped to A_2 since the party arrives after the first one.

This process can be repeated for all other parties, yielding $A_1, A_2, \dots, A_{10}$ and $M = 10$. It is assumed that there is not a so-called priority ordering for the tables. Because of this, the table indices can be stored inside of index set B_1, thus implying that $N = 1$.

Recall that the static and dynamic problem settings have slightly different interpretations from one another. The former variant is focused on maximizing the expected revenue regardless of each parties' position in the queue, whereas the latter attempts to accommodate the largest number of parties with respect to the order in which they arrive to the restaurant. With that being said, it is undesired to greedily select the largest sized parties in the dynamic setting, but instead configure the arrangement of seating areas to maximize the number of parties that can be seated with a higher priority given to parties that arrive earliest to the restaurant. In addition, it might be preferred to assign parties to seating areas that most closely match in size. For example, assigning a party of four to a seating area with four seats makes more sense than assigning the same party to a seating area with six seats if both seating areas are available. (4.39)-(4.40) can be modified to incorporate these features by scaling objective coefficient $c_{i,y}^s$ ($\forall i \in T \cup B, y \in P, s \in \{1, \dots, S\}$) based on the position of each party in line, as well as the difference between the number of seats at a seating area and the size of the party.

All revenue information is excluded entirely in the dynamic variant to avoid potentially giving larger sized parties higher weights in the objective function. Let q_i denote the number of seats at seating area $i \in T \cup B$.

$$L = 1 + 2\min\{|P|, |T \cup B|\} \tag{4.41}$$

$$\hat{c}_{i,y}^s = \begin{cases} (L(|P| + 1 - y) - L\left(1 - \dfrac{1}{\left(q_i - p_y^s\right)^2 + 1}\right), & if\ z_{i,p_y^s} = 1, \\ 0 & \text{otherwise.} \end{cases} \tag{4.42}$$

Note from (4.42) that $\hat{c}^s_{i,y'} > \hat{c}^s_{i,y''}$ if $y' < y''$ in order to give a higher priority to party y' since they arrive first. Furthermore, $\hat{c}^s_{i,y} > \hat{c}^s_{j,y}$ if $q_i < q_j$ when $z_{i,p^s_y} = z_{j,p^s_y} = 1$. That is, if party y can be seated in both seating areas i and j while the number of seats at i is smaller than that of j, it is preferred to seat the party at seating area i. These values can be computed a priori when given scenario data, thus allowing for the second stage problem to still be represented as an LP. Also, it is guaranteed that the range of the objective coefficients for each distinct party will be non-overlapping from all other parties, which is intended to enforce the priority of parties that arrive earliest. From (4.41)-(4.42), objective function (4.34) can be rewritten to formulate the dynamic variant of the two-stage stochastic program, which can be solved to optimality by applying Benders Decomposition.

$$\text{Maximize } \frac{1}{S}\sum_{s=1}^{S}\left[\sum_{i\in T\cup B}\sum_{y\in P}\hat{c}^s_{i,y}x^s_{i,y}\right] \tag{4.43}$$

subject to

(4.2)-(4.28), (4.34)-(4.36), (4.38).

4.6 Numerical results

The numerical results obtained by implementing the models from Section 4.5.2 and Section 4.5.2.1 are reported in this section (referred to as the static approach (*S'*) and dynamic with preference approach (*D-P'*), respectively). Another instance of the dynamic variant was tested where the component in (4.42) that assigns a larger coefficient weight to tables with fewer seats is excluded to assess the impact it has on an optimal solution (referred to as the dynamic with no preference approach (*D-NP'*)). All runs were executed using the same server environment as reported in Section 4.4. The restaurant

schematic of case 3 from Section 4.4 (Figure 4.5) is referenced for this experiment. Like Section 4.4, two values of $\alpha \in \{0.5, 1\}$ are considered to illustrate how the layout design is impacted with and without the reduced capacity constraint. In addition, d_{min} is set equal to 6 feet.

In this study (and without loss of generality), it is assumed that the generated revenue from an accommodated party is linearly proportional to the party size. Thus, it was established that each party member will spend $10 at the restaurant, on average, suggesting that a party of 1 will generate $10 in revenue, party of 2 will generate $20, etc. Varying values for λ are applied to represent different hourly arrival rates of customers (5, 10, 15, and 20). Table 4.1 contains the party size probabilities that were used for generating demand scenarios to illustrate how the layout design changes across five different cases (from Thompson, 2002 and Guerriro et al., 2014). For each case, 500 randomly generated scenarios are drawn from a multinomial distribution to create λ parties consisting of one to eight customers. Distinct sets of potential seating areas are constructed for each case based on the probability distributions of the party size, (as shown in Table 4.2), where each pair of values connected by a colon represents the capacity of a seating area and the number of seating areas with that capacity, respectively. Although a seating area can be either a table or a booth, it will be referred to as a table in what follows for simplicity. The dimensions of the four, six, and eight seat tables are 4.333' x 4.333', 5.333' x 5', and 5.5' x 7', respectively.

For the sake of comparison, the MILP from Section 4.3 is also considered in this experimental design to assess its performance when customer demand is excluded entirely from the formulation (referred to as the naïve approach (N')). A 1-hour time limit

was imposed for the situation where an optimal solution could not be found for *N'* in a reasonable amount of time. Tables 4.3 and 4.4 display the table assortments that are output from each approach with and without reduced capacity constraints enforced, respectively. Similar to Table 4.2, a pair of numbers connected by a colon represents the capacity of a table and the number of selected tables with that capacity in the solution, respectively. Solution times for all approaches where reduced capacity is enforced and not enforced are reported in Tables 4.5 and 4.6, respectively.

Table 4.1: Party size probabilities for each case

	Case				
Party size	1	2	3	4	5
1	0.16	0.05	0.02	0	0
2	0.51	0.17	0.12	0.5	0.25
3	0.15	0.34	0.16	0	0
4	0.1	0.27	0.20	0.3	0.25
5	0.04	0.08	0.23	0	0
6	0.02	0.04	0.13	0.1	0.25
7	0.01	0.03	0.09	0	0
8	0.01	0.02	0.05	0.1	0.25

Table 4.2: Distribution of potential tables for each case

Case	Potential tables considered for placement (table capacity: number tables)
1	{4: 15, 6: 6, 8: 1}
2	{4: 10, 6: 8, 8: 2}
3	{4: 6, 6: 8, 8: 4}
4	{4: 11, 6: 6, 8: 3}
5	{4: 7, 6: 6, 8: 5}

Table 4.3: Table assortments with reduced capacity constraints enforced

		Case				
	λ	1	2	3	4	5
S'	5	4: 2 / 6: 6 / 8: 1	4: 0 / 6: 6 / 8: 2	4: 0 / 6: 3 / 8: 4	4: 1 / 6: 4 / 8: 3	4: 0 / 6: 2 / 8: 5
	10	4: 6 / 6: 3 / 8: 1	4: 6 / 6: 2 / 8: 2	4: 3 / 6: 4 / 8: 2	4: 6 / 6: 2 / 8: 2	4: 2 / 6: 2 / 8: 4
	15	4: 11 / 6: 0 / 8: 1	4: 8 / 6: 2 / 8: 1	4: 3 / 6: 4 / 8: 2	4: 6 / 6: 2 / 8: 2	4: 2 / 6: 2 / 8: 4
	20	4: 8 / 6: 2 / 8: 1	4: 8 / 6: 2 / 8: 1	4: 3 / 6: 4 / 8: 2	4: 6 / 6: 2 / 8: 2	4: 2 / 6: 2 / 8: 4
D-P'	5	4: 6 / 6: 3 / 8: 1	4: 5 / 6: 2 / 8: 2	4: 4 / 6: 3 / 8: 2	4: 5 / 6: 2 / 8: 2	4: 4 / 6: 2 / 8: 3
	10	4: 8 / 6: 2 / 8: 1	4: 6 / 6: 2 / 8: 2	4: 3 / 6: 4 / 8: 2	4: 6 / 6: 2 / 8: 2	4: 4 / 6: 2 / 8: 3
	15	4: 11 / 6: 0 / 8: 1	4: 8 / 6: 2 / 8: 1	4: 5 / 6: 4 / 8: 1	4: 8 / 6: 2 / 8: 1	4: 4 / 6: 2 / 8: 3
	20	4: 13 / 6: 0 / 8: 0	4: 10 / 6: 2 / 8: 0	4: 5 / 6: 4 / 8: 1	4: 11 / 6: 0 / 8: 1	4: 6 / 6: 2 / 8: 2
D-NP'	5	4: 0 / 6: 5 / 8: 1	4: 1 / 6: 3 / 8: 2	4: 0 / 6: 1 / 8: 4	4: 1 / 6: 2 / 8: 3	4: 1 / 6: 1 / 8: 5
	10	4: 6 / 6: 3 / 8: 1	4: 6 / 6: 2 / 8: 2	4: 3 / 6: 4 / 8: 2	4: 6 / 6: 2 / 8: 2	4: 4 / 6: 2 / 8: 3
	15	4: 11 / 6: 0 / 8: 1	4: 8 / 6: 2 / 8: 1	4: 5 / 6: 4 / 8: 1	4: 8 / 6: 2 / 8: 1	4: 6 / 6: 2 / 8: 2
	20	4: 13 / 6: 0 / 8: 0	4: 10 / 6: 2 / 8: 0	4: 5 / 6: 4 / 8: 1	4: 11 / 6: 0 / 8: 1	4: 6 / 6: 2 / 8: 2
N'	-	4: 4 / 6: 6 / 8: 0	4: 2 / 6: 6 / 8: 1	4: 1 / 6: 8 / 8: 0	4: 1 / 6: 4 / 8: 3	4: 2 / 6: 2 / 8: 4

Table 4.4: Table assortments without reduced capacity constraints enforced

		Case				
	λ	1	2	3	4	5
S'	5	4: 10 / 6: 5 / 8: 1	4: 6 / 6: 8 / 8: 2	4: 6 / 6: 6 / 8: 4	4: 8 / 6: 5 / 8: 3	4: 6 / 6: 5 / 8: 5
	10	4: 10 / 6: 5 / 8: 1	4: 6 / 6: 8 / 8: 2	4: 6 / 6: 6 / 8: 4	4: 9 / 6: 4 / 8: 3	4: 6 / 6: 5 / 8: 5
	15	4: 10 / 6: 5 / 8: 1	4: 9 / 6: 5 / 8: 2	4: 6 / 6: 6 / 8: 4	4: 9 / 6: 4 / 8: 3	4: 6 / 6: 5 / 8: 5
	20	4: 10 / 6: 5 / 8: 1	4: 9 / 6: 5 / 8: 2	4: 6 / 6: 6 / 8: 4	4: 9 / 6: 4 / 8: 3	4: 6 / 6: 5 / 8: 5
D-P'	5	4: 5 / 6: 2 / 8: 1	4: 5 / 6: 2 / 8: 1	4: 6 / 6: 3 / 8: 1	4: 4 / 6: 5 / 8: 3	4: 5 / 6: 2 / 8: 5
	10	4: 10 / 6: 3 / 8: 1	4: 8 / 6: 6 / 8: 1	4: 6 / 6: 8 / 8: 2	4: 8 / 6: 6 / 8: 1	4: 7 / 6: 4 / 8: 5
	15	4: 12 / 6: 3 / 8: 1	4: 9 / 6: 7 / 8: 0	4: 6 / 6: 8 / 8: 2	4: 11 / 6: 4 / 8: 1	4: 7 / 6: 5 / 8: 4
	20	4: 12 / 6: 3 / 8: 1	4: 9 / 6: 7 / 8: 0	4: 6 / 6: 7 / 8: 3	4: 11 / 6: 2 / 8: 3	4: 6 / 6: 5 / 8: 5
D-NP'	5	4: 2 / 6: 2 / 8: 1	4: 0 / 6: 5 / 8: 1	4: 1 / 6: 3 / 8: 1	4: 1 / 6: 3 / 8: 1	4: 0 / 6: 3 / 8: 4
	10	4: 4 / 6: 5 / 8: 1	4: 2 / 6: 6 / 8: 2	4: 0 / 6: 7 / 8: 4	4: 2 / 6: 5 / 8: 3	4: 1 / 6: 4 / 8: 5
	15	4: 9 / 6: 5 / 8: 1	4: 9 / 6: 4 / 8: 2	4: 5 / 6: 6 / 8: 4	4: 8 / 6: 4 / 8: 3	4: 5 / 6: 5 / 8: 5
	20	4: 10 / 6: 5 / 8: 1	4: 9 / 6: 5 / 8: 2	4: 6 / 6: 6 / 8: 4	4: 9 / 6: 4 / 8: 3	4: 6 / 6: 5 / 8: 5
N'	-	4: 10 / 6: 5 / 8: 1	4: 2 / 6: 10 / 8: 3	4: 2 / 6: 7 / 8: 6	4: 7 / 6: 6 / 8: 3	4: 2 / 6: 7 / 8: 6

Table 4.5: Solution times (seconds) with reduced capacity constraints enforced

		Case				
	λ	1	2	3	4	5
S'	5	18.56	15.06	16.2	13.79	11.74
	10	30.03	26.83	34.33	25.13	34.07
	15	44.21	39.34	41.99	39.69	41.17
	20	44.99	42.77	45.26	45.69	37.95
D-P'	5	42.76	41.33	42.22	47.9	40.3
	10	77.8	81.52	66.8	64.86	54.65
	15	107.99	72.99	59.77	86.66	97.12
	20	83.01	50.37	76.66	88.85	87.44
D-NP'	5	22.02	14.68	13.17	14.95	18.27
	10	28.23	26.97	41.45	34.29	34.7
	15	39.85	33.42	30.96	44.42	57.83
	20	35.44	19.79	33.93	38.36	51.36
N'	-	1.74	1.37	1.11	1.32	1.14

Table 4.6: Solution times (seconds) without reduced capacity constraints enforced

		Case				
	λ	1	2	3	4	5
S'	5	19.28	40.76	42.29	36.87	1,919.86
	10	18.22	43.85	58.84	38.37	22.93
	15	22.37	19.1	73.05	44.68	44.28
	20	25.12	23.35	95.86	26.89	61.13
D-P'	5	55.39	42.78	76.55	30.25	98.07
	10	91.87	93.72	343.82	94.01	201.38
	15	166.52	107.46	155.1	132.32	282.49
	20	192.78	113.5	537.13	149.86	270.1
D-NP'	5	25.36	39.23	28.63	25.2	37.28
	10	27.11	47.48	39.87	43.41	42.52
	15	51.13	64.39	86.92	53.04	109.39
	20	60.74	56.15	122.12	77.06	84.98
N'	-	26.22	3,600*	3,600*	70.89	3,600*

* Gurobi optimizer timed-out after 3600 seconds.

Observe how the table placement varies with respect to the party size probabilities that are used for sampling in Tables 4.3 and 4.4. The arrival rates also influence the output from the static and dynamic approaches where a smaller arrival rate tends to place fewer tables in the restaurant versus larger arrival rates. For example, when *D-NP'* is applied, only six tables are placed for $\lambda = 5$ in case 1, while 13 tables are placed for $\lambda = 20$ (table 4.3). This behavior is still apparent for a few instances even when the margin between arrival rates is small (i.e., *D-P'* with arrival rates 15 and 20 for case 2 in Table 4.3; *D-P'* with arrival rates 15 and 20 for case 3 in Table 4.4).

More diverse solutions are produced in Table 4.4 versus Table 4.3 since the number of seats from potential tables (referring to Table 4.2) is nearly twice as large as the maximum occupancy of the restaurant (105 individuals) when reduced capacity ($\alpha = 0.5$) is enforced. Note that the naïve approach *N'* produces the same assortment of tables regardless of the value assigned to λ. It is also noteworthy that *N'* yielded the same optimal solution for some of the runs (i.e., *S'* with arrival rates 10, 15, 20 for case 5 in Table 4.3; *S'* with arrival rates 5, 10, 15, 20 for case 1 in Table 4.4). Similar to the behavior exhibited in the numerical results of Section 4.4, *N'* has a preference to place as many of the largest-sized tables into the restaurant as possible before the smaller ones. This is particularly evident when enforcing reduced capacity where a majority of the placed tables have six or more seats for all cases. There is one exception in Table 4.4 where 10 of the placed tables have four seats, which is greater than number of six and eight seat tables in the layout combined.

When reduced capacity constraints are enforced, the best computational performance was achieved using *N',* while *D-P'* consumed the most computational time

for finding an optimal solution (Table 4.5). This result is consistent with the empirical study in Section 4.4, where *N'* was quickly solved to optimality for all of the restaurant schematics with reduced capacity constraints enforced. The computational performance when enforcing the reduced capacity constraints is similar where *D-P'* required more computational time than *S'* and *D-NP'* (Table 4.6). Case 5 shows that *S'* required nearly 2,000 seconds to perform the Benders Decomposition algorithm, which is unusual since all other runs were solved to optimality in less than 96 seconds. Note that the Gurobi solver timed-out for several runs when optimizing the table configuration for *N'* in Table 4.6.

To evaluate the quality of solutions that are produced from each approach, a simulation model was written in Python (3.7.3, 2018) to mimic restaurant operations over a 5-hour period. Party arrivals are modelled as a Poisson process, where the time between two consecutive arrivals follows an exponential distribution with a mean of $1/\lambda$. The simulation prioritizes assigning parties to the smallest available table in an effort to improve table utilization and increase the likelihood that larger-sized tables are available for larger-sized parties upon arrival. As before, it is assumed that tables cannot be combined together to accommodate a large-sized party. A triangular distribution is used to simulate meal durations to each party based on its size using the average values populated in Table 4.7 (from Guerriro et al., 2014). Let m_{p_y} denote the average meal duration for a party of size p_y. Then, the parameter values (minimum, mode, maximum) of the triangular distribution are set as $(m_{p_y} - 5, m_{p_y}, m_{p_y} + 10)$, respectively.

Table 4.7: Average meal duration based on the party size

Party size	1	2	3	4	5	6	7	8
Minutes	26.25	30	33.75	37.5	41.25	45	48.75	52.5

Two ways to handle the case where no tables are readily available for a party are tested. First, parties are rejected from the restaurant if there are no available tables (queue disabled), and the second allows for parties to wait in line for another available table (queue enabled). 200 replications were run to collect relevant performance statistics including the average revenue, table utilization, party rejection rates (when the queue is disabled), and average waiting times (when the queue is enabled). The average revenue and utilization rate provide insight into how well parties are being matched to tables, whereas the rejection rate and average waiting time metrics serve as a proxy for measuring the customer satisfaction level, where smaller values for both are ideal.

4.6.1 Simulation output with queue disabled

The simulation model with the queue option disabled was applied using the table assortments shown in Tables 4.3 and 4.4. Parties are accepted only if there is an available table that can accommodate them upon their arrival. Utilization rates are computed by taking the ratio between the number of customers currently seated at a given table and the table capacity, and gets updated whenever a new party is assigned to it. The utilization rates of all tables in a single replicate are aggregated by taking the average of all of them, yielding the utilization rate of the restaurant. This process is repeated for all replicates, and the average of all utilization rates is the final value that is presented for reporting. Rejection rates are calculated by taking the ratio between the number of parties that were

rejected and the total number of parties that visited the restaurant. The rejection rates across all replications are aggregated in a similar fashion as the utilization rates.

Table 4.8 displays the simulation output with reduced capacity constraints enforced across all five cases of party size probabilities. The largest revenue values are bolded in Table 4.8 for convenience.

Table 4.8: Output with reduced capacity constraints enforced – queue disabled

		λ	Revenue	Utilization Rate	Rejection Rate
Case 1	*S'*	5	**511.6**	49.7%	0.2%
		10	965.6	55.0%	1.5%
		15	**1,299.7**	55.4%	4.9%
		20	**1,565.45**	52.9%	12.4%
	D-P'	5	509.6	62.6%	0.1%
		10	**977.45**	59.5%	0.9%
		15	**1,299.7**	55.4%	4.9%
		20	1,474.9	54.9%	10.1%
	D-NP'	5	498.3	39.9%	2.4%
		10	965.6	55.0%	1.5%
		15	**1,299.7**	55.4%	4.9%
		20	1,474.9	54.9%	10.1%
	N'	5	489.95	51.8%	2.2%
		10	919.95	48.6%	2.5%
		15	1,261.4	47.3%	8.2%
		20	1,464.95	47.1%	16.3%
Case 2	*S'*	5	718.15	58.0%	0.4%
		10	**1,327.45**	72.7%	4.0%
		15	**1,743.35**	74.0%	9.9%
		20	**1,987.45**	72.8%	18.2%
	D-P'	5	**721**	76.6%	0.3%
		10	**1,327.45**	72.7%	4.0%
		15	**1,743.35**	74.0%	9.9%
		20	**1,987.55**	74.7%	15.7%
	D-NP'	5	677.6	58.5%	4.3%
		10	**1,327.45**	72.7%	4.0%
		15	**1,743.35**	74.0%	9.9%
		20	**1,987.55**	74.7%	15.7%
	N'	5	707.3	64.9%	0.9%
		10	1,282.9	62.5%	6.1%

		15	1,615	60.8%	17.2%
		20	1,784.65	60.5%	27.7%
Case 3	*S'*	5	**887.6**	64.5%	2.5%
		10	**1,530.6**	77.9%	10.5%
		15	**1,848.45**	75.2%	23.0%
		20	2,012.1	74.2%	33.6%
	D-P'	5	882.65	82.5%	2.4%
		10	**1,530.6**	77.9%	10.5%
		15	1,832.7	79.4%	21.4%
		20	**2,038.65**	78.2%	30.2%
	D-NP'	5	805.15	59.7%	11.5%
		10	**1,530.6**	77.9%	10.5%
		15	1,832.7	79.4%	21.4%
		20	**2,038.65**	78.2%	30.2%
	N'	5	726.9	72.4%	13.2%
		10	1,354.7	71.2%	16.5%
		15	1,711.05	70.5%	24.8%
		20	1,924.7	70.1%	33.9%
Case 4	*S'*	5	**759.8**	64.5%	0.8%
		10	**1,341**	72.5%	4.3%
		15	**1,680.7**	68.5%	13.2%
		20	**1,866.9**	66.1%	23.1%
	D-P'	5	742.8	78.5%	0.8%
		10	**1,341**	72.5%	4.3%
		15	1,666.9	71.6%	11.6%
		20	1,822.6	69.8%	18.3%
	D-NP'	5	705.6	57.4%	4.6%
		10	**1,341**	72.5%	4.3%
		15	1,666.9	71.6%	11.6%
		20	1,822.6	69.8%	18.3%
	N'	5	**759.8**	64.5%	0.8%
		10	1,273.3	58.0%	9.9%
		15	1,515.7	55.7%	23.8%
		20	1,636.3	54.5%	34.1%
Case 5	*S'*	5	970.8	70.4%	3.2%
		10	1,579.1	76.8%	14.8%
		15	1,851.8	75.2%	30.0%
		20	1,964.3	72.4%	40.4%
	D-P'	5	969.6	87.0%	3.1%
		10	**1,586.5**	82.8%	13.7%
		15	**1,897.5**	80.0%	25.9%
		20	**2,004.6**	80.3%	33.7%
	D-NP'	5	971	74.2%	3.3%

		10	**1,586.5**	82.8%	13.7%
		15	1,815.3	81.3%	25.8%
		20	**2,004.6**	80.3%	33.7%
	N'	5	**992.2**	83.6%	1.9%
		10	1,579.1	76.8%	14.8%
		15	1,851.8	75.2%	30.0%
		20	1,964.3	72.4%	40.4%

For case 1, it can be noticed that *S'* resulted in a majority of the occurrences of the most revenue amongst all approaches (three out of four), and is followed by *D-P'*. Utilization rates are highest when λ = 5, 10, 20 when using *D-P'* (tied with *D-NP'* when λ = 20), and are equal to the utilization rates from *S'* and *D-NP'* when λ = 15. *N'* yielded the lowest utilization rates for most values of λ, with one exception being where *D-NP'* had the lowest value when λ = 5. Regardless, *D-NP'* had greater than or equal to utilization rates than *S'* for three out of four values assigned to λ (10, 15, and 20). The lowest rejection rates are associated with *D-P'*, whereas *N'* had the highest for most runs. It is worth mentioning that case 1 consists mainly of parties with one, two, and three guests, thus resulting in relatively low utilization rates since tables with four, six, and eight seats are strictly used.

Looking at case 2, *D-P'* yielded the most revenue when λ = 5, and was tied with all other variants of the stochastic program for the remaining hourly arrival rates. Utilization rates are initially the lowest from *S'* and *D-NP'* for λ = 5, but are gradually improved as the hourly arrival rate increases, resulting in *N'* to then exhibit the worst utilization. In addition, *D-P'* also had the lowest rejection rates. *S'* had a similar utilization rate as *D-NP'* when λ = 5 (58% versus 58.5%), but the rejection rate is much larger in the former approach (0.4% versus 4.3%). This difference is ultimately reflected in the revenue that is generated from both approaches, where *D-NP'* produced the least

revenue for $\lambda = 5$. This outcome is likely associated with the number of tables that are placed inside the restaurant from each approach, where *D-NP'* resulted in only six tables while the others placed at least eight.

Similar observations can be made for the remaining cases. In summary (with ties included), *S'* and *D-P'* both resulted in the most revenue for 12 occurrences, whereas *D-NP'* and *N'* did for nine and two occurrences, respectively. This is reasonable since *S'* is solely focused on optimizing the expected revenue in its stochastic programming representation, while others used metrics that approximate the revenue. *D-P'* also considers the revenue but from a different perspective where it is preferred to seat parties in the order they arrive, which is not the case for *S'* since parties can be selectively chosen by the owner to increase the revenue. The utilization/rejection rates were highest/lowest for *D-P'* and *D-NP'* (19 and 14 occurrences, respectively; 18 and 14 occurrences, respectively). This result also makes sense because *D-P'* has a preference to match parties to tables that are most similar in size. *D-NP'* does not enforce this preference, but perhaps applying the dynamic seating policy inherently results in parties to be assigned to tables more strategically. *S'* outperformed *N'* with respect to utilization and rejection rates in the sense of the number occurrences where the highest/lowest values were observed (five versus zero; six versus two).

The same instance of the simulation with the queue option disabled was executed using the table assortments in Table 4.4 to determine if the observations from the previous result are consistent when relaxing the reduced capacity constraints. Larger values are to be expected for the generated revenue since more tables can be placed

inside of the restaurant (particularly for larger values of λ). It is not clear, however, to what degree the other metrics will be impacted.

Table 4.9 displays the simulation output without reduced capacity constraints enforced.

Table 4.9: Output without reduced capacity constraints enforced – queue disabled

		λ	Revenue	Utilization Rate	Rejection Rate
Case 1	*S'*	5	511.85	63.8%	0.1%
		10	995	64.0%	0.3%
		15	1,418.35	58.8%	0.5%
		20	1,795.8	55.3%	1.3%
	D-P'	5	**514.45**	60.0%	0.4%
		10	**996.2**	63.5%	0.3%
		15	**1,425.55**	61.0%	0.6%
		20	**1,799.5**	58.0%	1.5%
	D-NP'	5	476.45	47.0%	6.4%
		10	967.1	50.5%	1.4%
		15	1424	57.5%	0.6%
		20	1,795.8	55.3%	1.3%
	N'	5	511.85	63.8%	0.1%
		10	995	64.0%	0.3%
		15	1,418.35	58.8%	0.5%
		20	1,795.8	55.3%	1.3%
Case 2	*S'*	5	**734.25**	79.0%	0.1%
		10	1,399.3	72.8%	0.3%
		15	**2,004.35**	75.3%	0.8%
		20	**2,464.25**	71.9%	3.3%
	D-P'	5	696.55	76.9%	1.4%
		10	1370	75.6%	1.5%
		15	1,814.6	73.1%	5.1%
		20	2,276.2	70.8%	6.2%
	D-NP'	5	673.55	56.9%	5.1%
		10	1,332.75	62.3%	3.1%
		15	1,963.5	75.1%	1.3%
		20	**2,464.25**	71.9%	3.3%
	N'	5	731.85	67.5%	0.0%
		10	**1,407.1**	65.0%	0.1%
		15	1,982.95	61.4%	0.8%
		20	2,440.8	59.3%	3.7%

Case 3	*S'*	5	926.45	84.9%	0.0%
		10	1,753.6	84.4%	0.3%
		15	2,461.45	81.8%	2.1%
		20	2,980.5	79.7%	6.1%
	D-P'	5	894.85	80.4%	1.3%
		10	1703.5	84.3%	3.0%
		15	2,356.25	82.2%	4.9%
		20	2,932	80.3%	6.8%
	D-NP'	5	743.2	73.6%	14.9%
		10	1,723.65	69.6%	2.4%
		15	2,432.1	80.4%	2.7%
		20	2,980.5	79.7%	6.1%
	N'	5	**927**	81.4%	0.0%
		10	**1,764.45**	78.6%	0.1%
		15	**2,495.6**	73.4%	1.6%
		20	**3,001.35**	71.3%	6.2%
Case 4	*S'*	5	**755.4**	83.2%	0.1%
		10	**1,412**	80.0%	0.3%
		15	2,025.7	75.9%	1.2%
		20	**2,503**	72.3%	3.7%
	D-P'	5	**755.4**	74.9%	0.1%
		10	1,372.9	78.5%	1.4%
		15	1,824.6	76.3%	5.5%
		20	2,454.6	74.0%	4.0%
	D-NP'	5	639.2	60.8%	11.5%
		10	1,376.3	62.5%	3.5%
		15	2,012.7	73.9%	1.6%
		20	**2,503**	72.3%	3.7%
	N'	5	**755.4**	84.3%	0.1%
		10	**1,412**	76.3%	0.3%
		15	**2,030.1**	72.1%	1.1%
		20	2,493.1	69.3%	3.3%
Case 5	*S'*	5	**1,019.4**	89.9%	0.0%
		10	1,919.5	89.8%	1.1%
		15	2,632.8	87.4%	3.9%
		20	3,148.5	84.5%	8.6%
	D-P'	5	1,015.8	87.4%	0.2%
		10	1,906.4	88.9%	1.3%
		15	2,514.8	87.3%	6.5%
		20	3,148.5	84.5%	8.6%
	D-NP'	5	969	73.6%	3.6%
		10	1,805.8	76.1%	6.0%
		15	2,598.6	85.9%	4.2%

		20	3,148.5	84.5%	8.6%
	N'	5	**1,019.4**	88.9%	0.0%
		10	**1,927.8**	84.6%	0.3%
		15	**2,679.5**	80.9%	2.8%
		20	**3,164.6**	77.6%	8.8%

The output of case 1 shows that *D-P'* yielded the most revenue for all hourly arrival rates, while *S'* and *N'* both exhibited the lowest rejection rates. Utilization rates are highest for *S'* and *N'* when λ = 5, 10, and for *D-P'* when λ = 15, 20. It can be noticed in case 4 that *D-P'* generated the least revenue when λ = 15, which is $205.50 less than the best result that was produced from *N* (the rejection rate was also the highest for *D-P'*). Interestingly, *D-P'* had the highest utilization rate in this situation. In cases 3 and 5, *N'* generated the most revenue, and was also effective in keeping rejection rates relatively low, of which are competitive with the other approaches.

In summary (with ties included), *N'* displayed 12 occurrences of the most revenue, while *S', D-P',* and *D-NP'* had seven, five, and two occurrences, respectively. Utilization rates were the highest in *S'* across 12 occurrences, and is followed by *D-P', N',* and *D-NP'* for eight, three, and two occurrences, respectively. Given the fact the utilization rates were not as high for *N'* compared to *S'* and *D-P',* this aligns with the observation in Section 4.4 where large sized tables tend to be dominant in the layout by applying *N'*, making it easier to accommodate parties of varying sizes. In other words, having more large sized tables makes it more likely for smaller sized parties to be assigned to them, thus reducing the overall utilization rate of the restaurant. One exception is evident in Table 4.4 for case 1 where most tables in the restaurant have four seats. The gradually decreasing utilization rates for *N'* are likely a result of smaller sized parties being assigned to larger tables. Rejection rates were lowest using *N'* in 17

occurrences, whereas *S'*, *D-NP'*, and *D-P'* were lowest for 12, four, and three of them, respectively.

Although the results do not display obvious patterns with respect to the maximum number of occupants that be seated, it can be observed that *S'* and *D-P'* contribute towards a majority of the occurrences of the best values for all metrics (after combining the with and without reduced capacity constraints counts). Table 4.10 displays the number of occurrences where the best values of revenue and utilization/rejection rates were observed relative to each approach.

Table 4.10: Number of occurrences for the best values of each metric – queue disabled

	Revenue	**Utilization Rate**	**Rejection Rate**
S'	**19**	17	18
D-P'	17	**27**	**21**
D-NP'	11	16	18
N'	14	3	19

In consideration of the revenue, *S'* and *D-P'* generated the most revenue 19 and 17 times, respectively, while *N'* and *D-NP'* did so 14 and 11 times, respectively. Similarly, *D-P'*, *S'*, and *D-NP'* had drastically more occurrences of the largest utilization rates (27, 17, and 16, respectively) in comparison to *N'* (3). The rejection rates are similar between each approach, but *D-P'* is responsible for a majority of the occurrences where the rejection rates were lowest (21).

4.6.2 Simulation output with queue enabled

This subsection reports the outcomes when parties are placed in a queue if there are no available tables inside of the restaurant upon arrival. It is assumed that 30% of guests will leave the restaurant if there are any parties in line upon their arrival, while the

remaining percentage of guests will wait to be seated. A maximum waiting time of each party is generated from a uniform distribution (a = 20, b = 60) to represent how long the party is willing to wait before leaving the restaurant (in minutes). When a table becomes available, parties in line are checked in the order in which they arrived to see if they can be seated. Parties are removed from the queue if they are seated at a table or if their time in line exceeds their tolerable wait time. Meal durations for parties are assigned in the same way as described in Section 4.6.1. Average waiting times (in minutes) are reported in substitute of the rejection rates, and are computed by taking the ratio between the cumulative wait time across all parties and the total number of parties that were placed in queue. The average waiting time across all replications are aggregated in a similar fashion to the metrics in Section 4.6.1.

Table 4.11 displays the simulation output with reduced capacity constraints enforced across all five cases of party size probabilities.

Table 4.11: Output with reduced capacity constraints enforced – queue enabled

		λ	**Revenue**	**Utilization Rate**	**Waiting Time**
Case 1	*S'*	5	**516.85**	49.6%	0.60
		10	983.55	55.1%	3.51
		15	1,306.85	56.4%	19.16
		20	**1,691.75**	53.6%	6.82
	D-P'	5	516.8	62.6%	0.59
		10	**1,001.05**	59.8%	4.18
		15	1,306.85	56.4%	19.16
		20	1,504.3	55.0%	2.15
	D-NP'	5	512.6	41.2%	2.54
		10	983.55	55.1%	3.51
		15	1,306.85	56.4%	19.16
		20	1,504.3	55.0%	2.15
	N'	5	488	51.8%	0.01
		10	933.8	48.6%	1.11
		15	**1,314.9**	47.0%	4.28

		20	1,591.65	47.2%	5.42
Case 2	*S'*	5	**740.4**	58.2%	0.93
		10	**1,381.7**	72.9%	6.06
		15	**1,829.65**	75.0%	12.52
		20	**2,119.75**	73.9%	10.88
	D-P'	5	731.75	76.7%	0.78
		10	**1,381.7**	72.9%	6.06
		15	**1,829.65**	75.0%	12.52
		20	2,053.9	75.7%	10.57
	D-NP'	5	708.9	59.4%	4.51
		10	**1,381.7**	72.9%	6.06
		15	**1,829.65**	75.0%	12.52
		20	2,053.9	75.7%	10.57
	N'	5	714.65	64.6%	4.26
		10	1,337.3	63.0%	9.43
		15	1,759.5	61.5%	9.68
		20	1,981.9	61.3%	10.54
Case 3	*S'*	5	**916.5**	65.1%	2.90
		10	**1,630.95**	78.7%	12.04
		15	**2,037.5**	77.0%	12.54
		20	**2,248**	76.7%	15.02
	D-P'	5	906.55	82.2%	5.04
		10	**1,630.95**	78.7%	12.04
		15	1,867.2	81.2%	19.08
		20	2,144.25	80.4%	18.22
	D-NP'	5	897.3	59.9%	8.34
		10	**1,630.95**	78.7%	12.04
		15	1,867.2	81.2%	19.08
		20	2,144.25	80.4%	18.22
	N'	5	729.9	72.1%	0.03
		10	1,375.45	70.6%	3.26
		15	1,868.1	70.8%	6.57
		20	2,123.2	70.3%	9.09
Case 4	*S'*	5	**755.6**	64.1%	1.27
		10	**1,394.4**	73.1%	10.28
		15	**1,814.5**	70.5%	10.81
		20	**2,076.5**	69.3%	10.67
	D-P'	5	754.6	78.1%	2.36
		10	**1,394.4**	73.1%	10.28
		15	1,682.8	72.7%	19.40
		20	1,643.5	70.4%	29.37
	D-NP'	5	734.4	57.9%	4.17
		10	**1,394.4**	73.1%	10.28

		15	1,682.8	72.7%	19.40
		20	1,643.5	70.4%	29.37
	N'	5	**755.6**	64.1%	1.27
		10	1,374.7	57.6%	6.82
		15	1,710.9	56.3%	9.71
		20	1,856.3	55.9%	14.50
Case 5	*S'*	5	1,004.1	70.6%	3.68
		10	**1,766.6**	78.4%	10.21
		15	2,094.8	76.7%	13.88
		20	**2,234.7**	76.6%	18.35
	D-P'	5	1,013.3	86.7%	5.82
		10	1,698.7	83.8%	13.89
		15	**2,098.6**	82.8%	15.98
		20	2,102.2	82.8%	23.67
	D-NP'	5	1,017.1	74.3%	3.90
		10	1,698.7	83.8%	13.89
		15	1,855.4	83.3%	22.56
		20	2,102.2	82.8%	23.67
	N'	5	**1,019.6**	83.6%	2.60
		10	**1,766.6**	78.4%	10.21
		15	2,094.8	76.7%	13.88
		20	**2,234.7**	76.6%	18.35

To summarize the output in Table 4.9 (with ties included), *S'* produced the most revenue 16 times, whereas *D-P', N', D-NP'* did so six, five, and four times, respectively. This outcome is consistent with the results from Table 4.6 where *S'* resulted in the most revenue for a majority of the occurrences (tied with *D-P'*). The best utilization rates come from *D-P'* and *D-NP'* for a majority of the runs (19 and 14, respectively), followed by *S'* in five occurrences. It is worth mentioning that *N'* was unable to produce the largest utilization rates, which is also the case in Table 4.8. The smallest average waiting times for parties were experienced in 16 occurrences by *N'*, followed by *S'* for six times.

Table 4.12 displays the simulation output without reduced capacity constraints enforced across all five cases of party size probabilities.

Table 4.12: Output without reduced capacity constraints relaxed – queue enabled

		λ	Revenue	Utilization Rate	Waiting Time
Case 1	*S'*	5	**516.55**	63.7%	0.62
		10	**1,009.1**	63.7%	1.87
		15	**1,440.95**	59.0%	4.37
		20	1,801.1	55.7%	7.06
	D-P'	5	514.95	60.2%	0.73
		10	1,004.45	63.2%	1.53
		15	1,426.35	61.2%	4.60
		20	**1,818.9**	58.5%	8.18
	D-NP'	5	500.1	47.4%	5.44
		10	994.7	50.8%	2.94
		15	1,447.1	57.3%	5.16
		20	1,801.1	55.7%	7.06
	N'	5	**516.55**	63.7%	0.62
		10	**1,009.1**	63.7%	1.87
		15	**1,440.95**	59.0%	4.37
		20	1,801.1	55.7%	7.06
Case 2	*S'*	5	**732**	78.8%	0.49
		10	1,398.05	72.7%	2.05
		15	1,995	75.3%	4.87
		20	2,479.5	72.9%	8.65
	D-P'	5	713.6	76.8%	4.15
		10	1,364.05	76.2%	11.01
		15	1,810.45	73.1%	0.10
		20	2,304.85	71.0%	2.08
	D-NP'	5	702.3	57.2%	5.49
		10	1,381.85	62.7%	5.37
		15	**2,006.35**	75.1%	5.43
		20	2,479.5	72.9%	8.65
	N'	5	731.85	67.5%	0.00
		10	**1,407.6**	64.9%	0.36
		15	2,004.55	61.6%	1.93
		20	**2,508.2**	59.6%	4.70
Case 3	*S'*	5	**927.05**	84.9%	0.04
		10	1,759.3	84.4%	1.11
		15	2,516.9	82.0%	5.28
		20	3,064.5	80.3%	8.57
	D-P'	5	917.4	80.2%	2.67
		10	1,709.5	84.5%	11.46
		15	2,306.6	82.6%	17.56
		20	2,976.75	81.2%	11.88
	D-NP'	5	818.4	74.6%	13.97

		10	1,745.25	69.9%	3.68
		15	**2,469.1**	80.7%	5.91
		20	3,064.5	80.3%	8.57
	N'	5	927	81.4%	0.00
		10	**1,765.75**	78.6%	0.19
		15	2,522.45	73.4%	2.67
		20	**3,114**	71.6%	5.53
Case 4	*S'*	5	**757.2**	83.2%	0.26
		10	**1,429.3**	80.3%	1.85
		15	**2,066.2**	76.0%	4.98
		20	2,542.5	72.7%	9.32
	D-P'	5	**7,57.2**	74.9%	0.26
		10	1,393.2	78.8%	7.54
		15	1,745	76.7%	26.31
		20	2,522.8	74.9%	9.48
	D-NP'	5	680.3	61.2%	10.26
		10	1,416.4	62.6%	4.32
		15	2,042	74.2%	5.37
		20	2,542.5	72.7%	9.32
	N'	5	**757.2**	84.3%	0.26
		10	**1,429.3**	76.5%	1.85
		15	2,058.8	72.1%	5.21
		20	**2,566.1**	70.1%	8.34
Case 5	*S'*	5	1,019.4	89.9%	0.00
		10	**1,934**	89.9%	3.10
		15	2,705.2	87.6%	8.64
		20	3,246.1	85.9%	11.03
	D-P'	5	**1,026.9**	87.4%	0.39
		10	1,925.4	88.7%	3.46
		15	2,569.6	88.0%	12.50
		20	3,246.1	85.9%	11.03
	D-NP'	5	1,024	74.0%	4.49
		10	1,883.5	76.7%	6.51
		15	2,694.8	86.2%	8.71
		20	3,246.1	85.9%	11.03
	N'	5	1,019.4	88.9%	0.00
		10	1,926.8	84.6%	1.06
		15	**2,730.5**	81.0%	5.31
		20	**3,323.5**	78.6%	7.74

Unlike any of the previous results (with ties included), it turns out that *N'* most frequently

produced the most revenue (12 occurrences) compared to the other approaches. *S'* and *D-P'* are tied for the most incidences of the largest utilization rates (10 occurrences).

4.7 Conclusion

In this chapter, an optimization model is proposed to assist restaurant owners who need to comply with the safety rules under the pandemic of an infectious disease. The objective of the model is to maximize the capacity of the dining area by rearranging tables while satisfying social distancing and reduced capacity requirements. Several cases are presented to demonstrate the effectiveness of the proposed model. From this, a two-stage stochastic program with recourse model is then proposed for optimizing TMP and PMP concurrently in consideration of maximizing the expected revenue instead of the capacity. To the author's knowledge, this is the first time any study has proposed such an approach that allows both problems to be expressed and solved within a single optimization model. The performance of the proposed models is evaluated in a simulation study, where the results show that the stochastic program more frequently yields the best performance from an operational (i.e., revenue, utilization rate) and customer service perspective (i.e., rejection rate) when waiting is not permitted.

Solutions obtained from the proposed approaches can benefit restaurant practitioners by improving their revenue stream while accounting for seating assortment and arrangement simultaneously in consideration of customer demand, as well as newly emerging criteria due to the recent global pandemic, such as social distancing between parties and reduced capacity constraints. Research extensions include exploring valid inequalities to further improve the computational efficacy. Also, allowing for tables to be

combined and incorporating queueing theory dynamics in the stochastic programming model can be investigated.

CHAPTER 5: AUTOMATED LAYOUT GENERATION IN CONSIDERATION OF ADJACENCY SPECIFICATIONS

5.1 Introduction

Architects are heavily involved in the construction process of new facilities from start to finish. During the early phase of a construction project, common tasks that are delegated to the architect(s) include (but not limited to) requirements gathering, data collection and analysis, defining physical space specifications, etc. The accumulation of these deliverables is typically referred to as the *design program*, and is used to qualify and quantify the stakeholders' needs during the pre-design process (Karlen and Fleming, 2016). The design program is a critical component to the project since it has a strong influence on the overall quality of the planning and design processes. Therefore, it is essential that all parties are satisfied with its contents before progressing because it can become costly to modify the program as the project matures.

During the pre-design process, departments to be placed in the facility are identified, and *relationship diagrams* are created to visualize the program specifications such as the desired adjacency between departments, travel or circulation patterns, relative sizes of spaces, etc. These specifications are derived collaboratively between the architect(s) and stakeholders based on a variety of factors, such as facility codes, acoustics, spatial quality, architect experience, etc. (Karlen and Fleming, 2016). Once the specifications are defined, preliminary layout designs can be generated to provide a glimpse as to how physical spaces will be arranged within the facility interior. *Bubble diagram* is an example of a sketching technique that architects use to quickly explore a large number of layout options. However, bubble diagrams can sometimes ignore the shape of the facility envelope, thus making it difficult in certain cases to transform the

bubble diagram into a more realistic depiction of the layout design. In addition, it can be time consuming to transform the bubble diagram into an actual layout design since it is possible that not all of the specifications can be satisfied simultaneously due to physical space limitations.

To assist architects in the aforementioned transformation process, researchers have proposed various techniques to automate the layout generation process. Most of these techniques require the user to provide the physical space specifications as input, and the system will output a layout design for the architect to evaluate. One important specification is the adjacency preference between spaces, which can be classified into one of three categories: *adjacency*, *proximity*, and *separation*. Adjacency is used when it is desired for two spaces to be directly accessible to each other. If adjacent, they must share a minimum common boundary length (e.g., the minimum size of the door or opening between the spaces). When two spaces are desired to be proximate, the distance between them is within a given threshold value. If the distance is larger than the threshold value, two spaces are considered separated.

Proximity and separation specifications have been addressed for facility layout planning problems where departments can be placed anywhere within the facility (Tari and Neghabi, 2015; Neghabi and Tari, 2016; Klausnitzer and Lasch, 2019). Separation is synonymous with minimum clearances in the engineering literature, and is frequently encountered in row layout problems where it is of interest to place departments along a series of rows to minimize the material flow cost (Simmons, 1969; Heragu and Alfa, 1992; Chung and Tanchoco, 2010; Zhang and Murray 2012). In the context of architecture, studies on the automated layout generation typically address only the

adjacency constraints, whereas the proximity and separation constraints are ignored. However, it should be pointed out that there are certain situations where it is not required for pairs of departments to be placed directly beside one another, but rather to be within a tolerable distance. Similarly, it might be necessary for certain departments to be separated by a minimum distance due to a variety of factors, such as safety, efficiency, noise, etc. Ignoring these specifications can result in less attractive layout designs to be generated, thus consuming additional time and resources before arriving to a final design which all parties are satisfied with.

In this chapter, a mathematical optimization model is first proposed to accommodate as many adjacency specifications as possible with the intention of generating a layout design that faithfully reflects a given relationship diagram. The resulting layout itself can be selected as a final design, but more practically, it can serve as a reference design while the architect explores various design options. While the layout design itself offers an arrangement of departments, a separate effort needs to be subsequently made to create circulation in the facility. Thus, the second part of this chapter considers another optimization problem for configuring the aisle structure in the facility. This allows for the layout design to better represent what the facility interior will look like and how occupants will navigate within it. Therefore, travel distances between departments using the aisle structure can be used to calculate the overall cost. The remainder of this chapter is organized as follows. In the next section (Section 5.2), a literature review of relevant studies is provided. In Section 5.3, the proposed MILP for arranging the departments in the facility is presented. The MILP model for determining the optimal aisle structure based on the arrangement of departments is introduced in

Section 5.4, and Section 5.5 presents a computational study that was performed for two case studies. Final remarks are addressed in Section 5.6.

5.2 Literature Review

Various solution methodologies have been employed for automating the layout generation process for architectural design problems, including (but not limited to) *mathematical optimization*, *machine learning*, and *heuristics*. Each approach has their respective strengths and weaknesses, and it is up to the architect to decide which approach to use based on (a) the desired solution quality, (b) the amount of time to spend on finding a solution, and (c) the assumptions that are made for representing the final layout design (i.e., rectangular-shaped departments, discretization of the facility into cells, convex/non-convex shaped facility envelope, etc.). Another step in the layout generation process consists of configuring the aisle structure to allow occupants to navigate the facility. Approaches have been developed that integrate layout and aisle generation into a single model, as well as considering both problems separately where aisle generation serves as a post-processing technique after a block layout is obtained. A review of these approaches and the corresponding shortcomings in the literature are discussed in the subsequent sections.

5.2.1 Mathematical optimization

Koopmans and Beckmann (1957) introduced the QAP where the facility was represented as a grid with cells of equal size, and departments are assigned to cell-based units to minimize the material flow cost of the manufacturing facility. Sahni and Gonzales (1976) showed the NP-hardness of QAP, so it would be difficult to find an optimal solution using exact methods as the size of the layout design problem increases.

Glover et al. (1985) used mixed-integer and nonlinear programming for generating bubble diagrams based on architectural program requirements. MILP was used to cluster the departments into subsets where the relationship between departments in each subset is maximized. NLP was then used for creating bubble diagrams based on topological requirements and the output of the MILP. This sequential approach was intended to assist the architects in understanding the layout design problem by allowing the user to manipulate the output from both models as they see fit.

Montreuil et al. (1989) applied a LP model to transform the design skeletons created by layout planners into actual layout designs based on the relative positioning of departments in the design skeleton. Since not all positioning requirements can be satisfied in the layout (thus leading to model infeasibility), additional decision variables and constraints are added to guarantee a feasible solution is produced by extending the boundaries of the facility. Then, the layout planner determines which requirements are not satisfied in the resulting layout. This extension to the model may not be practical for the case where the surface area of the facility is fixed and cannot be increased due to limited space in the surrounding environment, thus resulting in impractical layouts to be generated. Montreuil (1990) proposed another MILP model where departments can be freely placed anywhere within the facility. However, its excessive computational demand made the implementation of the model impractical.

Meller et al. (1998) modified the model proposed by Montreuil to reduce the computational complexity in order to solve larger sized problem instances. Valid inequalities are introduced, and it is shown that they are helpful in improving the solution time. Sherali et al. (2003) enhanced the model presented by Meller et al. by introducing

additional valid inequality constraints, reducing problem symmetry, and constructing the partial convex hull representation of the non-overlapping and separation constraints. An improved linearization of the nonlinear departmental area constraints is also introduced, and it is shown that the overall accuracy of the layout design is increased as a result.

Michalek et al. (2002) formulated two models for optimizing the geometric and topological aspects of a facility. Their procedure generates a layout design using a combination of evolutionary algorithms and sequential quadratic programming to iteratively update the facility topology using an initial layout diagram as input. Architectural objectives are considered during the optimization, including heating, cooling, and lighting costs, as well as space utilization. Model infeasibility is prevented by assigning a penalty to the layout score, thus making it a less attractive solution to the layout planner. Doing so allows for a feasible solution to still be produced even if it is deemed impractical by the layout planner. Michalek and Papalambros (2002) extended the work of Michalek et al. by developing an interactive system that allows the user to add/remove departments, constraints, objectives, etc. during the pre-design phase until a satisficing solution is found. Kamol and Krung (2005) applied MILP for generating a layout design based on a functional diagram that is provided by the user. Alternative layout designs are created by forcing departments to be placed in the top-left region of the facility based on the architect's preference. The facility is represented as a grid with cells of unit-length in consideration of orthogonal boundary shapes, and rectangular-shaped departments are placed accordingly.

Tari and Neghabi (2015) proposed an alternative model where adjacency is the primary criterion in layout design generation. The authors define two departments as

being adjacent if they are placed within a tolerable distance of each other and share a minimum common boundary length. The material flow between departments is used to represent which pairs have a higher importance for being adjacent in the layout design. The adjacency between each pair of departments is measured by so-called adjacency degree, which is bounded between 0 and 1, and the objective is to maximize the weighted sum of those adjacency degrees.

Wu et al. (2018) proposed a hierarchical framework for generating the layout design of facility interiors using MIQP. The facility is decomposed into a set of sub-domains where each sub-domain consists of multiple rectangles that serve a similar functional purpose. The layouts of the sub-domains are optimized iteratively to remove irregularities from the layout, including poor-space utilization, inaccessible areas, etc. Travel distances between departments are unaccounted for in the model since the objective function considers the optimization of space utilization inside of the facility. Adjacency requirements for department pairs are specified by adding additional constraints to the model. It is worth noting that model infeasibility can occur if an excess number of adjacency requirements are included for a particular problem instance since binary decision variables and disjunctive constraints are used to enforce them.

5.2.2 Heuristic methods

Río-Cidoncha et al. (2007) integrated slicing trees, feng shui, and the shortest distance problem for determining the position, location, and orientation of departments in a layout design. After creating the layout design, a routing module is executed to determine the flow of materials/occupants within the space. Verma and Thakur (2010) developed a GA that generates floor plans for single and multi-story facilities.

Topological solutions are generated first using design requirements from the layout planner, and then a dimensional analysis is performed to determine the dimensions of each department. Evacuation time is one of the criteria for evaluating the generated layouts since the authors consider the residential setting.

Merrell et al. (2010) applied the Bayesian network theory to generate bubble diagrams using a list of high-level requirements that are provided by the layout planner as input. The Bayesian network creates the bubble diagram by referencing a corpus of layouts with a similar facility type and is optimized using structure learning. The Metropolis algorithm is then performed using the output from the Bayesian network to generate the layout design in consideration of four factors, namely accessibility, area and aspect ratio, the number of floors, and department shapes.

Guo and Li (2017) proposed an automation method to create facility topology and layout design by applying a set of evolutionary algorithms. A multi-agent system is applied first to make the functional diagram of the facility more compact and easier to interpret. The layout design is then created using the topology from the multi-agent system while accounting for important architectural criteria and objectives, such as aspect ratio, facility shape, and energy consumption. Laignel et al. (2021) integrated constraint programming and GA for their automated apartment plan generation. The facility is discretized into a grid with unequally sized cells, and the departments are assigned to cells so that architectural and functional constraints are satisfied. Multiple layout designs are generated and clustered together to make it easier to perform the evaluation.

Grason (1971) used the graph theory for generating layout designs in consideration of adjacency and departmental dimension requirements. A planar graph is

generated using a placement algorithm, and its dual graph is constructed to determine the relative location of departments with respect to one another in the layout design. The department dimensions are not considered until the dual graph is created, which can result in infeasible layouts for some cases. Ruch (1978) proposed a three-phase interactive procedure for generating layout designs. Graph theory is used in the first phase for generating planar graphs where the nodes and edges represent departments and departmental adjacencies, respectively. The second phase consists of transforming the planar graph to a bubble diagram (drawn to scale based on departmental area requirements), which is then followed by placing each bubble inside of a minimum area rectangular envelope to yield a final layout design in the third phase.

5.2.3 Machine learning

Yeh (2006) proposed a heuristic framework, so-called annealed neural network that combines Hopfield neural networks and SA. It was observed that the model is sensitive to the selection of algorithmic parameters, thus making it difficult to systematically explore the solution space. Zawidski et al. (2010) presented a decision support system for generating *functional layouts* of a facility. The functional layout is represented as a unit grid where departments are assigned to sets of neighboring cells in consideration of topological requirements, geometrical complexity, and corridor size. Functional layouts are converted to architectural layouts to provide the user with a more detailed representation of the layout design within the facility envelope. A machine learning method is then applied to classify the layout as proper or improper according to internal communication criteria.

Chaillou (2019) proposed a floorplan generation framework that utilizes a three-phase stack of deep networks. The framework allows for the generation of a floorplan in consideration of its footprint, layout generation of the departments, and furniture placement simultaneously. Hu et al. (2020) used deep learning to suggest layout designs based on user-defined constraints. Adjacency graphs are stored in a database and referenced by the deep learning procedure, which represent actual floor plans that incorporate architecture-design principles. The model will readjust the suggested layouts based on the shape of the facility and then automatically generate the resulting layout design.

5.2.4 Aisle generation

Consideration of the aisle structure is typically ignored in the approaches discussed so far since the layout design problem alone is challenging enough already. For approaches that do address aisle generation in conjunction with layout generation, it is common for the aisle structure to be constructed with respect to the boundaries of departments after a layout is obtained based on certain criteria and constraints. (Norman, Arapoglu, and Smith, 2001; Wu and Appleton, 2002; Xiao et al., 2017; Friedriech et al., 2018). Given a block layout, Lee et al. (2009) applied the network flows optimization for minimizing the material flow cost while determining the locations of input/output points of each department. Peng et al. (2016) applied integer programming for optimizing the navigation, which was applied to urban design problems. They incorporated a variety of functional specifications, such as the length of the network and travel distance from points of interest to a sink node, and introduced constraints to make the flow look more realistic, including dead-end avoidance, zig-zag avoidance, etc. Li and Hua (2019)

extended the work of Peng et al. by proposing a formulation with fewer variables. Other approaches attempted to optimize the layout design and aisle structure simultaneously for minimizing the material flow cost, but typically require prohibitively expensive computational efforts (Klausnitzer and Lasch, 2019; Pourvaziri, Pierreval, and Marian, 2021). Hence, these integrated approaches simplify the problem by prescribing the number of vertical and horizontal aisle segments a priori, which would result in suboptimal solutions.

5.2.5 Shortcomings in the literature and contribution

From this literature review, a few shortcomings were identified regarding the current state of the automated layout generation within the architecture community. First, a large portion of the literature in the automated layout generation only consider adjacency specifications between departments, while the proximity and separation requirements are overlooked. Second, current MILP models can handle a limited number of adjacency specifications to avoid model infeasibility and worsen the computational tractability by including more binary variables and constraints. Examples of the second shortcoming can be found in Kamol and Krung (2005), Tari and Neghabi (2015), and Wu et al. (2018), where each adjacency requirement is specified by a binary variable to indicate if the pair of departments are adjacent in the layout. It results in a large number of binary variables, which is likely to increase the overall computational complexity. Doing so also puts the MILP at risk of infeasibility since it may not be possible to satisfy all requirements simultaneously in the layout design.

Another shortcoming is that existing optimization studies rarely consider regulatory requirements set forth by the International Building Code (2021). For

example, facility exits must be separated by a threshold value, which is dependent on the facility dimensions. Hosseini et al. (2020) reference the International Building Code for satisfying minimum wall lengths and area ratios of departments for temporary housing units. Gao et al. (2020) consider an evacuation problem where the configuration of doorways is optimized for reducing the evacuation time based on doorway separation requirements set forth by the Chinese Fire Code. It is assumed that the facility layout and its corresponding aisle structure are known in advance (along with the placement of already existing doorways).

While addressing the aforementioned shortcomings of the automated layout generation techniques, a two-phase procedure for layout design and circulation optimization is proposed in this study. For the first phase, a penalty-based MILP model that can produce a feasible layout design without requiring additional binary decision variables for addressing the adjacency specifications is proposed to produce a block layout. To be more precise, continuous variables, namely *deviational* variables, for pairs of departments are introduced to represent the extent of the deviations from the desired adjacency preferences. Accordingly, penalties are imposed to positive values of the deviational variables. Removing the binary variables for representing the adjacency specifications is advantageous for: (a) reducing the computational burden for solving the problem, and (b) allowing a large number of specifications to be considered simultaneously during the optimization while avoiding model infeasibility. Lastly, the proposed model also addresses the proximity and separation specifications in addition to the adjacency, thus extending the applicability of the proposed automated layout generation model.

In the second phase, an aisle generation problem is solved given the block layout generated in the first phase. The aisle generation problem is not only to generate aisles, but also to determine the optimal locations of department doorways and the facility accessways (i.e., main entrance and emergency exits), while minimizing the total travel distance from the doorways to the accessways. Detailed descriptions of those models are presented in the next two sections, respectively.

5.3 Phase I: Layout Design

In this section, an MILP model is presented for generating layout design alternatives based on the adjacency specifications furnished by the bubble diagram and adjacency matrix. The following notation of the problem parameters is used to present the model:

n: number of departments

$I = \{1,2, \dots, n\}$: index set of departments

$W = \{x, y\}$: set of axes for two-dimensional layout design problem

L^w: length of the facility along direction $w \in W$

a_i: area requirement of department $i \in I$

lb_i^w / ub_i^w: lower/upper bounds on the half-length of department $i \in I$ along direction $w \in W$

$\bar{x}$: various points that are used for generating tangential supports of the nonlinear area equations

A: set of unordered pairs $\{i, j\}$ of departments i and j that are preferred to be adjacent

P: set of unordered pairs $\{i, j\}$ of departments i and j that are preferred to be proximate

S: set of unordered pairs $\{i, j\}$ of departments i and j that are preferred to be separated

T: the set of adjacency specification (i.e., $T = \{A, P, S\}$)

f_{ij}^{t}: per unit distance penalty imposed when deviating from adjacency specification $t \in T$ for $\{i, j\} \in t$

r_{ij}: minimum common boundary length for $\{i, j\} \in A$ such that $r_{ij} \leq 2 \min\{lb_i^w, lb_j^w\}$

p_{ij}: maximum boundary-to-boundary distance for $\{i, j\} \in P$

s_{ij}: minimum boundary-to-boundary distance for $\{i, j\} \in S$

$M \gg 1$: a large constant.

Specification sets A, P, and S are populated directly from the bubble diagram and the adjacency matrix prior to the optimization. Note that department pairs without any adjacency specification can simply be excluded from the specifications sets. Parameters lb_i^w and ub_i^w are calculated using the approximation by Sherali et al. (2003) for the nonlinear area equations (reference the paper for further details). In the proposed model, the l_1 norm distance metric is applied for addressing the proximity and separation specifications. The Chebyshev distance metric (i.e., l_∞ norm) can be applied by replacing certain constraints and introducing additional decision variables, which will be briefly discussed later. Note that applying either of these distance metrics will likely result in different layout design alternatives for the same problem instance. The Euclidean distance metric (i.e., l_2 norm) is excluded for two reasons: 1) l_1 is more stuitable for addressing distances between rectangular shaped departments as aisles are built on the boundaries, and 2) l_2 results in a nonlinear and nonconvex problem, which significantly increases the computational complexity of the problem.

Figure 5.1 depicts the proper classification of adjacency between departments i and j. Figures 5.2 and 5.3 illustrate the concept of proximity and separation, respectively. Note that the depiction of the shaded region for P and S is dependent on the distance metric that is used for performing the layout optimization.

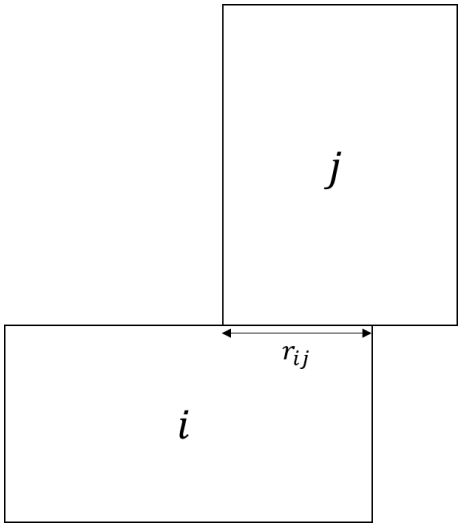
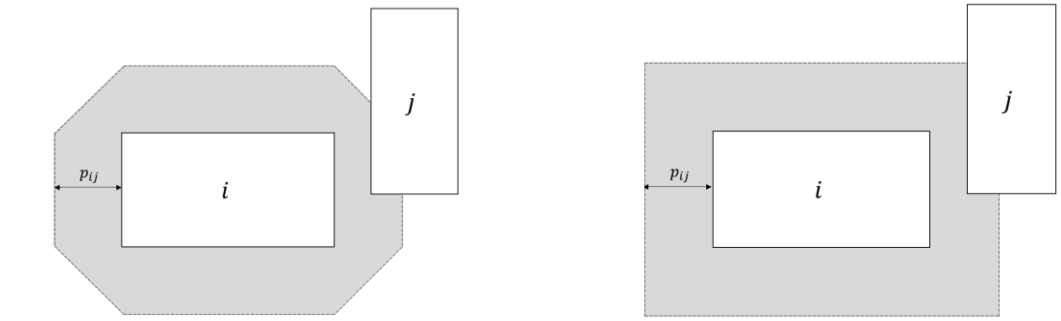


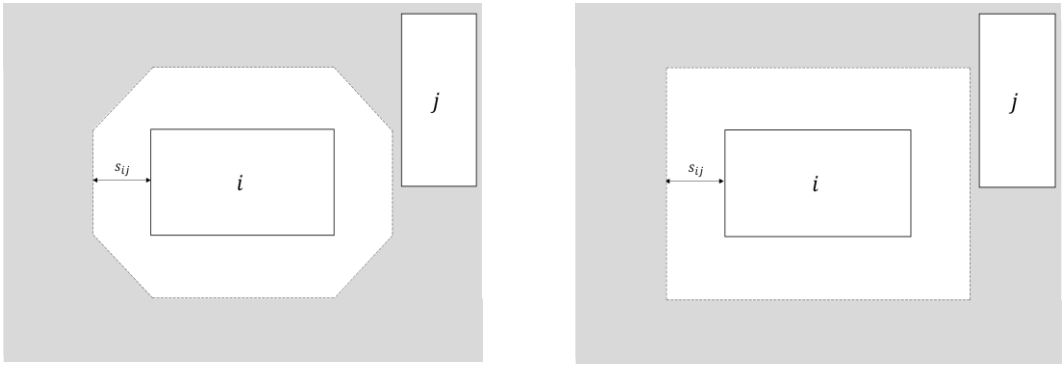
Figure 5.1: Graphical depiction of satisfied adjacency specification



(a)
l_1 norm distance

(b)
l_∞ norm distance

Figure 5.2: Graphical depiction of satisfied proximity specification



(a)
l_1 norm distance

(b)
l_∞ norm distance

Figure 5.3: Graphical depiction of satisfied separation specification

Observe in Figure 5.1 that the borders of departments i and j are touching and share a common boundary length of r_{ij} units, thus satisfying the desired adjacency specification (of course the common boundary length can exceed r_{ij} for satisfying the adjacency). However, if the common boundary length is less than r_{ij} (even by a slight amount), then the adjacency is not satisfied. Figure 5.1 considers the case where departments i and j adjoin each other along the x-direction, thus requiring these departments to be touching along the y-direction.

In Figures 5.2 (a) and 5.2 (b), department j overlaps the shaded region that is surrounded by department i, indicating that these two departments are within proximity of each other. Note in Figure 5.2 (a) that the l_1 norm generates an octagon shaped proximity region, whereas a rectangular shaped region is produced using the l_∞ norm in Figure 5.2 (b). In Figures 5.3 (a) and 5.3 (b), the entirety of department j must be within the shaded region to be considered as separated from department i.

Given the problem parameters defined above, the proposed model will produce a layout design solution by determining the values of the following decision variables:

c_i^w: center point of department $i \in I$ along direction $w \in W$

l_i^w: half-length of department $i \in I$ along direction $w \in W$

d_{ij}^w: center-to-center distance between departments i and j along direction $w \in W$ $(i, j \in I, i < j)$

D_{ij}^w: boundary-to-boundary distance between departments i and j along direction $w \in W$ $(\forall \{i, j\} \in P \cup S)$

$\Delta_{ij,w}^a$: deviation from the adjacency condition for $\{i, j\} \in A$ in direction $w \in W$

Δ_{ij}^p: deviation from the proximity condition for $\{i, j\} \in P$

Δ_{ij}^s: deviation from the separation condition for $\{i, j\} \in S$

z_{ij}^{w}: 1 if department i precedes department j along direction $w \in W$; 0 otherwise $(i, j \in I, i < j)$

σ_{ij}^{w}: 1 if the center point of department i precedes the center point of department j along direction $w \in W$; 0 otherwise $(i, j \in I, i < j)$

o_{ij}^{w}: 1 if departments i and j are overlapping along direction $w \in W$; 0 otherwise $(i, j \in S, i < j)$.

Note that deviational variables $\Delta_{ij,w}^{a}$, Δ_{ij}^{p}, and Δ_{ij}^{s} determine if the adjacency specification between departments i and j is satisfied. If the adjacency specification is satisfied, then the corresponding deviational variable will equal 0 since its value is minimized in the objective function. On the other hand, the deviational variable will take on some positive value when the specification is not satisfied. The idea is to minimize the weighted sum of all deviational variables to find the layout design where the deviation between the bubble diagram and adjacency matrix is minimized. The definitions of deviations are as follows.

Definition 5.1 (deviation for adjacency):

(5.1.1) If two departments do not overlap in both x- and y-directions, the deviation for adjacency is defined as the boundary-to-boundary distance between two departments plus the minimum common boundary length (r_{ij}).

(5.1.2) If two departments overlap in w-direction for $w \in \{x, y\}$, the deviation for adjacency is defined as the sum of two direction-specific deviations for adjacency: (a) $\max\{0, r_{ij} - \text{length of overlap}\}$ in w-direction, and (b) the boundary-to-boundary distance between two departments in direction $\overline{w} \in W \backslash \{w\}$.

Definition 5.2 (deviation for proximity):

The deviation for proximity is defined as max{0, the boundary-to-boundary distance between two departments$-p_{ij}$}.

Definition 5.3 (deviation for separation):

The deviation for separation is defined as max{0, s_{ij} − the boundary-to-boundary distance between two departments}.

Now, the proposed MILP model can be stated as follows:

$$\text{Minimize} \quad \sum_{\{i,j\}\in A} f_{ij}^{a}(\Delta_{ij,x}^{a} + \Delta_{ij,y}^{a}) + \sum_{\{i,j\}\in P} f_{ij}^{p}\Delta_{ij}^{p} + \sum_{\{i,j\}\in S} f_{ij}^{s}\Delta_{ij}^{s} \tag{5.1}$$

subject to

$$a_i l_i^x + 4\bar{x}^2 l_i^y \geq 2a_i\bar{x} \qquad \forall lb_i^x \leq \bar{x} \leq ub_i^x \tag{5.2}$$

$$\sum_{w\in W}\left(z_{ij}^{w} + z_{ji}^{w}\right) = 1 \qquad \forall i,j \in I, i<j \tag{5.3}$$

$$c_i^w + l_i^w \leq c_j^w - l_j^w + M(1 - z_{ij}^w) \qquad \forall i,j \in I, i \neq j, w \in W \tag{5.4}$$

$$-d_{ij}^w \leq c_j^w - c_i^w \leq d_{ij}^w \qquad \forall i,j \in I, i<j, w \in W \tag{5.5}$$

$$d_{ij}^w \leq c_j^w - c_i^w + M(1 - \sigma_{ij}^w) \qquad \forall i,j \in I, i<j, w \in W \tag{5.6}$$

$$d_{ij}^w \leq c_i^w - c_j^w + M\sigma_{ij}^w \qquad \forall i,j \in I, i<j, w \in W \tag{5.7}$$

$$l_i^w \leq c_i^w \leq L^w - l_i^w \qquad \forall i \in I, w \in W \tag{5.8}$$

$$lb_i^w \leq l_i^w \leq ub_i^w \qquad \forall i \in I, w \in W \tag{5.9}$$

$$\Delta_{ij,w}^{a} \geq r_{ij}\left(z_{ij}^{w'} + z_{ji}^{w'}\right) - \left[\left(l_i^w + l_j^w\right) - d_{ij}^w\right] \qquad \forall \{i,j\} \in A, w \in W, w' \in W \setminus \{w\} \tag{5.10}$$

$$D_{ij}^w \geq d_{ij}^w - l_i^w - l_j^w \qquad \forall \{i,j\} \in P, w \in W \tag{5.11}$$

$$\Delta_{ij}^{p} \geq \left(D_{ij}^x + D_{ij}^y\right) - p_{ij} \qquad \forall \{i,j\} \in P \tag{5.12}$$

$$d_{ij}^w - l_i^w - l_j^w \leq M\left(1 - o_{ij}^w\right) \qquad \forall \{i,j\} \in S, w \in W \tag{5.13}$$

$$d_{ij}^w - l_i^w - l_j^w \geq -Mo_{ij}^w \qquad \forall \{i,j\} \in S, w \in W \tag{5.14}$$

$$D_{ij}^w \leq d_{ij}^w - l_i^w - l_j^w + Mo_{ij}^w \qquad \forall \{i,j\} \in S, w \in W \tag{5.15}$$

$$D_{ij}^w \leq M\left(1 - o_{ij}^w\right) \qquad \forall \{i,j\} \in S, w \in W \tag{5.16}$$

$$\Delta_{ij}^{s} \geq q_{ij} - \left(D_{ij}^x + D_{ij}^y\right) \qquad \forall \{i,j\} \in S \tag{5.17}$$

$$o_{ij}^x + o_{ij}^y \leq 1 \qquad \forall i,j \in \{P\} \cup \{S\}, i<j \tag{5.18}$$

$$\Delta_{ij,w}^{a} \geq 0 \qquad \forall\{i,j\} \in A, w \in W \qquad (5.19)$$

$$\Delta_{ij}^{p} \geq 0 \qquad \forall\{i,j\} \in P \qquad (5.20)$$

$$\Delta_{ij}^{s} \geq 0 \qquad \forall\{i,j\} \in S \qquad (5.21)$$

$$D_{ij}^{w} \geq 0 \qquad \forall\{i,j\} \in P \cup S, w \in W \qquad (5.22)$$

$$c_i^w, l_i^w, d_{ij}^w \geq 0 \qquad \forall i,j \in I, i < j, w \in W \qquad (5.23)$$

$$o_{ij}^{w} \in \{0,1\} \qquad \forall\{i,j\} \in S, w \in W \qquad (5.24)$$

$$z_{ij}^{w} \in \{0,1\} \qquad \forall i,j \in I, i \neq j, w \in W. \qquad (5.25)$$

The objective function (5.1) minimizes the weighted sum of the deviational variables for the adjacency specifications. When (5.1) is equal to 0 in an optimal solution, this indicates that all specification constraints are satisfied in the generated layout design. On the other hand, a positive optimal value indicates that at least one specification is not satisfied in the layout design. These particular specifications can be identified by looking at the positive values of the deviational variables in the optimal solution.

Constraint (5.2) is the linearization constraint proposed by Sherali et al. for the nonlinear area equations. Values of $\bar{x}$ are generated using the following expression:

$$\bar{x} = lb_i^x + \frac{\lambda}{(\Delta - 1)}(ub_i^x - lb_i^x) \qquad \forall \lambda = 0,1,\ldots,\Delta - 1 \text{ for any selected integer } \Delta \geq 2.$$

where Δ represents the number of tangential supports to include in the model. A larger value of Δ improves the quality of the approximation, but also increases the computational burden of the model. Constraints (5.3)-(5.4) prevent departments from overlapping each other in the layout design. Constraints (5.5)-(5.7) assign the exact center-to-center distance in the direction w (i.e., $\left|c_i^w - c_j^w\right|$) to d_{ij}^w. While constraint (5.5) enforces the lower bounds on the distance (d_{ij}^w), constraints (5.6)-(5.7) enforce the upper bounds with disjunction so that $d_{ij}^w = \max\left\{c_i^w - c_j^w, c_j^w - c_i^w\right\}$. Constraint (5.8) ensures

that departments are placed within the facility envelope. Constraint (5.9) enforces the lower and upper bounds for the half-lengths of each department in the x- and y-directions.

As shown in Proposition 5.1 later, constraint (5.10) prescribes the lower bound on the deviational variable for adjacency, $\Delta_{ij,w}^{a}$, for $\{i,j\} \in A$. Constraints (5.11) enforces the lower bound on the nonnegative variable D_{ij}^{w}, which represents the distance between boundaries of departments i and j for $\{i,j\} \in P$. In turn, (5.12) prescribes the lower bound on the deviational variable for proximity. While deviations for adjacency and proximity specifications need lower bounds observing that the objective function is minimized, the deviation for separation needs extra consideration since the distance is desired to be longer for the objective function to be minimized. Hence, instead of enforcing a lower bound on the distance between boundaries, an upper bound is enforced for D_{ij}^{w} in constraints (5.13)-(5.16). In (5.13)-(5.14), the value of the binary variable o_{ij}^{w} is determined based on the sign of $d_{ij}^{w} - l_{i}^{w} - l_{j}^{w}$. Recall that o_{ij}^{w} determines whether departments i and j are overlapping in w-direction. If $d_{ij}^{w} - l_{i}^{w} - l_{j}^{w} > 0$, then two departments do not overlap in w-direction since the center-to-center distance is greater than the sum of half-lengths, and accordingly, constraint (5.13) results in $o_{ij}^{w} = 0$. On the other hand, $d_{ij}^{w} - l_{i}^{w} - l_{j}^{w} < 0$ indicates that two departments overlap in w-direction, and constraint (5.14) effectively assigns $o_{ij}^{w} = 1$. When $o_{ij}^{w} = 0$, i.e., no overlap between departments i and j in w-direction, (5.15) prescribes the upper bound on D_{ij}^{w} as the exact distance between boundaries, i.e., $d_{ij}^{w} - l_{i}^{w} - l_{j}^{w}$. Otherwise, the distance between boundaries becomes zero as enforced in (5.16). In constraint (5.17), the lower bound on the deviational variable for separation is prescribed. Again, note that the sign of the

distances between boundaries is negative in (5.17). Therefore, minimizing the deviational variable corresponds to maximizing the distance between boundaries until the threshold is reached, which is effectively constrained by the upper bounds in (5.15)-(5.16). Constraint (5.18) prevents departments from overlapping in more than one dimension. Constraints (5.19)-(5.25) are for the nonnegativity and integrality of decision variables.

As mentioned above, the deviation for adjacency is determined by constraint (5.10). The following proposition provides its correctness.

Proposition 5.1: Consider MILP in (5.1)-(5.25). Suppose that the minimum common boundary satisfies $r_{ij} \leq 2\min\{lb_i^w, lb_j^w\}$. Then, constraint (5.10) in conjunction with minimizing the objective function guarantees that $\left(\Delta_{ij,x}^a + \Delta_{ij,y}^a\right)$ results in the deviation for adjacency as defined in Definition 5.1.

Proof: First, consider a pair of departments $\{i, j\} \in A$ that do not overlap in any direction. Since the pair do not overlap, the boundary-to-boundary distance in w-direction is given by $d_{ij}^w - \left(l_i^w + l_j^w\right)$. Hence, the boundary-to-boundary distance between i and j is $\sum_{w \in W}\left[d_{ij}^w - \left(l_i^w + l_j^w\right)\right]$. From (5.10), we have $\Delta_{ij,x}^a \geq r_{ij}\left(z_{ij}^y + z_{ji}^y\right) + d_{ij}^x - \left(l_i^x + l_j^x\right)$ and $\Delta_{ij,y}^a \geq r_{ij}\left(z_{ij}^x + z_{ji}^x\right) + d_{ij}^x - \left(l_i^x + l_j^x\right)$. Adding two inequalities and from (3), we have $\Delta_{ij,x}^a + \Delta_{ij,y}^a \geq r_{ij}\left(z_{ij}^x + z_{ji}^x + z_{ij}^y + z_{ji}^y\right) + d_{ij}^x - \left(l_i^x + l_j^x\right) + d_{ij}^x - \left(l_i^x + l_j^x\right) = r_{ij} + \sum_{w \in W}\left[d_{ij}^w - \left(l_i^w + l_j^w\right)\right]$, the right-hand-side (RHS) of which is equivalent to Definition (5.1.1). In conjunction with minimizing the objective function, $\Delta_{ij,x}^a + \Delta_{ij,y}^a$ will be the deviation for adjacency.

Now, consider department pair $\{i, j\} \in A$ that overlap in direction $w \in W$. WLOG, assume $w = x$, and hence, we have $z_{ij}^x + z_{ji}^x = 0$ and $z_{ij}^y + z_{ji}^y = 1$. Since these

departments cannot overlap in y-direction, the direction-dependent deviation in y as defined in Definition (5.1.2.b) is $d_{ij} - (l_i^y + l_j^y)$. From (5.3) and $z_{ij}^x + z_{ji}^x = 0$, constraint (5.10) is given by $\Delta_{ij,y}^a \geq d_{ij} - (l_i^y + l_j^y)$. Next, consider direction-dependent deviation in x. Since two departments overlap in x, the overlap length can be expressed as $\min\{c_i^x + l_i^x, c_j^x + l_j^x\} - \max\{c_i^x - l_i^x, c_j^x - l_j^x\} > 0$. There are four cases to consider: Case (1) $c_i^x + l_i^x < c_j^x + l_j^x$ and $c_i^x - l_i^x < c_j^x - l_j^x$, Case (2) $c_i^x + l_i^x < c_j^x + l_j^x$ and $c_i^x - l_i^x \geq c_j^x - l_j^x$, Case (3) $c_i^x + l_i^x \geq c_j^x + l_j^x$ and $c_i^x - l_i^x < c_j^x - l_j^x$, and Case (4) $c_i^x + l_i^x \geq c_j^x + l_j^x$ and $c_i^x - l_i^x \geq c_j^x - l_j^x$.

In Case (1), the overlap length is given by $c_i + l_i^x - (c_j - l_j^x)$. Furthermore, adding two inequalities results in $c_j - c_i > 0$, and hence, we have $d_{ij}^x = |c_j - c_i| = c_j - c_i$. Then, the overlap length can be written as $l_i^x + l_j^x - d_{ij}^x$. From (10), (3) and $z_{ij}^x + z_{ji}^x = 0$, we have $\Delta_{ij,x}^a \geq r_{ij}(z_{ij}^y + z_{ji}^y) - [(l_i^x + l_j^x) - d_{ij}^x] = r_{ij} - [(l_i^x + l_j^x) - d_{ij}^x]$, the RHS of which is the direction-dependent deviation in x as defined in Definition (5.1.2.a). In Case (2), the overlap length is given by $c_i + l_i^x - (c_i - l_i^x) = 2l_i^x$. Note that, since $l_i^x \geq lb_i^x$, we have $r_{ij} \leq 2\min\{lb_i^w, lb_j^w\} \leq 2l_i^x$. From Definition (5.1.2.a), the direction-dependent deviation in x-direction is zero since $r_{ij} - 2l_i^x \leq 0$. Since $\Delta_{ij,x}^a$ is a nonnegative variable, it is sufficient to show the RHS of (5.10) is nonpositive. Adding two inequalities after matching the direction of inequalities, we have $c_i^x + c_j^x + l_i^x - l_j^x < c_j^x + c_i^x + l_j^x - l_i^x$, which can be simplified to $l_i^x < l_j^x$. Using this inequality and $d_{ij}^x \geq 0$, the RHS of (5.10) becomes $r_{ij} - [(l_i^x + l_j^x) - d_{ij}^x] = r_{ij} - (l_i^x + l_j^x) + d_{ij}^x < r_{ij} - 2l_i^x \leq 0$.

Case (3) is symmetric to Case (2) and can be proved using the same argument. Similarly, Case (4) is symmetric to Case (1) and the same proof as for Case (1) can be applied.

Adding the two direction-dependent deviations, $\left(\Delta_{ij,x}^{a} + \Delta_{ij,y}^{a}\right)$ results in the deviation for ajoint specification as defined in Definition 5.1 when $w = x$. This completes the proof. ■

5.3.1 Representing proximity/separation specifications using Chebyshev distance

Recall that (5.12) and (5.17) address the proximity and separation specifications using the l_1 norm distance metric, respectively, thus resulting in an octagon shaped region for satisfying either of these requirements. For the situation where a rectangular shaped region is desired by the layout planner, the Chebyshev (or l_∞ norm) distance metric can be applied by modifying the optimization model that was presented earlier. This is accomplished by decomposing the proximity and separation specification deviational variables into x- and y-components (in a similar fashion to the adjacency specification deviational decision variables). Let $\Delta_{ij,w}^{p} \geq 0$ $(i, j \in P, i < j)$ and $\Delta_{ij,w}^{s} \geq 0$ $(i, j \in S, i < j)$ represent the total distance in which the proximity and separation requirements between departments i and j are not satisfied along direction $w \in W$, respectively. Using these decision variables, (5.12) and (5.17) can be replaced with the following constraints:

$$D_{ij}^{w} \leq p_{ij} + \Delta_{ij,w}^{p} \qquad \forall i, j \in P, i < j, w \in W \qquad (5.26)$$

$$D_{ij}^{w} \geq q_{ij}\left(z_{ij}^{w} + z_{ji}^{w}\right) - \Delta_{ij,w}^{s} \qquad \forall i, j \in S, i < j, w \in W. \qquad (5.27)$$

Constraint (5.26) requires departments i and j to have a boundary-to-boundary distance less than or equal to q_{ij} for satisfying the proximity specification along the x- and y-directions to prevent $\Delta_{ij,w}^{p}$ from being positive in value, whereas (5.27) requires i and j to be separated by at least s_{ij} units along the x- or y-direction to avoid $\Delta_{ij,w}^{s}$ being assigned a positive value (reference Figures 5.2 (b) and 5.3 (b) for an illustrative representation of (5.26) and (5.27), respectively). In conjunction with replacing (5.12)

and (5.17) with (5.26)-(5.27), respectively, the weighted sum of deviation variables expressed in (5.1) can be modified as follows:

$$\sum_{u\in\{a,p,s\}}\sum_{i}\sum_{\substack{j>i\\(i,j)\in u}} f_{ij}^{u}\sum_{w\in W}\Delta_{ij,w}^{u} \tag{5.28}$$

Note that (5.1) is almost identical to (5.28) with the exception being that the proximity and separation deviational variables are expressed as x- and y-components in the latter. Updating the base model in consideration of the Chebyshev distance metric yields the following optimization model:

$$\begin{aligned}
&\text{Minimize} && (5.28) && \\
&\text{subject to} && (5.2)\text{-}(5.11),\ (5.13)\text{-}(5.16)\ (5.18),\ (5.21)\text{-}(5.24),\ (5.26)\text{-}(5.27) && \\
& && \Delta_{ij,w}^{u} \geq 0 && \forall u \in \{P,S\}, i,j \in u,\ w \in W. \quad (5.29)
\end{aligned}$$

Depending on the situation, it might be desired by the layout planner to apply multiple distance metrics for a given problem instance, in which the metric for each pair of departments is selected a priori. This can easily be accommodated in the formulation by assigning department pairs to two sets, denoted by L_1 and L_∞, such that the l_1 and Chebyshev distance metrics are applied, respectively, to the elements that they contain. Xie et al. (2018) propose a similar approach in the context of manufacturing systems.

5.3.2 Valid inequalities

A phenomenon that often appears in combinatorial optimization problems is the problem symmetry. The symmetry typically results in longer solution times during the solution process since alternative solutions that yield the same objective function value can be unnecessarily explored (Margot, 2010). One approach for mitigating these effects is to include additional constraints, formally known as valid inequalities, into the

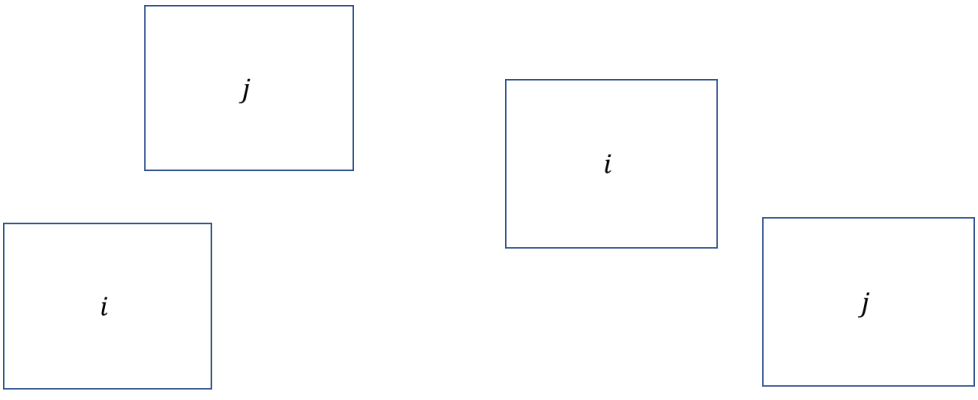
optimization model. In general, an inequality is said to be valid if it is satisfied by all other possible solutions that belong to the feasible region (Cornu´ejols, 2008). The benefit of implementing valid inequalities into a mathematical model allows for the feasible region to shrink in size, or tighten, while ensuring that the original problems constraints are still enforced, thus resulting in potentially reduced computational times. In an effort to reduce the computational burden, three valid inequality constraints that can be applied are introduced.

$$z_{ij}^{w} \leq \sigma_{ij}^{w} \qquad \forall i,j \in I, i < j, w \in W \tag{5.30}$$

$$z_{ji}^{w} + \sigma_{ij}^{w} \leq 1 \qquad \forall i,j \in I, i < j, w \in W \tag{5.31}$$

$$o_{ij}^{w} \leq z_{ij}^{w'} + z_{ji}^{w'} \qquad \forall i,j \in P \cup S, i < j, w \in W, w' \in W \setminus \{w\} \tag{5.32}$$

(5.30) and (5.31) eliminate the symmetry with respect to the department precedence binary decision variables, namely z_{ij}^{w} and σ_{ij}^{w}. (5.32) constrains the value of o_{ij}^{w} with respect to $z_{ij}^{w'}$ and $z_{ji}^{w'}$, where $w' = W \setminus \{w\}$. It is only possible for departments i and j to be overlapping along direction w if either department precedes another along direction w'. The motivation behind this expression is displayed in Figure 5.4.



$z_{ij}^{y} = 1, o_{ij}^{x} = 1$ (a)

$z_{ij}^{x} = 1, o_{ij}^{y} = 1$ (b)

Figure 5.4: Graphical depiction of SBCs

Observe in Figure 5.4 (a) that department i precedes department j along the y-axis, thus implying that $z_{ij}^{y} = 1$. In addition, it can be noticed that the bottom-left corner of department j is overlapping department i along the x-axis. As a result of this second observation, the model will force $o_{ij}^{x} = 1$ because of (5.12)-(5.13) in the proposed model. Since both conditions occurring in conjunction with one another result in $o_{ij}^{x} = 1$, it is then deemed that (5.33) is valid. Figure 5.4 (b) represents the situation where $z_{ij}^{x} = o_{ij}^{y} = 1$. The same logic with respect to Figure 5.4 (a) can be applied for this case.

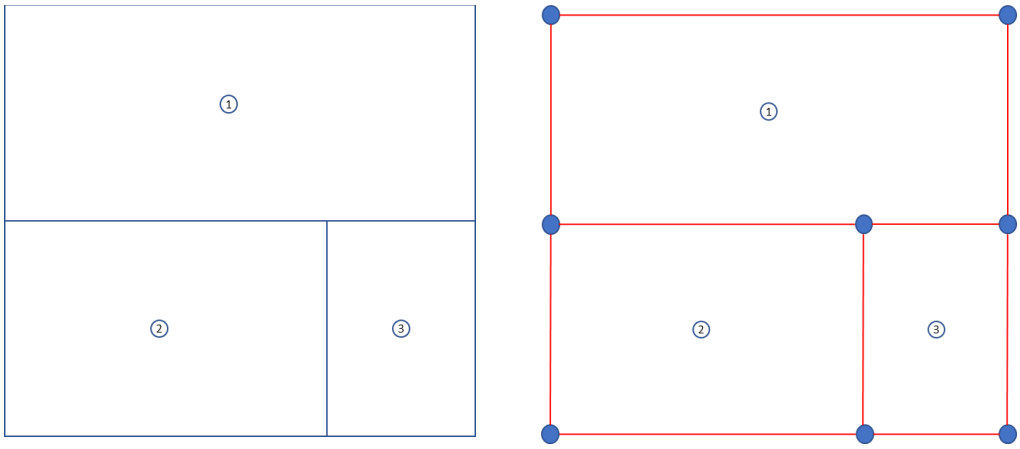
5.4 Phase 2: Aisle Network Generation

In this section, an aisle generation model is proposed for transforming the block layout into a more-detailed representation of the facility. This second-phase model determines the optimal configuration of the aisle network for improving the navigability of occupants. What makes this model distinct from what is currently in the literature is that the facility entrance and exits are represented as decision variables, and their configuration along the facility are optimized with respect to requirements set forth by International Building Code (2021). In addition, the doorway configuration to each department is optimized for reducing the travel distance to the entrance and emergency exits (or primary accessway and secondary accessways, respectively) in the event of an emergency. Following the formulation of the base aisle generation model, additional functional specifications are considered for allowing additional criteria to be addressed during the optimization.

Given a layout design, the boundary of each department is in a rectangular shape with four sides. A side of a department may partially or entirely overlap a side of another

department, or does not overlap at all. For example, in Figure 5.5 (a), the lower side of department 1 partially overlaps the upper side of department 2. It also means the upper side of department 2 entirely overlaps the lower side of department 1. Furthermore, the left, upper and right sides of department 1 do not overlap sides of any other department. Let us define an aisle segment as the smallest overlapping section of a side. If the side is not overlapping, the entire side will be the aisle segment. Then, an aisle structure of a facility layout is configured by selecting a subset of aisle segments. Figure 5.5 (b) displays 10 potential aisle segments (colored in red) that can be used for navigating the facility. For the sake of simplicity, it is assumed that each department has one doorway for ingress and egress, and the doorway can be located along one of the department corners.

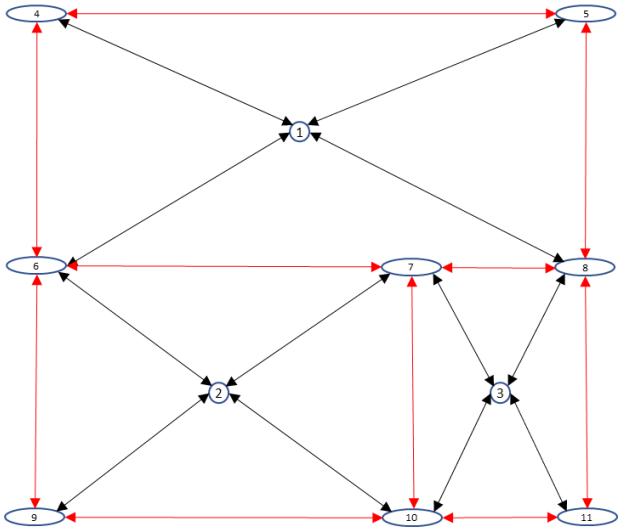
To develop a solution strategy, the layout is first represented as a directed planar graph $G_0(V_0, E_0)$, where V_0 consists of two sets of nodes, V_D and V_C, that represent the center points of departments and the corner points of departments, respectively. Nodes in V_D are the origins and destinations of flows on the graph, and nodes in V_C represent potential doorways of departments. Alexander et al. (1977) claim that placing the doorway away from the corners negatively impacts the occupant flow within in, as well its respective space utilization. E_0 consists of edges that are aligned with aisle segments, plus those joining the node representing a department and its potential doorways. Associated with each edge is the length that corresponds to the actual length of the portion of that aisle segment. The length of the edges joining V_D and V_C (i.e., the distance from a department to its doorway) is assumed to be zero. Figure 5.6 displays the resulting graph for the layout in Figure 5.5 (a).



(a)
layout for three department case

(b)
corresponding graph G_0

Figure 5.5: Potential aisle segments using department boundaries



$$V_D = \{1, 2, 3\}$$
$$V_C = \{4, 5, 6, 7, 8, 9, 10, 11\}$$

Figure 5.6: Illustration of transformation to a graph

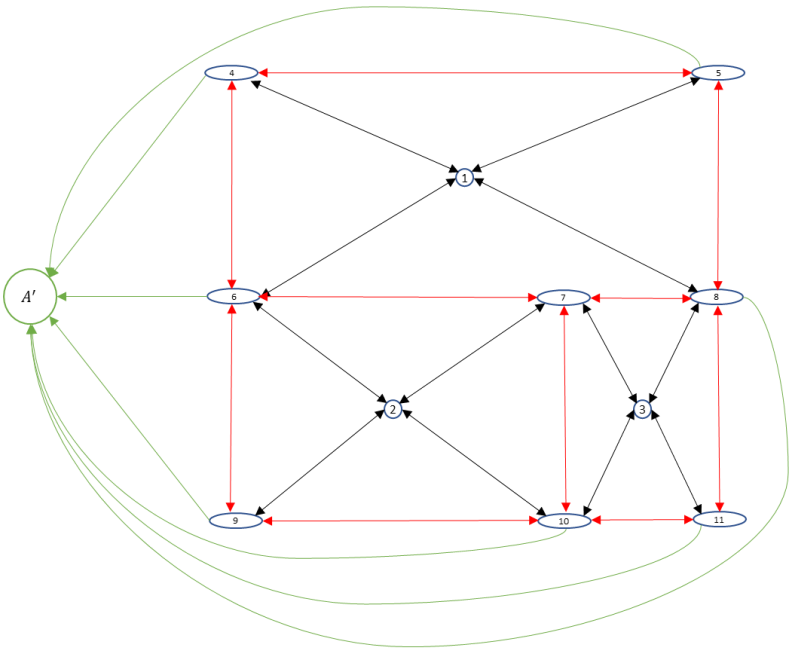
This representation of G_0 is similar to what Lee et al. (2009) proposed for minimizing the material flow cost when a block layout of the facility is given. The corner points of departments are labeled as candidate locations for the doorways in their research.

However, they also allow the doorways to be located at the intersection points corresponding to departments that are touching because it is proven that these locations are optimal in this problem setting (Kiran and Tansel, 1989; Tanchoco and Sinreich, 1992; Kim and Kim; 1999). However, they did not address the configuration of accessways for the facility, which is a crucial component during the pre-design phase of the facility because of the relevant building codes that must be satisfied (International Building Code, 2021). It is worth mentioning that this representation of G_0 is smaller in size compared to the approaches from Peng et al. (2016) and Li and Hua (2019), where it is necessary to transform the entire facility blueprint into a mesh consisting of unit-sized cells that overlap the departments.

One requirement enforces the accessways to be separated by at least one-half (without sprinkler system installed) or one-third (with sprinkler system installed) of the length of the maximum overall diagonal dimension of the facility (denoted by L). When there are three or more accessways, then it is required for at least two of them to be separated by the minimum distance threshold, and all other pairs of accessways must be separated by a reasonable distance. Additional requirements address other functional specifications such as the maximum allowable travel distance from the departments to the exit accessways, the number of required exits based on occupancy, etc. Gao et al. (2020) consider an evacuation problem where the configuration of department doorways and facility accessways are optimized for improving the facility evacuation protocol. The difference between their work and this research is that the facility layout and its corresponding aisle structure are known *a priori* in the former, whereas only the block layout of the facility is known in the latter.

Hence, the objective function is modified not only to accommodate the doorway configuration of each department, but also to address the configuration of the primary and secondary accessways when designing the aisle network. The former serves as a primary entrance/exit when entering/leaving the facility under normal conditions, whereas the latter are strictly used for evacuating the facility during an emergency event (also known as an exit accessway). Note that the primary accessway(s) can also be classified as an exit accessway during an emergency event since occupants are most familiar with its location and might have a preference to evacuate from that point.

To effectively handle primary and secondary accessways in the proposed model, $G_0(V_0, E_0)$ is transformed into $G(V, E)$ as follows. Let A_P and A_S denote the index sets of primary and secondary accessways, respectively. Also, let B denote the set of nodes that are along the boundary of the facility (e.g., nodes 4, 5, 6, 8, 9, 10, and 11 in Figure 5.6). From this, each element from $A' = A_P \cup A_S$ is inserted into G (i.e., $V = V_0 \cup A'$), and an edge is drawn from elements in B to elements in A'. Hence, newly added edges constitute the cut-set of a cut (V_0, A'). These newly introduced nodes in A' will serve as sink nodes when finding a path from each department to the accessways. The rationale behind this expansion of G_0 to G is to map the accessways to a node along the facility boundary, where each boundary node can accommodate at most one accessway. Figure 5.7 displays a graphical representation of this transformation to G, where $A' = A_P \cup A_S$ (circled in green) and the green arrows are the edges that connect nodes from B to A'. Although only one green arrow from each node in B to the set A' is displayed in Figure 5.7 to avoid cluttered visualization, assume that all elements in A' have an incoming edge from all elements in B.



$V_{Department} = \{1, 2, 3\}$
$V_{Corner} = \{4, 5, 6, 7, 8, 9, 10, 11\}$
$B = \{4, 5, 6, 8, 9, 10, 11\}$

Figure 5.7: Illustration when adding accessway nodes to the facility graph representation

The primary concern in the second phase is the travel distance between each department to (both primary and secondary) accessways, which needs to be kept minimal. Furthermore, note that, when an aisle is included in the final layout design, this in turn reduces the amount of available space that can be allocated towards the departments. This can potentially impact their respective functionalities since less space would be made available for them, and hence, it is desired to have minimally spanned aisles. From this observation, the objective of the second phase problem is to carefully select edges in E to generate aisle segments for minimizing the travel distance between departments and the accessways, while simultaneously minimizing the total length of the aisle network, thus resulting in a multi-objective optimization problem as presented below.

Let an ordered pair (m, n) denote the directed edge from node m to node n. Furthermore, let E_{adj} denote the set of undirected edges that lie along the boundary between department pairs that are desired to be adjacent as in Phase 1. The following notation of the problem parameters is used to present the model:

$D_{(a,b)}$: Euclidean distance between node $a \in B$ and $b \in B$

$d_{(m,n)}$: length of edge $(m, n) \in E$

$w_1/w_2/w_3$: objective function weight associated with minimizing the distance between departments and the primary accessway / distance between departments and the secondary accessway(s)/total length of the aisle network.

α_i: threshold travel distance from the doorway of department i to any accessway.

β: minimum separation distance that must be satisfied by at least one pair of accessways as specified by the International Building Code (2021).

γ: reasonable separation distance that must be satisfied between the remaining accessways.

$M \gg 1$: a large constant.

Let $FS(i) = \{b \in V: (i, b) \in E\}$ and $RS(i) = \{a \in V: (a, i) \in E\}$ represent the forward- and backward-star of node i, respectively. The following decision variables are then used for the optimization model:

$f_{ik}^{(m,n)}$: network flow from department $i \in V_D$ and to accessway $k \in A'$ through edge (m, n).

x_i^b: variable equal to 1 if the doorway of department $i \in V_D$ is placed at node $b \in FS(i)$. Otherwise, 0.

y_k^b: binary variable equal to 1 if accessway $k \in A'$ is placed at node $b \in B$. Otherwise, 0.

$z_{(m,n)}$: binary variable equal to 1 if edge $(m, n) \in E$ or $(m, n) \in E$ is used as an aisle in the layout ($m < n$). Otherwise, 0.

$q_{(k,l)}$: binary variable equal to 1 if accessways $k \in A'$ and $l \in A'$ are separated by at least β units. Otherwise, 0, implying they are separated by at least γ units of distance ($k < l$).

Using this notation, the base MILP model can be formulated as follows:

$$\text{Minimize} \quad w_1 \sum_{i \in V_D} \sum_{k \in A_P} \sum_{(m,n) \in E} d_{(m,n)} f_{ik}^{(m,n)} + w_2 \sum_{i \in V_D} \sum_{k \in A_S} \sum_{(m,n) \in E} d_{(m,n)} f_{ik}^{(m,n)} + w_3 \sum_{\substack{(m,n) \in E \\ m<n}} d_{(m,n)} z_{(m,n)} \tag{5.33}$$

subject to

$$\sum_{b \in FS(i)} x_i^b = 1 \qquad \forall i \in V_D \tag{5.34}$$

$$\sum_{b \in B} y_k^b = 1 \qquad \forall k \in A' \tag{5.35}$$

$$\sum_{k \in A_P \cup A_S} y_k^b \leq 1 \qquad \forall b \in B \tag{5.36}$$

$$f_{ik}^{(i,b)} = x_i^b \qquad \forall i \in V_D, k \in A', b \in FS(i) \tag{5.37}$$

$$f_{ik}^{(b,k)} = y_k^b \qquad \forall i \in V_D, k \in A', b \in B \tag{5.38}$$

$$\sum_{m \in RS(n)} f_{ik}^{(m,n)} = \sum_{o \in FS(n)} f_{ik}^{(n,o)} \qquad \forall i \in V_D, k \in A', v \in V \setminus (V_D \cup A') \tag{5.39}$$

$$\sum_{i \in V_D} \sum_{k \in A_P \cup A_S} \left(f_{ik}^{(m,n)} + f_{ik}^{(n,m)} \right) \leq M z_{(m,n)} \qquad \forall (m,n) \in E, m < n \tag{5.40}$$

$$\sum_{(m,n) \in E} d_{(m,n)} f_{ik}^{(m,n)} \leq \alpha_i \qquad \forall i \in V_D, k \in A' \tag{5.41}$$

$$\sum_{k \in A_P \cup A_S} \sum_{\substack{l \in A_P \cup A_S \\ k<l}} q_{(k,l)} = 1 \tag{5.42}$$

$$\sum_{b\in B}\sum_{\substack{c\in B\\ b\neq c\\ (b,c):D_{(b,c)}\geq\gamma}} D_{(b,c)}y_k^b\, y_l^c \geq \beta q_{(k,l)} + \gamma(1-q_{(k,l)}) \qquad \forall k,l \in A', k<l \qquad (5.43)$$

$$\sum_{(m,n)\in E_{adj}} z_{(m,n)} = 0 \qquad (5.44)$$

$$f_{ik}^{(m,n)} \geq 0 \qquad \forall i \in V_D, k \in A', (m,n) \in E \qquad (5.45)$$

$$x_i^b \geq 0 \qquad \forall i \in V_D, b \in FS(i) \qquad (5.46)$$

$$y_k^b \in \{0,1\} \qquad \forall\, k \in A', b \in B \qquad (5.47)$$

$$q_{(k,l)} \in \{0,1\} \qquad k,l \in A', k \neq l \qquad (5.48)$$

$$z_{(m,n)} \in \{0,1\} \qquad (m,n) \in E, m<n. \qquad (5.49)$$

Objective function (5.33) minimizes the distance between each department and all accessways, and the total length of the aisle network. The weights can be adjusted based on design preferences, but for the numerical study in this research, it is assumed that $w_1 \gg w_2$, i.e., assigning a higher priority to the distance that occupants regularly travel through the primary accessway under normal circumstances, whereas the secondary accessways are only used during an emergency event. (5.36) assigns the doorways of each department to one of its corner points. (5.35) assigns the primary/secondary accessway to one of the boundary nodes in B. (5.36) prevents more than one accessway from being assigned to a boundary node. (5.37)-(5.38) represent the origin and destination flow balance constraints for department i and accessway k, respectively. (5.39) represents the intermediary flow balance constraint.

(5.40) forces $z_{(m,n)} = 1$ if edge (m,n) is navigated along for any path between the departments and accessways. This constraint ties $z_{(m,n)}$ to the objective function such that every occurrence of $z_{(m,n)} = 1$ increases the objective function value by its

associated length, namely $d_{(m,n)}$. (5.41) requires the travel distance between departments and accessways to be no more than α_i units. Note that the threshold travel distance to the accessways is α units (constant). However, α needs to be adjusted with respect to the hypotenuse of each department (also constant) to account for the worst-case scenario of occupants traveling to the department doorway from the opposite side and corner of the room. This gives us $\alpha_i = \alpha - h(i)$, where $h(\cdot)$ is a function for calculating the hypotenuse of department i.

(5.42)-(5.43) jointly determine which pair of accessways should be separated by the minimum threshold distance as specified by the International Building Code. This constraint is replicated for all pairs of accessways, namely $\forall k, l \in A', k < l$, where the value that is assigned to $q_{(k,l)}$ dictates how far apart the accessways are with respect to their respective Euclidean distance. The product of all pairs of y_k^b and y_l^c is necessary on the left-hand side for guaranteeing that accessways k and l are separated by β units if $q_{(k,l)} = 1$ (or γ units if $q_{(k,l)} = 0$) when $y_k^b = y_k^c = 1$. (5.44) forces all edges in E_{adj} to not be converted to aisles so that direct doorways between two departments with the adjacency requirement can be built on those aisles. (5.45)-(5.49) are used for defining the decision variables in the model.

5.4.1 Extensions to the base aisle generation model

As addressed by Peng et al. (2016) and Li and Hua (2019), there are a variety of functional specifications that can be incorporated into the aisle generation model for improving the navigability of occupants. Some of these specifications include minimizing the number of turning points to reduce the level of confusion of occupants during an emergency evacuation (*zig-zag avoidance*), removing all dead-ends that may occur along

any corridor in the aisle network (*dead-end avoidance*), and minimizing the travel distance between particular points of interest other than the sink nodes (*point-to-point constraint*).

The first two of these specifications fall under the context of wayfinding, where it is of interest to generate simpler paths to make it easier for occupants to navigate the facility. Arthur and Passini (1992) suggest that aisle networks with fewer zig-zags reduce the amount of stress and anxiety that occupants might have. Typically, dead ends are prohibited within the aisle network because it can make the evacuation time of occupants longer and put them at severe risk if they make an incorrect turn. Incorporating the point-to-point constraint in the aisle generation model will allow for more efficient navigation between various departments in the facility, rather than strictly generating the aisle network in consideration of primary and secondary accessways. Recall from Section 5.3 that P contains department pairs that are desired to be within proximity in the layout design. From this, point-to-point constraints can be applied not only to minimize the boundary-to-boundary distance between proximity desired departments using the model from Section 5.3, but to also minimize the travel distance between these corresponding department pairs. In addition, department pairs with an unsatisfied adjacency requirement in the layout design can be included in the set of departments for point-to-point constraints to keep them in the vicinity.

5.4.1.1 Zig-zag avoidance

For minimizing the number of zig-zag occurrences in the aisle network, it is necessary to find all pairs of horizontal-vertical segments in G prior to the optimization. Note that a zig-zag is only formed when a flow occurs in the network that transitions

from a horizontal (or vertical) segment to a vertical (or horizontal) segment. These pairs of horizontal-vertical aisle segments can be stored in set Φ. Without loss of generality, suppose the horizontal and vertical segments are defined along edges (l, m) and (n, o), respectively. The direction of the flow along these two segments is irrelevant in this situation. From this, let $u^{(l,m),(n,o)}$ be a binary variable equal to 1 if both segments are used in the aisle network, namely $z_{(l,m)} = z_{(n,o)} = 1$. Introducing $u^{(l,m),(n,o)}$ and expressing it as a term in the objective function with positive weight w_4 then allows for the number of zig-zag occurrences to be reduced when also enforcing the following constraint.

$$z_{(l,m)} + z_{(n,o)} \leq 1 + u^{(l,m),(n,o)} \qquad \forall\{(l,m),(n,o)\} \in \Phi. \qquad (5.50)$$

Constraint (5.50) prevents $z_{(l,m)}$ and $z_{(n,o)}$ from both being equal to 1 simultaneously when $u^{(l,m),(n,o)} = 0$. On the other hand, when $u^{(l,m),(n,o)} = 1$, no restriction is imposed on the sum of $z_{(l,m)}$ and $z_{(n,o)}$, at the consequence of introducing an undesired zig-zag junction in the aisle network. Although the decision variable indices are represented using different notation in (5.50), it is required for at least two of them to be equal based on the construction of G. For example, in Figure 5.6, observe the zig-zag formed at node 7 from edges (7,8) and (7,10). This results in $z_{(7,8)} + z_{(7,10)} \leq 1 + u^{(7,10),(7,8)}$ to be included as a constraint in the model (where $l = n$).

5.4.1.2 Dead-end avoidance

A series of inequality constraints can be introduced to guarantee there are no abrupt dead-ends within the aisle network. However, dead-end avoidance is only necessary when an aisle does not have an accessway at the end of it. Since the location of

the accessways are variable and can only be placed on the facility boundary, it is necessary to consider two-cases for dead-end avoidance. The first case addresses edges that are within the interior of the facility, where neither of the nodes along these edges are in B. The second case addresses the opposite situation, namely edges where at least one of the nodes are in B. The former is the most straightforward, whereas for the latter, dead-end avoidance constraints become redundant if and only a boundary node along the edge is used as an accessway for the facility, guaranteeing that occupants can safely exit when traveling along this aisle. To formulate the dead-end avoidance constraints, let $z'_{(m,n)}$, $\forall (m,n) \in E$, be a binary variable equal to 1 if the flow along edge (m,n) is from node m to n. Otherwise, if $z'_{(m,n)} = 0$ and $z_{(m,n)} = 1$, this implies that $z'_{(n,m)} = 1$, indicating that the flow along edge (m,n) is from node n to m. Using this newly defined variable, the dead-end avoidance constraints can now be expressed.

$$z'_{(m,n)} + z'_{(n,m)} = z_{(m,n)} \qquad \forall (m,n) \in E, m < n \qquad (5.51)$$

$$z'_{(m,n)} \leq \sum_{o \in FS(n)} z'_{(n,o)} \qquad \forall (m,n) \in E, n \notin B \qquad (5.52)$$

$$z'_{(m,n)} \leq \sum_{o \in RS(n)} z'_{(o,m)} \qquad \forall (m,n) \in E, n \notin B \qquad (5.53)$$

$$z'_{(m,n)} \leq \sum_{o \in FS(n)} z'_{(n,o)} + \sum_{k \in A_P \cup A_S} y_k^n \qquad \forall (m,n) \in E, n \in B \qquad (5.54)$$

$$z'_{(m,n)} \leq \sum_{o \in RS(n)} z'_{(o,m)} + \sum_{k \in A_P \cup A_S} y_k^m \qquad \forall (m,n) \in E, n \in B. \qquad (5.55)$$

Constraint (5.51) selects a direction for the flow along edge (m,n). (5.52)-(5.53) guarantees a continuous flow within the aisle network for all edges that do not have a node along the facility boundary. (5.54)-(5.55) are similar to (5.52)-(5.53), with the exception that the sum of all y_k^n decision variables makes the dead-end avoidance

unnecessary when node n is selected for accessway assignment. Note that when $\sum_{k\in A_P\cup A_S} y_k^n = 0$, (5.54)-(5.55) are reduced to (5.52)-(5.53). It is worth mentioning that $z'_{(m,n)}$ is strictly used for avoiding dead-ends in the aisle network, and does not dictate the flow of occupants within the aisle network. For example, $z'_{(n,m)} = 1$ does not imply that $f_{ik}^{(m,n)} = 0$ and $f_{ik}^{(n,m)} = 1$. If unidirectional flow is desired along the aisle segments, then another set of constraints can be formulated to do so using $z'_{(n,m)}$.

5.4.1.3 Point-to-point constraints

Minimizing the travel distance between proximity desired department pairs can be incorporated into the aisle generation model by introducing a series of constraints similar to (5.37)-(5.40). In addition, department pairs with an unsatisfied adjacency requirement can be included in the set of departments for point-to-point constraints to be applied towards. Let A_{uns} denote the set of department pairs with an unsatisfied adjacency requirement. These department pairs can be determined by referencing the $\Delta_{ij,x}^a$ and $\Delta_{ij,y}^a$ decision variables from the model in Section 5.3, and checking to see if $\Delta_{ij,x}^a + \Delta_{ij,y}^a > 0$. Also, let $g_{(m,n)}^{ij}$ be a continuous variable equal to 1 if (m,n) is on the shortest path between department $i \in V_D$ and department $j \in V_D$ $(i < j,\ i,j \in A_{uns} \cup P)$. From this definition, the following constraints can be used for minimizing the distance between proximity desired departments.

$$g_{ij}^{(i,b)} = x_i^b \qquad \forall i,j \in V_D, i<j, i,j \in A_{uns}\cup P,\ b \in FS(i) \tag{5.56}$$

$$g_{ij}^{(b,j)} = x_j^b \qquad \forall i,j \in V_D, i<j, i,j \in A_{uns}\cup P, b \in B \tag{5.57}$$

$$\sum_{m\in RS(n)} g_{ij}^{(m,n)} = \sum_{o\in FS(n)} g_{ij}^{(n,o)} \qquad \forall i,j \in V_D, i<j, i,j \in A_{uns}\cup P,\ v \in V \setminus (V_D \cup A') \tag{5.58}$$

$$\sum_{i \in V_D} \sum_{\substack{j \in V_D \\ j > i \\ i,j \in A_{uns} \cup P}} \left(g_{ij}^{(m,n)} + g_{ij}^{(n,m)} \right) \leq M z_{(m,n)} \qquad \forall (m,n) \in E, m < n. \tag{5.59}$$

5.4.2 Modified aisle generation model

The final aisle generation model in consideration of all functional specifications takes on the following form.

$$\text{Minimize} \quad (35) + w_4 \sum_{(l,m),(m,n) \in \Phi} u^{(l,m),(m,n)} + w_5 \sum_{i \in V_D} \sum_{\substack{j \in V_D \\ j > i \\ i,j \in A_{uns} \cup P}} \sum_{(m,n) \in E} d_{(m,n)} g_{ij}^{(m,n)} \tag{5.60}$$

subject to (36)-(58)

$$g_{ij}^{(m,n)} \geq 0 \qquad \forall i,j \in V_D, i < j, i,j \in A_{uns} \cup P \tag{5.61}$$

$$u^{(l,m),(n,o)} \in \{0,1\} \qquad \forall (l,m),(n,o) \in \Phi \tag{5.62}$$

$$z'_{(m,n)} \in \{0,1\} \qquad \forall (m,n) \in E. \tag{5.63}$$

Incorporating the zig-zag avoidance and point-to-point distances in the objective function makes the base aisle generation model more complex in the sense that an optimal solution needs to be found for a larger multi-objective problem with conflicting criteria.

5.4.3 Consideration of facility resilience during pandemic events

The recent global pandemic had a negative impact on business operations across a wide span of industries. Not only were financial losses incurred and a large number of people lost their jobs because of the pandemic, but many people were also exposed to the virus and became infected as a result. According to the NC Department of Health and Human Services (NC DHHS), the manufacturing and meat and poultry processing sectors contributed to the majority of infections for facilities belonging to the Workplace category in North Carolina (NC DHHS, 2021). Examples of other facility types that contributed to many infections include K-12 schools, colleges/universities, and child-care.

It is likely that people were infected from these facility types since infection prevention was not addressed during the layout design process, thus placing occupants at a higher risk of infection when navigating the facility during a pandemic. As addressed in Chapter 4, social distancing is an essential design feature for reducing the infection spread in the context of modifying the layout design of an interior space. In addition, it might also be of interest to address the flow of occupants and adjust how people navigate the facility by imposing restricted walking paths to reduce the overall infection spread. One approach to accommodate this is to enforce unidirectional flow along the aisle network. Several studies have shown that enforcing unidirectional flow was effective in reducing the spread of infection between healthcare workers (Lenaghan and Schwedhelm, 2015; Zimring et al., 2018; Wong, 2019). The CDC also recommends unidirectional flow along walking paths as an administrative control for promoting social distancing (CDC, 2019). Thus, designing the facility with an aisle structure in mind that allows for unidirectional flow to quickly be enforced can be advantageous to keep occupants healthy and resume facility operations under pandemic conditions. Incorporating this functional specification during the pre-design phase will likely incur additional costs when generating the aisle network because it is necessary to guarantee that a path between all departments exists with unidirectional flow enforced, thus requiring one or more cycles to be formed in the network for guaranteeing feasibility. At the same time, this allows for facility operations to carry on at a reduced capacity during a pandemic event with safety measures enforced that are effective in reducing the spread of infection.

The symmetry which exists for department-to-department navigation (as shown in (59)) no longer holds when enforcing unidirectional flow in the aisle network (in other words, the path from department i to j is not the same as the path from department j to i). As a result, it is necessary to apply asymmetric department indices so that these distinct paths can be determined. Department pairs that have a proximity, separation, or an unsatisfied adjacency requirement are considered since adjacent departments already have a direct passageway between them, making it unnecessary to construct a path between them along the aisle network. Let $h_{ij}^{(m,n)}$ be a continuous variable equal to 1 if the path from department i to j traverses from node m to n, and 0 otherwise ($i, j \in V_D, i \neq j, i, j \in A_{uns} \cup P \cup S, (m,n) \in E$). Constraints are shown below using binary variable $z'_{(m,n)}$ from Section 5.4.1.2 for determining the aisle direction.

$$h_{ij}^{(i,b)} = x_i^b \qquad \forall i, j \in V_D, i \neq j, i, j \in A_{uns} \cup P \cup S, \; b \in FS(i) \tag{5.64}$$

$$h_{ij}^{(b,j)} = x_j^b \qquad \forall i, j \in V_D, i \neq j, i, j \in A_{uns} \cup P \cup S, \; b \in B \tag{5.65}$$

$$\sum_{m \in RS(n)} h_{ij}^{(m,n)} = \sum_{o \in FS(n)} h_{ij}^{(n,o)} \qquad \forall i, j \in V_D, i \neq j, i, j \in A_{uns} \cup P \cup S, \; v \in V \setminus (V_D \cup A') \tag{5.66}$$

$$\sum_{i \in V_D} \sum_{\substack{j \in V_D \\ j \neq i \\ i,j \in P \cup S}} h_{ij}^{(m,n)} \leq M z'_{(m,n)} \qquad \forall (m,n) \in E \tag{5.67}$$

$$y_k^m \leq \sum_{n \in FS(m)} z'_{(m,n)} \qquad \forall k \in A_P, m \in B \tag{5.68}$$

$$y_k^n \leq \sum_{n \in RS(m)} z'_{(n,m)} \qquad \forall k \in A_S, m \in B. \tag{5.69}$$

Constraint (5.64)-(5.66) are similar to (5.56)-(5.58) with the exception that asymmetric indices are applied for i and j. (5.67) determines the aisle direction if there is at least one

path that navigates it from node m to n. (5.68)-(5.69) control the aisle direction for primary and secondary accessways, respectively, where occupants enter the facility using primary accessways, and exit the facility using secondary accessways. The motivation behind this is to have dedicated entry/exit points into the facility, which is a practice that was commonly adapted during the pandemic (Tesla, 2020; Lear, 2020). Note that (5.51) is applied in conjunction with (5.64)-(5.69) in a similar fashion as the dead-end avoidance functional specification. It is worth mentioning that (5.51)-(5.55) and (5.64)-(5.69) can be applied simultaneously if unidirectional flow is desired since it is guaranteed that no dead-ends will be present in the aisle network. Only the department pairs with a proximity or unsatisfied adjacency requirement are included in the objective function to reduce their respective travel lengths. From this, the extended aisle generation model with unidirectional flow requirements is as follows.

$$\text{Minimize} \quad (60) + w_6 \sum_{i \in V_D} \sum_{\substack{j \in V_D \\ j \neq i \\ i,j \in A_{uns} \cup P}} \sum_{(m,n) \in E} d_{(m,n)} h_{ij}^{(m,n)} \tag{5.70}$$

$$\text{subject to} \quad (36)\text{-}(68)$$

$$h_{ij}^{(m,n)} \geq 0 \qquad \forall i, j \in V_D, i \neq j, i, j \in A_{uns} \cup P \cup S, (m,n) \in E. \tag{5.71}$$

5.5 Computational Study

In this section, the proposed framework is tested using two case studies (one small-scale and one large-scale) to demonstrate how an optimal arrangement of departments and configuration of aisles can be generated. The small-scale and large-scale case studies consist of seven and 27 departments, respectively, both of which are associated with educational facilities. The intention of the small-scale case study is to showcase how the layout design is subject to change under different configurations of the

functional specifications when applying the two-phase method. In addition, it is also of interest to gather feedback from professional architects using output from the small-scale problem to gauge the practicality of the approach.

On the other hand, the large-scale case study considers a more realistic project that an architect might take on and how they can apply the proposed framework for generating alternative layout design options. It was observed during these experiments that enforcing an excessive number of adjacency requirements amongst homogeneously sized departments with similar purposes results in block layouts containing departmental clusters scattered across the facility, thus reducing the overall attractiveness of the design. In an effort to improve the quality of the solutions that are generated under these conditions, it is suggested to cluster the similarly sized departments together to transform them into a series of larger-sized departments (as a pre-processing technique), and then proceed with optimizing the block layout in consideration of the adjacency specifications using the modified input data (output of the first phase model). Not only does this reduce the impact of the phenomenon as mentioned before, but it also improves the overall computational tractability since the number of departments in the problem instance is reduced by clustering subsets of them together. More details regarding this are provided later in this section. In addition, a discussion on calibrating the objective weights and functional specifications for the second phase model is presented.

The result of the small-scale case study will be presented first, which is then followed by the outcome of the large-scale case study afterwards. The proposed framework is implemented using Python (3.7.3, 2018) and solved by a commercial optimization solver, Gurobi (8.1.1, 2018), which is run in a server equipped with four

processors of Intel Xeon CPU E7-8890 v3 (3.60 GHz) and an OS of Windows Server 2012 R2. A 1-hour time limit is imposed in case an optimal solution could not be found in a reasonable amount of time. It is assumed that all departments have a sufficiently small occupancy rate in which only one doorway is required. The second phase model can easily be extended to accommodate two doorways for departments with large occupancy rates. However, the corresponding mathematical formulation is out of scope for this research.

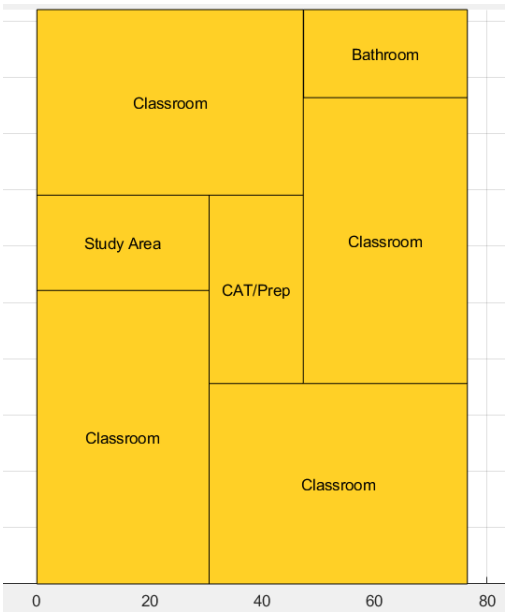
5.5.1 Small-scale case study

To demonstrate how the two-phase method can produce an optimal layout design alternative, a small-scale problem resembling a floor inside of an educational facility (76.5' x 102') was generated consisting of seven departments, namely four classrooms (1,634 sq. ft; 1,560 sq. ft.; 1,596 sq. ft.; 1,480 sq. ft.), one bathroom (455 sq. feet), one computer lab (560 sq. ft; referred to as CAT/Prep), and a study area (518 sq. ft.). An aspect ratio of 2.25 was applied for all departments in the small-scale problem. The adjacency specifications for this problem instance are presented in Table 5.1.

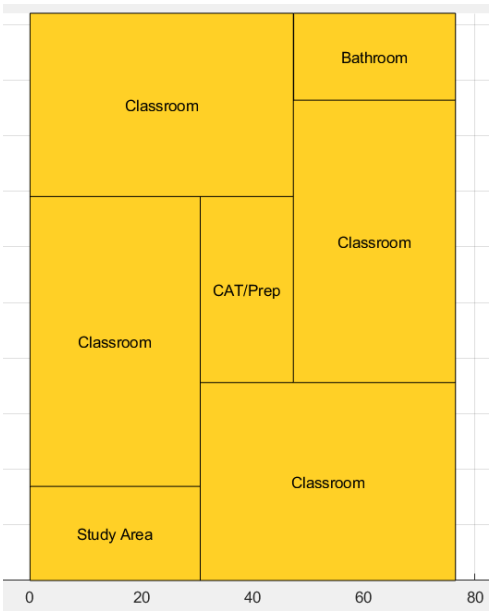
Table 5.1:

	Classroom	Bathroom	CAT/Prep	Study Area
Classroom	-	Separate	Proximity	-
Bathroom	-	-	Separate	Separate
CAT/Prep	-	-	-	-
Study Area	-	-	-	-

It is assumed that all adjacency specifications are of equal importance in this problem instance, so all corresponding values of f_{ij}^{p} and f_{ij}^{s} are set equal to 1. In addition, p_{ij} and q_{ij} are set equal to 10 ft. for defining the threshold distances in which the proximity and separation requirements are satisfied, respectively. The l_1- and l_∞-norm distance metrics are applied for producing block layouts (separately) and are shown below in Figure 5.8.



l_1-norm distance
Obj. value: 19.931
Solution time: 741.89 secs.
(a)



l_∞-norm distance
Obj. value: 19.931
Solution time: 242.12 secs.
(b)

Figure 5.8: Block layouts for small-scale problem

Notice that applying either distance metric for enforcing the proximity and separation requirements results in block layouts that are almost identical, with the exception being the location of the study area. Observe how the study area is in the center of the layout in Figure 5.8 (a), whereas it is located at the bottom-left corner in Figure 5.8 (b), which is on the opposite side of where the bathroom is located at the top-right corner. The block layout in Figure 5.8 (a) can be retrieved in (b) by swapping the location of the study area and the classroom that is directly above it (a similar operation can be performed to retrieve the layout in Figure 5.8 (b) from (a)). An optimal solution was found for both cases when applying the l_1- and l_∞-norm (741.89 and 242.12 seconds, respectively). Note that the objective function values for both cases are positive valued, indicating that at least one of the adjacency specifications are violated in the layout. It can be observed that the separation of the study area and bathroom is more severe in Figure

5.8 (b) compared to 5.8 (a). In addition, the objective values are identical in both layouts, indicating that alternative layouts exist in which the adjacency specifications are satisfied.

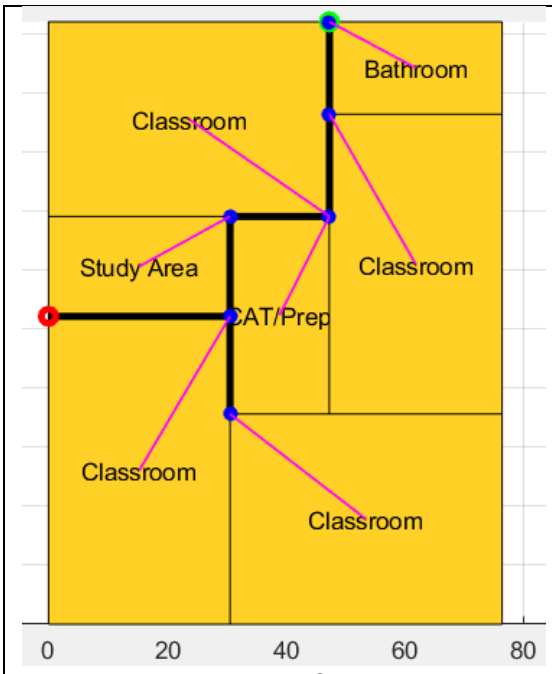
Referring to the adjacency specifications in Table 5.1 and the block layouts in Figure 5.8, it appears that the proximity requirement between all classrooms and the CAT/Prep room are satisfied since they are placed directly beside each other. The separation requirements between the bathroom and CAT/Prep room (as well as the study area) also appear to be satisfied at a quick glance (these observations can be verified directly using the optimal values of the proximity and separation deviational variables). Separation between the bathroom and study area is also apparent in both layouts with respect to the distance metric that is applied during the optimization. The only requirement that remains unchecked is between the classrooms and the bathroom, and it can be verified visually that this requirement is unsatisfied for two of the classrooms that are located beside the bathroom in both Figure 5.8 (a) and (b). Since an optimal solution was found for this small-scale problem, the model output suggests that it is necessary to violate the separation requirement between two of the classrooms and the bathroom to maximize the most adjacency specifications in the layout (assuming all adjacency specifications have equal preference).

Aisle networks were then constructed using the second phase model following the generation of block layouts from the first phase. Since the layouts in Figure 5.8 are similar to one another, it was decided to apply the second phase model on the block layout generated in Figure 5.8 (a). For this exercise, one primary and one secondary accessway were considered for placement along the perimeter of the facility, and the minimum separation distance between them was manipulated to see how the aisle

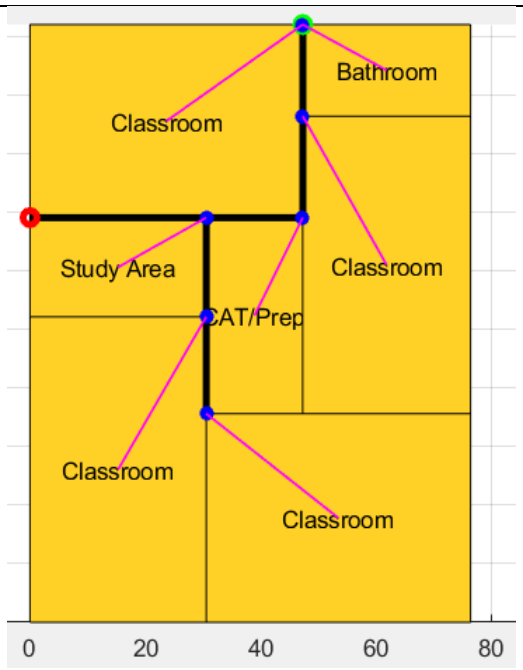
network changes as a result of doing so. The maximum diagonal length of the facility L is used for enforcing the minimum separation distance between accessways depending on whether the building has an automatic sprinkler system ($L/3$) or not ($L/2$). Zig-zag avoidance, dead-end avoidance, and point-to-point constraints were also incorporated for modifying the aisle network, resulting in 10 combinations of the functional specifications (shown in Table 5.2). Six distinct aisle networks were generated after executing these test cases (TCs), and are displayed in Figure 5.9. Note that the thick black lines indicate an activated aisle segment, the magenta lines from the department centers represent which corner is used for the doorway, the green circle along the boundary represents the primary accessway, and the red circles along the boundary represent the secondary accessway. The values assigned to $w_1, w_2, w_3, w_4, w_5, w_6$ for this exercise are 10, 5, 20, 10, 5, and 1, respectively.

Table 5.2: TCs for the small-scale problem

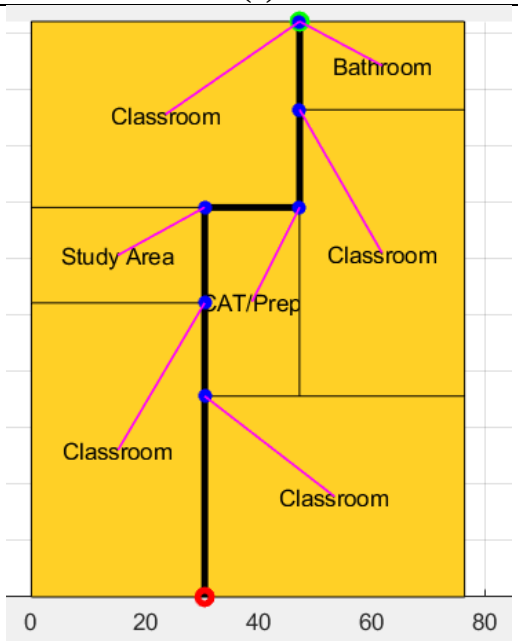
TC	Distance between exits	Zig-zag avoidance	Dead-end avoidance	Point-to-point	Unidirectional flow
1	$L/2$				
2	$L/3$				
3	$L/2$	X	X		
4	$L/3$	X	X		
5	$L/2$		X	X	
6	$L/3$		X	X	
7	$L/2$		X		X
8	$L/3$		X		X
9	$L/2$	X	X	X	
10	$L/3$	X	X	X	



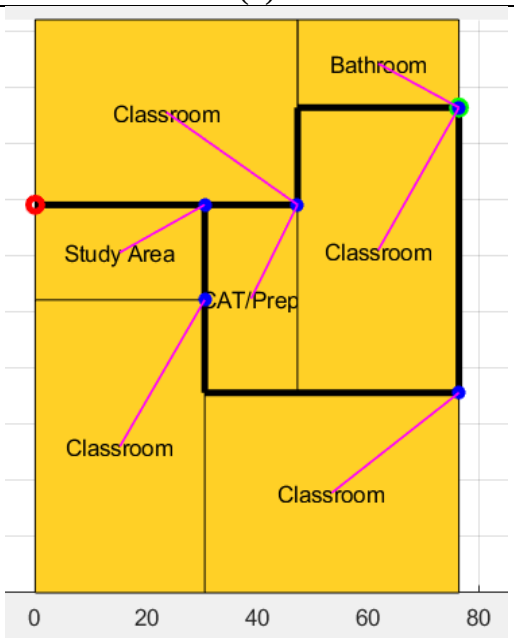
Layout 1 (output from TC 1,5)
Solution time: 0.28 secs.
Aisle network length: 113.8 ft.
(a)



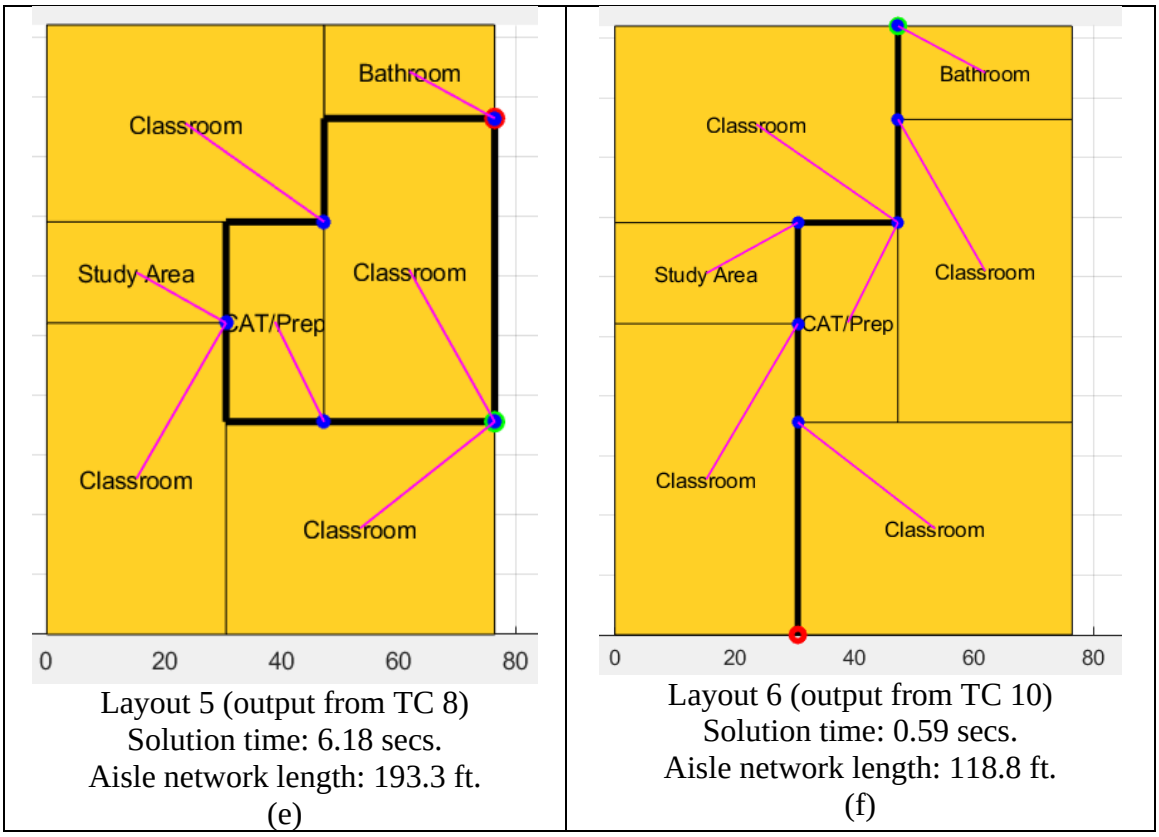
Layout 2 (output from TC 2,6)
Solution time: 0.25 secs.
Aisle network length: 113.8 ft.
(b)



Layout 3 (output from TC 3,4,9)
Solution time: 0.43 secs.
Aisle network length: 118.8 ft.
(c)



Layout 4 (output from TC 7)
Solution time: 29.90 secs.
Aisle network length: 223.9 ft.
(d)



Layout 5 (output from TC 8)
Solution time: 6.18 secs.
Aisle network length: 193.3 ft.
(e)

Layout 6 (output from TC 10)
Solution time: 0.59 secs.
Aisle network length: 118.8 ft.
(f)

Figure 5.9: Aisle network configurations for the small-scale problem

Although certain pairs of the generated aisle networks result in similar layout designs (i.e., layout 1 and 2; layout 4 and 5; layout 3 and 6), there are unique properties about them that make each one distinct. For example, it can be noticed that the separation distance between the primary and secondary accessway is greater in layout 1 compared to layout 2. An additional aisle is necessary for satisfying the accessway separation distance in layout 4, resulting in the total aisle network length to be greater than the aisle length of layout 5 when enforcing unidirectional flow along the aisles. Lastly, the configuration of the doorway for the classroom along the top-border of the facility is different in layouts 3 and 6 since the latter addresses all functional specifications simultaneously (except for unidirectional flow along the aisles since it has a dominating effect compared to the others). As a result, the classroom doorway is moved away from the primary accessway

and towards one of the main circulatory aisles to reduce the travel distance between other departments.

5.5.1.1 Interviews with professional architects

A series of interviews with professional architects were scheduled after generating the layout alternatives in Section 5.5.1 to assess the efficacy of the proposed framework. Feedback was given by the architects while reviewing the layout alternatives regarding the practicality of certain functional specifications, as well as to identify certain factors that were excluded from the model output that are deemed important during the design process. Each interview lasted approximately 45 minutes in length, consisting of an overview of the proposed framework for automated layout generation, a review of the layout alternatives, and a brief question and answer session. This section will discuss the main takeaways from these interviews. All architects were made aware during the interviews that the purpose of this research is not intended to produce a final solution for an architectural project, but instead serve as a baseline that suggests how the departments, aisles, and doorways should roughly be configured for optimizing the quantitative criteria that are in-scope for this research.

One of the most frequently addressed comments was concerning the incorporation of the human experience into the layout design. In other words, how should the layout be modified such that the human experience is optimized? Ergan et al. (2018) administered a survey to 18 architectural experts, and it was found that a combination of quantitative and qualitative criterions are most influential on the human experience, including exposure to nature/lighting, ease of access and openness of spaces, and a balance between isolation and socialization, most of which can be expressed (at least to some degree)

quantitatively. Other qualitative factors that are much more challenging to express quantitatively include texture/material, symmetry of architectural components, contours of objects, spatial alignment, and colors. As the future of design becomes more data-driven, Ergan et al. suggest the use of body area sensor networks for measuring health data of occupants to observe how different stimuli might impact their well-being as a future research-extension.

From an industrial engineering perspective, it is of interest to optimize the facility with respect to operational efficiency measures that are typically expressed quantitatively (i.e., flow cost, production time, etc.). Contrary to this, it is important to capture the human experience for the sake of making people feel comfortable at the facility from an architectural perspective. However, the challenges that arise from this include (a) determining which features of the layout design have the largest effect on the human experience, (b) expressing these features quantitatively, and (c) generalizing the effects of these features since all architects have different perspectives that influence their decision making. After addressing these points to the architects, it was suggested that it is essential to rely on past experiences, best practices, and artistic creativity for reflecting the human experience into the layout rather than solely relying on mathematical equations. One qualitative criteria that was frequently mentioned during the interviews was the amount of daylight that is emitted into certain rooms and the corresponding views that are generated as a result. In addition, there are situations where a subset of departments need to be located along/away from the facility perimeter. Although such requirements can have a dramatic effect on the layout design itself, it is straightforward to include additional constraints that address these qualitative features by forcing certain

departments to be placed along the facility boundary. It was also mentioned that for educational/workplace-related facilities, it is preferred to have classrooms/offices aligned with one another and/or proportional in size for improving the navigability of occupants (similar to the flexible bay structure in FLP).

Following the human experience discussion, the next topic was focused on improving the facility resilience during a pandemic event. As shown from several studies, implementing an aisle network that can support a unidirectional flow of occupants has the potential to reduce the spread of infection, thus allowing for facility operations to continue at a reduced capacity. After presenting the generated layouts in Figure 5.9 (d) and ©, there were some mixed opinions that took a variety of factors into consideration, such as budgetary aspects of a construction project, utilization of the available space, and occupant behavior. There was a concern that configuring an aisle network to support the unidirectional flow of occupants would become costly because the aisle network length would be too long (as shown in Figure 5.9 where the layouts that enforce unidirectional flow are much longer than the others). In addition, the space utilization of the facility would be impacted since some of the square footage allocated to each department would need to be sacrificed for constructing the aisle network. Another interesting point that was made was regarding the behavior of occupants. Although the layout can be designed with the intention of forcing occupants to behave a certain way, there is still the chance that these policies will be ignored by individuals for the sake of improving their personal experience when navigating the facility (such as minimizing the total travel distance). The idea of unidirectional flow in the aisle network was deemed to make sense conceptually amongst the architects, but it is uncertain whether it is practical to do so

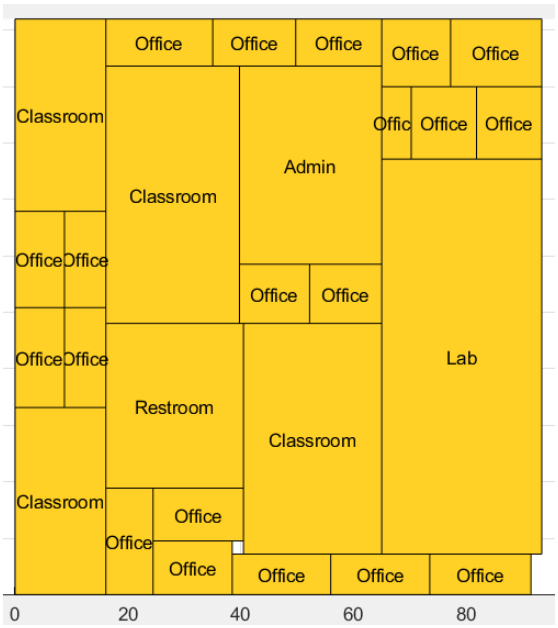
since there exists other strategies that are more cost effective (i.e., natural air ventilation, heating, ventilation, and air conditioning, etc.).

5.5.2 Large-scale case study

The large-scale problem consists of 27 departments within a 93.5' x 97.95' educational facility, including four classrooms (1,003.79 sq. ft., 1,081.05 sq. ft., 534.4 sq. ft., 549.83 sq. ft.), one engineering lab (1,989.42 sq. ft.), one restroom (712.81 sq. ft.; surface area for men and women bathrooms are combined into a single department), one administrative room (886.59 sq. ft.), and 20 offices (120 sq. ft. each). Unlike the previous case-study, two adjacency requirements are included in this problem instance. More specifically, it is desired to have the classrooms and offices adjacent to each other. The adjacency specifications and corresponding objective weights for this problem instance are presented in Table 5.3, where the most and least priority Go to the adjacency and proximity requirements, respectively. The separation requirements are similar in preference, except there being a higher emphasis on separating the restrooms from the offices. The l_1-norm is the selected distance metric in the first phase model. Figure 5.10 displays the block layout that was generated for this instance.

Table 5.3: Adjacency specifications for small-scale problem

	Classrooms	Lab	Restroom	Admin	Office
Classrooms	Adjacent (5)	Proximity (1)	Separate (2)	-	-
Lab	-	-	Separate (2)	-	-
Restroom	-	-	-	Separate (2)	Separate (3)
Admin	-			-	-
Office	-	-	-	-	Adjacent (5)



Solution time: 3,600 seconds (timed out)

Figure 5.10: Block layout for large-scale problem (no clusters)

As mentioned earlier, an unintended phenomenon is observed in the block layout when applying the large-scale problem input data to the first phase model. Observe in Figure 5.10 that there are four clusters of offices scattered throughout the facility, where the clusters along the north and south sides of the facility contain more offices compared to the other two. Similarly, there appears to be one case of adjacent classrooms towards the top-left corner, while the other classrooms are placed in other regions of the layout. Due to the large number of adjacency requirements that are enforced amongst the classrooms and offices, it becomes essentially impossible for all of them to be satisfied simultaneously due to the physical limitations of the facility. At the same time, the computational complexity also increases because of the problem symmetry that exists amongst the homogeneously sized departments (in conjunction with the number of departments in the large-scale problem).

To mitigate the effect of this issue, it was decided to transform the input data by clustering groups of departments together into larger-sized ones, and then apply the first

phase model using the new input data. Upon finding an optimal solution using the transformed input data, each individual department within a cluster can be arranged within the clustered department boundaries. This output can then be fed into the second phase model for generating the corresponding aisle network. Doing so is advantageous in two ways, specifically for improving the tractability of the proposed approach, as well as to make the generated layout designs more attractive for the architect. Additional constraints can be added to the phase one model that require the aspect ratios of clustered departments to be greater than or equal to some threshold value (in consideration of the comment from Section 5.5.1.1 regarding proportionally sized departments). Let α_c^{min} denote the minimum aspect ratio of cluster $c \in C$, where C is an index set containing the indices of all clusters. Also, let ϕ_c be a binary variable equal to 1 if $l_c^x / l_c^y \geq \alpha_c^{min}$; Otherwise, 0 if $l_c^y / l_c^x \geq \alpha_c^{min}$. Note that $C \subseteq I$ because clusters are treated as departments in the first phase model. The following constraints can be added to enforce a lower bound on the aspect ratio.

$$l_c^x \geq l_c^y \alpha_c^{min} - M(1 - \phi_c) \qquad \forall c \in C \qquad (5.71)$$

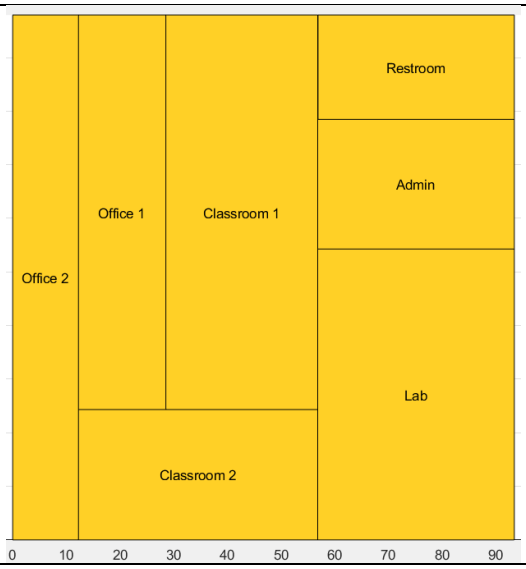
$$l_c^y \geq l_c^x \alpha_c^{min} - M\phi_c \qquad \forall c \in C. \qquad (5.72)$$

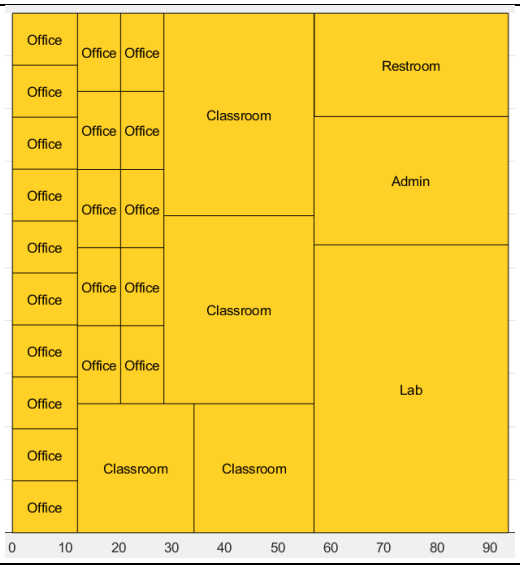
Clustering the classrooms is straightforward since there are only four of them in the problem instance, resulting in Clusters 1 and 2 to consist of classrooms with square footage requirements of 1,003.79 sq. ft. and 1,081.05 sq. ft., and 534.4 sq. ft. and 549.83 sq. ft., respectively. Performing this operation for the offices is not as straightforward because there are 20 of them in total and depending on how the clusters are formed, alternative layout designs will be generated. To demonstrate this, three different cases are considered, where for each case a different number of clusters are used. Table 5.4 provides details behind the configuration of the office clusters for each case.

Table 5.4: Office cluster configurations

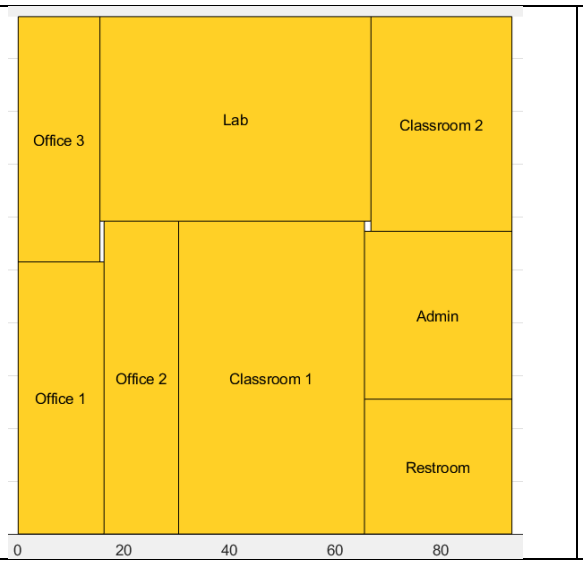
	Number of offices	Area requirement (sq. ft.)
Two clusters	10 / 10	1,200 / 1,200
Three clusters	7 / 7 / 6	840 / 840 / 720
Four clusters	5 / 5 / 5 / 5	600 / 600 / 600 / 600

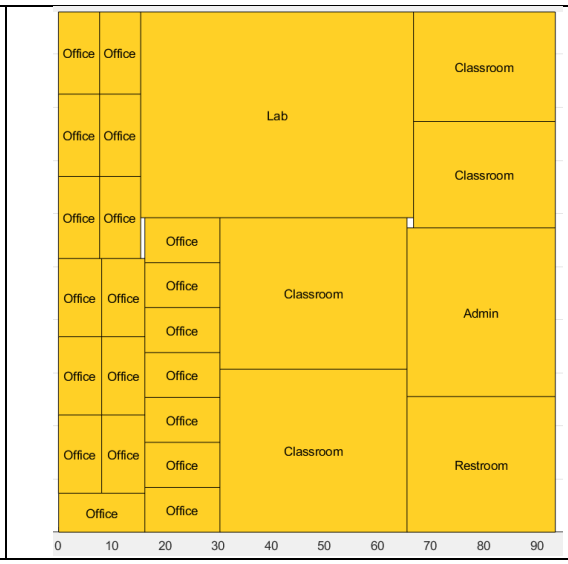
The layouts that are generated after performing the clustering operations and arranging the departments in each cluster are displayed in Figure 5.11. A lower bound was enforced on the aspect ratio of the office clusters using (5.71)-(5.72) in an effort to arrange them in an aligned fashion for improving the layout quality. The lower bound of the office clusters is set to three, while the upper bound is dependent on the number of offices in the cluster. For simplicity, it was decided to increase the maximum aspect ratio value by 1.5 units for each department that is assigned to the cluster. The printed objective function values and solution times in Figure 5.11 are from solving the first phase model using the transformed input data after clustering the departments.



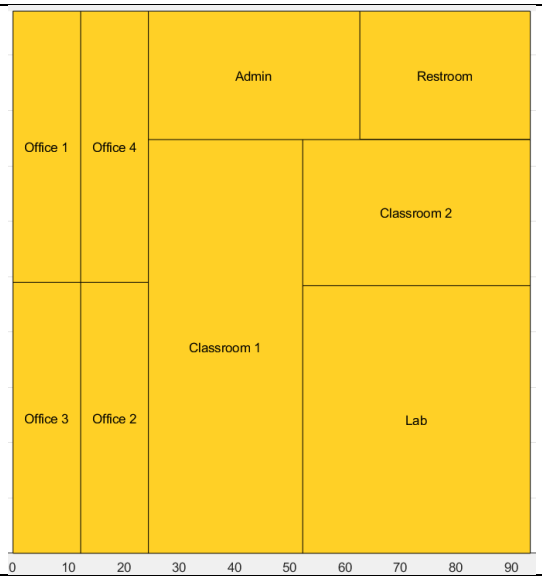


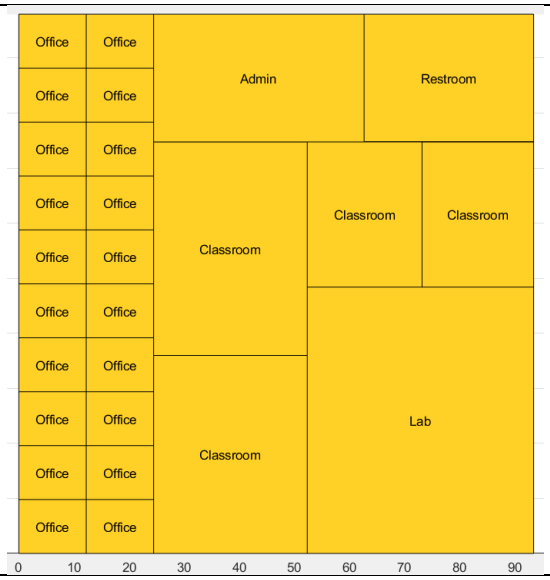
Obj. value: 19.881
Solution time: 41.12 secs.
(a)





Obj. value: 40.584
Solution time: 172.52 secs.
(b)





Obj. value: 59.877
Solution time: 1,179.54 secs.
(c)

Figure 5.11: Block layouts for large-scale problem (with clusters)

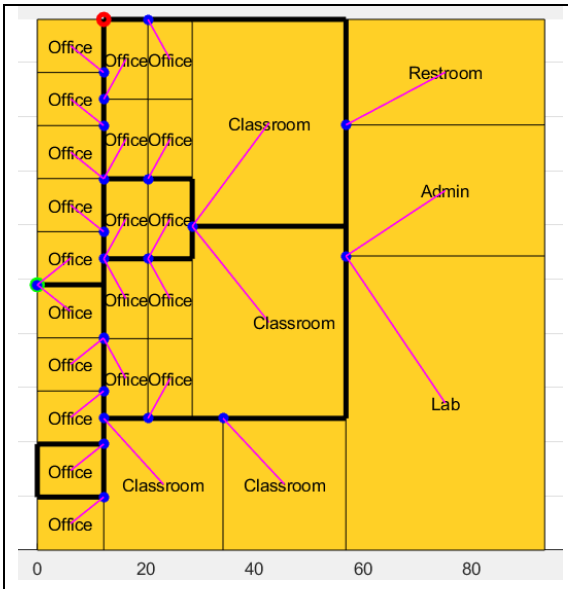
Figure 5.11 (a) shows the offices arranged in two differently shaped clusters, where one of them has offices placed in a vertical orientation (Office 1 cluster) and the other in a horizontal orientation (Office 2 cluster). The classroom clusters are placed directly beside each other, and each classroom is arranged in a similar fashion as the offices in the sense

that one pair of classrooms have a vertical orientation (Classroom 1 cluster) while the other pair are oriented horizontally (Classroom 2 cluster). Similar observations can be made for Figure 5.11 (b) and (c). Empty spaces are apparent in Figure 5.11 (b) because of restricting departments to rectangular shapes, and perhaps as a result of an uneven distribution of offices amongst the three clusters. Figure 5.11 (c) shows all office clusters aligned along the west-region of the facility and the classrooms running along the center across both axes.

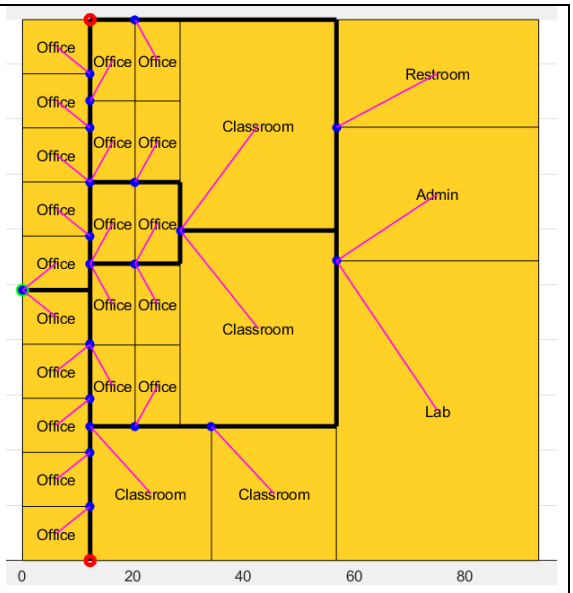
Applying two clusters yielded the best computational performance, whereas applying four clusters yielded the worst computational performance. Arguably, the four-cluster case suffers more from problem-symmetry than the others since the office clusters can freely be swapped amongst one another since they all share equal area requirements (as shown in Table 5.4). This is also evident for the two- and three-cluster cases, but is less severe since they were solved to optimality in a reasonable amount of time. Regardless, this exercise portrays how transforming a large-scale problem into a smaller sized one can benefit the architect in producing layout alternatives.

The corresponding aisle networks for the layouts in Figure 5.11 can now be generated since all 27 departments are placed in the facility. It is assumed that a sprinkler system will be installed, suggesting that a separation distance of at least $L/3$ feet between the accessways is necessary (equal to 45.14 feet). The number of secondary accessways in the facility is manipulated (between one and two in total) during this exercise to demonstrate how the aisle network is impacted by this parameter, and the number of primary accessways is set to one. Zig-zag and dead-end avoidance constraints are incorporated into the second phase model for constructing the aisle networks. The

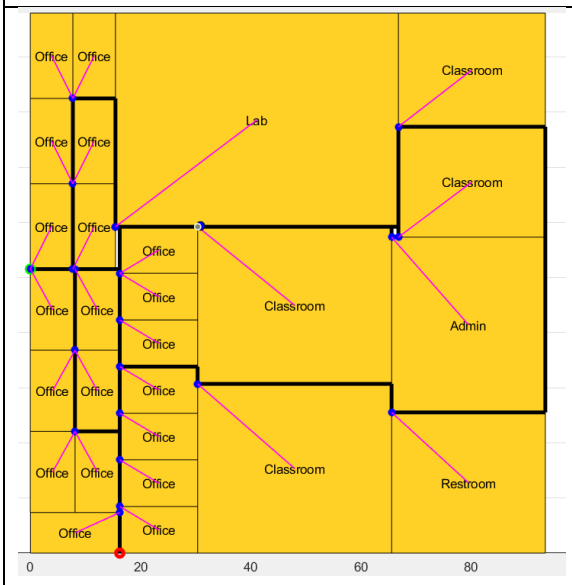
weights assigned to these functional specifications are identical to the ones shown in Section 5.5.1. Figure 5.12 contains the aisle network configurations that are generated for the large-scale problem.



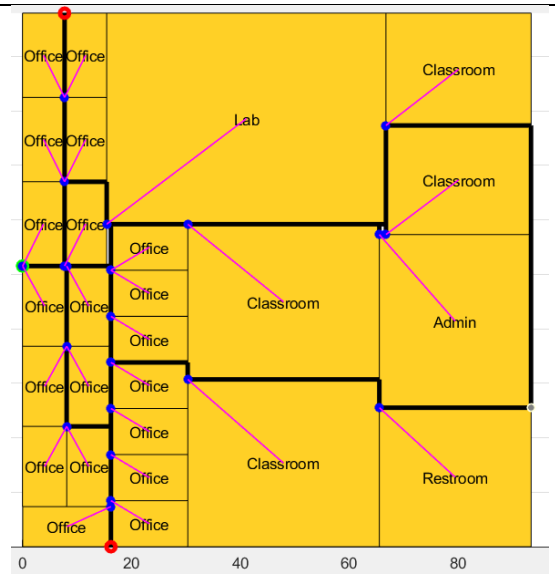
Solution time: 121.13 secs.
Aisle network length: 373.2 ft.
One secondary accessway
(a)



Solution time: 1,052.46 secs.
Aisle network length: 348.7 ft.
Two secondary accessways
(b)



Solution time: 58.18 secs.
Aisle network length: 413.5 ft.



Solution time: 390.15 secs.
Aisle network length: 413.5 ft.

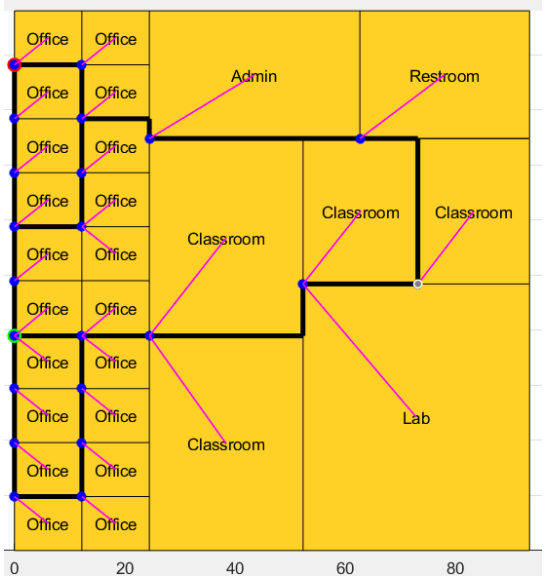
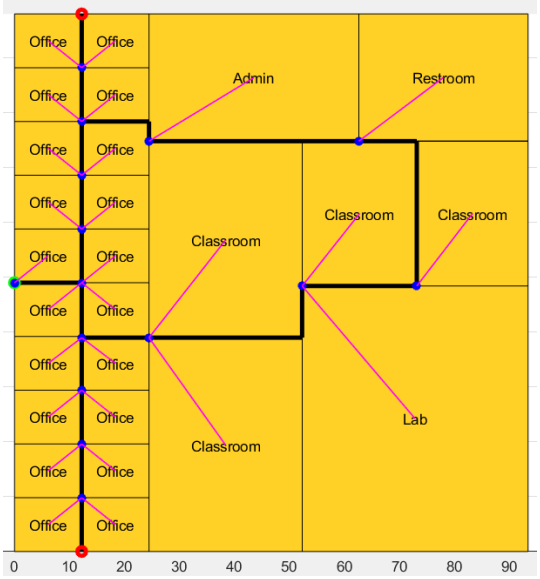
One secondary accessway (c)	Two secondary accessways (d)
	
Solution time: 107.1 secs. Aisle network length: 347.3 ft. One secondary accessway (e)	Solution time: 79.21 secs. Aisle network length: 271.6 ft. Two secondary accessways (f)

Figure 5.12: Aisle network configurations for the large-scale problem

Observe that aisle networks with one secondary accessway require one or more cycles for satisfying the dead-end avoidance constraints. For example, notice in Figure 5.12 (a) how aisles surround one of the offices in the bottom-left corner. Similar behavior is present in Figure 5.12 (c) where there are two sets of offices with aisles constructed along the perimeter of the corresponding offices (also apparent in (c)). Although the offices are still accessible for these cases, it is likely that a fraction of their pre-allocated surface areas will have to be sacrificed to accommodate the aisle network, resulting in a less-efficient utilization of the space. It turns out that the aisle network in Figure5.12 (c) is the shortest compared to the others when one secondary accessway is used.

For the aisle networks with two secondary accessways, fewer cycles are present in Figure 5.12 (b) and Figure 5.12 (f), whereas Figure 5.12 (d) has the same number of

cycles, with the only difference being that one of them surrounds one office instead of two. Longer cycles that run along the boundaries of the classrooms, restroom, admin department, and the engineering lab are present in all layouts, but are essential for improving the navigability of the facility. The aisle networks in Figure 5.12 (b) and Figure 5.12 (f) are similar in the sense that the primary and secondary accessways are configured almost identically, resulting in a long corridor with the secondary accessways at both ends and the primary accessway approximately located at the midpoint. Figure 5.12 (d) displays a similar structure where the endpoints are located at opposite ends of the facility but are misaligned since there is a horizontally oriented office located along the south-wall that prevents the aligned corridor from being built. Similar to the one secondary accessway case, Figure 5.12 (f) yields the shortest aisle network when two secondary accessways are included in the facility. Note that the accessways are placed near the offices in all layouts that were generated. Since most of the departments in the large-scale problem are offices, this results in a dominating effect in the objective function for minimizing the travel distance to the accessways from the offices.

5.5.2.1 Objective weight and functional specification calibration

Given the multi-objective nature of the second-phase model, it is challenging to determine an appropriate assignment of weights in the objective function that results in desired layout properties to be incorporated into the design. Moreover, all architects have different preferences, so it is impractical to use a single set of weights that are applicable for all situations. Although the previous subsections illustrate how layout alternatives can be produced by modifying certain functional specifications of the facility, such as the number of secondary accessways, another layer of inherent complexity is introduced in

the objective function during the optimization process because all objectives are conflicting with each other.

To illustrate how the calibration of objective weights and functional specifications impact the aisle structure, a 2-level factorial experimental design with four factors was created where a subset of the objective weights (departmental travel distance from the primary and secondary accessways, and the aisle generation costs) and the dead-end avoidance functional specification are configured between low- and high-levels (denoted by – and +, respectively). The response variable of interest is the so-called "layout score", which was gathered from an architect that agreed to evaluate each alternative with respect to three criteria, including the separation distance between the accessways, the number of turning points along egress paths, and the length of the aisle network. The location of the department doorways and facility accessways were also addressed in the evaluation. Each layout is assigned a score ranging from 1-10 based on the architect's preferences, where a larger value indicates a more attractive design. One replication is enforced for all combinations of the factor levels, yielding 16 layout alternatives in total for the architect to evaluate.

Table 5.5 displays the values that are assigned to each factor level in the experimental design. Table 5.6 shows the assignment of factor-levels to each layout in the experiment.

Table 5.5: Objective weight and functional specification calibration experimental design

	Levels	
Factor	-	+
1. Distance from departments to primary accessway	1	20
2. Distance from departments to secondary accessway	1	20
3. Aisle generation cost	10	30

4. Dead-end avoidance	Disabled	Enabled

Table 5.6: Experimental design setup

	Factors			
Layout	1	2	3	4
1	-	-	-	-
2	-	-	-	+
3	-	-	+	-
4	-	-	+	+
5	-	+	-	-
6	-	+	-	+
7	-	+	+	-
8	-	+	+	+
9	+	-	-	-
10	+	-	-	+
11	+	-	+	-
12	+	-	+	+
13	+	+	-	-
14	+	+	-	+
15	+	+	+	-
16	+	+	+	+

This experimental design was applied towards the block layouts in Figure 5.8 (a) and 5.11 (c) for the small- and large-scale problems, respectively. It is assumed that one primary and secondary accessway needs to be placed along the facility perimeter by a minimum separation distance of $L/3$. It was requested by the architect to avoid placing the primary and secondary accessways near the offices simultaneously for the large-scale problem due to potential issues that might arise that are unaccounted for during the optimization, such as congestion, lack of privacy, unsatisfactory occupant experience, etc. This was accommodated by excluding the $f_{(m,n)}^{ik}$ decision variables ($\forall k \in A', (m,n) \in E$) in the objective function for all departments labeled as offices. The aisle structures that were produced for the small- and large-scale problems after executing the experimental design and their corresponding scores are presented in Figure A1 and

Figure A2 in the Appendix, respectively. It can be noticed that the highest scored layouts for both problems assign factors 1,2, and 4 to their low-level values, while factor 3 is assigned the high- and low-level value in the large- and small-scale problems, respectively.

From this objective weight and functional specification calibration procedure, a large number of layout alternatives can be produced for the architect to evaluate. However, this procedure is limited in the sense that the architect is restricted to only the layouts that are presented to them, which is also the case for most layout evaluations methods that exist in the literature (as mentioned in Chapter 2). This is of concern because it is possible for all layout alternatives to receive a poor evaluation from the architect, thus increasing the amount of time spent in the pre-design phase. Because of this, researchers have developed interactive systems that enable the user to incorporate their feedback during the layout generation process to eventually arrive to a solution that is most satisfying (García-Hernández et al., 2015; García-Hernandez et al., 2020). It is common for these kinds of systems to rely on metaheuristic algorithms since they are capable of producing a diverse set of solutions in a reasonable amount of time. Although these methods can incorporate the evaluation feedback in an iterative fashion, they fail to guarantee optimality since they rely on randomness rather than exact methods. To the authors' best knowledge, such a system does not exist that utilizes exact methods for incorporating evaluation feedback during the layout optimization process. This dissertation research is not intended to propose a technique to fill this gap in the literature, but to instead emphasize its importance and suggest it is as a future research direction that is worth investigating.

5.6 Conclusion

This study considers the layout design problem in the architectural domain and how to automate the layout design process with respect to user-defined adjacency specifications using mathematical optimization. Existing models that utilize optimization techniques represent the adjacency requirements using binary variables, thus resulting in additional complexity to be introduced to the problem. In addition, these kinds of approaches can result in model infeasibility if an excess number of adjacency requirements are specified since it may not be possible to satisfy all of them simultaneously due to the physical limitations of the facility. This is disadvantageous for the architect because it is unclear which adjacency requirements will be violated prior to the optimization. This also constrains the architect to a lower level of detail for a problem instance due to the risk of model infeasibility. The proposed model addresses this shortcoming by using deviational variables that measure the total distance in which an adjacency requirement is violated without requiring binary variables, with the goal of minimizing the sum of all deviational variables in the objective function.

In addition, existing models in the architecture domain using mathematical optimization only address the adjacency requirements, whereas the proximity and separation requirements are disregarded. Excluding these requirements from the automated layout generation process may result in suboptimal designs to be generated. To accommodate this, the proposed model allows for all adjacency specifications to be addressed simultaneously by minimizing the total deviation between the specifications with respect to the arrangement of departments in the layout design. Following the

generation of the layout design, another optimization model is formulated for configuring the aisle structure to control occupant flow within the facility interior.

Future research directions include incorporating qualitative criteria during the layout generation process since existing studies show that they have a meaningful impact on how occupants behave inside of the facility. Also, further refinement of the calibration procedure in Section 5.5.2.1 can be investigated for appropriately incorporating feedback from architects to help produce layout alternatives that align with their preferences through the use of exact methods. Another research direction consists of relaxing the assumption where the facility blueprint needs to be specified a priori. Doing so would extend the applicability of the approach to consider cases where the site dimensions are unknown, or its surface area is greater than the cumulative surface area of the departments. Lastly, exploring additional valid inequalities and other model improvements to enhance the computational efficacy is also another promising research direction.

CHAPTER 6: CONCLUSION

In this dissertation, the facility layout problem (FLP) was considered. FLP is an important problem to address because it is directly correlated with how efficient operations are at the facility. Poorly designed facility layouts can result in excessive operational spending to occur. In the manufacturing setting, between 20% to 50% of operating expenses are contributed by the configuration of the material handling system, and can potentially be reduced by 10% to 30% if the facility is appropriately designed (Meller and Gau, 1996). This illustrates the impact that FLP can have not only in the manufacturing sector, but for other facility types as well because the same methodologies that have been applied since the 1950s can easily be extended to solve different layout problems. Although FLP does not have an overly complicated problem statement, the difficulty associated with it stems from the fact that it is an NP-hard combinatorial optimization problem. As a result, it is challenging to optimally solve relatively larger-sized problem instances using exact methods. This has forced researchers to exploit heuristic approaches that can generate good-quality solutions, but fail to guarantee an optimal solution once the procedure terminates.

Some of the shortcomings in the FLP literature were addressed and investigated across a set of separate (but still interconnected) research efforts in this dissertation. Shortcomings in the literature include (but not limited to) (1) coping with the computational complexity of the problem, (2) consideration of pandemic related events during the layout design process, and (3) incorporating proximity and separation requirements in conjunction with desired adjacencies that are desired in the layout. The theme that ties these shortcomings together is the idea of clearances, which are used for

separating objects in the layout design by a minimum threshold distance due to factors such as noise, compliance, efficiency, etc. To accommodate these shortcomings, a variety of optimization models were introduced that can be used for (1) generating layout designs more efficiently from a computational standpoint, (2) assisting restaurant owners in redesigning their facilities with respect to social distancing and reduced capacity constraints and (3) allowing architects to specify additional adjacency specifications for generating more desired layout designs in an automated fashion.

6.1 Contributions

The work presented in this dissertation is advantageous to the public and scientific communities in a variety of ways. First, improving the computational tractability allows for a larger spectrum of problem instances to be solved to optimality, thus increasing the applicability of mathematical programming to layout practitioners in public and scientific sectors. Also, the proposed model adjustments for improving the tractability of DRLP is a novel contribution to the scientific community pool of knowledge that allows other researchers to make further improvements based on the conclusions which were drawn in this dissertation.

Second, the incorporation of recently emerging criteria due to the COVID-19 pandemic, namely social distancing and reduced capacity, into the layout design process can assist in reducing the spread of infection amongst occupants. Since studies that address these factors are currently scarce, this research helps lay the foundation for encouraging new investigations to improve facility resilience during pandemic events. In addition, merging FLP and restaurant revenue management related problems, such as TMP and PMP, in this context results in a new optimization framework that can be used

for improving business operations under normal or pandemic-induced conditions. Lastly, providing architects with an automated layout generation tool that incorporates additional design features that were not previously addressed allows for more detailed layout designs to be created. Doing so helps them save time during the pre-design phase of the project, thus giving them more time to focus on other critical tasks.

6.2 Limitations

One of the main limitations of this dissertation is restricting departments to only rectangular shapes. There are cases where nonlinear shaped departments may result in additional efficiency, but due to the computational complexity that is present when introducing these nonlinearities, it was decided to primarily focus on rectangular shapes. Also, in Chapter 4, it is assumed that customers arrive at the restaurant simultaneously in the second-stage model based on the deployed hourly arrival rate. This does not accurately resemble how restaurants operate since customers typically enter and exit the restaurant throughout the hour. Even though this simplifies the problem setting and allows for decomposition algorithms to be used for finding an optimal solution, it is still unknown if incorporating queueing theory dynamics into the modelling process can result in higher quality solutions to be produced.

Another limitation is directed towards Chapter 5, where a series of parameters and weights are introduced for incorporating adjacency specifications and objective preferences to produce layout alternatives. Proper calibration of these values is a non-trivial task, which inherently introduces another layer of complexity during the layout design process. Although a simple procedure is proposed for weight calibration, additional research is still needed to streamline this process for architects with differing

perspectives/preferences. Along with this, occupancy rates of departments are ignored when generating the layout design. This can be concerning to architects because many building code requirements reference the departmental occupancy rates, and can have a dramatic effect on the feasibility of layouts that are produced from the approach. Finally, the proposed automated layout generation model is applicable towards facilities with only one floor. The mathematical model can be modified to accommodate multiple floors, but the computational complexity will greatly increase as a result.

6.3 Future research directions

The findings from this dissertation present several opportunities for future research. One research direction is the investigation of valid inequalities and further reducing the size of the models by removing unnecessary binary decision variables. All models in this dissertation are formulated as MILPs, so any efforts that can be made for improving their respective efficacies would help make these approaches more attractive in practice. Also, revisiting the stochastic programming model in Chapter 4 is another promising direction to assess if combining tables and incorporating a queueing system would further improve the restaurant operations. Finally, enhancing the objective weight and functional specification calibration procedure in Chapter 5 for further establishing a "feedback loop" between the physical and psycho-physical properties would be an invaluable contribution to the field since there are currently no approaches in the literature that allow for feedback to be incorporated when applying exact methods. Introducing qualitative criteria during the layout generation process would also be helpful in improving the quality of generated solutions.

REFERENCES

Abedzadeh, M., Mazinani, M., Moradinasab, N. and Emad, R. (2014). Parallel variable neighborhood search for solving fuzzy multi-objective dynamic facility layout problem. International Journal of Advanced Manufacturing Technology, 75(5–8), 687–691.

Adolphson, D. and Hu, T. (1973). Optimal linear ordering. SIAM Journal on Applied Mathematics, 25(3), 403-423.

Adrian, A., Utamima, A. and Wang, K. (2015). A comparative study of GA, PSO and ACO for solving construction site layout optimization. KSCE Journal of Civil Engineering, 19(3), 520–527.e

Afentakis, P. (1989). A loop layout design problem for flexible manufacturing systems. International Journal of Flexible Manufacturing Systems, 1, 175-196.

Ahmadi, A. and Jokar, M. (2016). An Efficient Multiple-Stage Mathematical Programming Method for Advanced Single and Multi-Floor Facility Layout Problems. Applied Mathematical Modelling, 40(9–10). 5605–5620.

Ahmadi-Javida, A. and Ardestani-Jaafari, A. (2020). The unequal area facility layout problem with shortest single-loop AGV path: how material handling method matters. International Journal of Production Research, 59(8), 2352-2374.

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

Aiello, G., La Scalia, G. and 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.

Aiello, G., La Scalia, G. and Mario, E. (2013). A non dominated ranking Multi Objective Genetic Algorithm and electre method for unequal area facility layout problems. Expert Systems with Applications, 40(12), 4812–4819.

Alaimo, R. and Lim, C. (2020). Double row layout test problems, DOI: https://doi.org/10.15139/S3/OIJN6V, UNC Dataverse, V1.

Alexander, C., Ishikawa, S.,and Silverstein, M. (1977). A Pattern Language: Towns, Buildings, Construction. Oxford University Press.

Al-Hawari, T., Mumani, A. and Momani, A. (2014). Application of the Analytic Network Process to facility layout selection. Journal of Manufacturing Systems, 33(4), 488-497.

Altuntas, S. and Selim, H. (2012). Facility layout using weighted association rule-based data mining algorithms: Evaluation with simulation, Expert Systems with Applications, 39(1), 3–13.

Amaral, A. (2012) The corridor allocation problem. Computers & Operations Research, 39(12), 3325–3330.

Amaral, A. (2013a). A parallel ordering problem in facilities layout. Computers & Operations Research, 40(12), 2930–2939.

Amaral, A. (2013b). Optimal solutions for the double row layout problem. Optimization Letters, 7, 407–413.

Amaral, A. (2019). A mixed-integer programming formulation for the double row layout of machines in manufacturing systems. International Journal of Production Research, 57(1), 34–47.

Amaral, A. (2020). A mixed-integer programming formulation of the double row layout problem based on a linear extension of a partial order. Optimization Letters, DOI: https://doi.org/10.1007/s11590-020-01651-7.

Anjos, M. and Vieira, M. (2016). An Improved Two-Stage Optimization-Based Framework for Unequal-Areas Facility Layout. Optimization Letters, 10(7), 1379–1392.

Anjos, M. and Vieira, M. (2017). Mathematical optimization approaches for FLPs: The state-of-the-art and future research directions. European Journal of Operational Research, 261(1), 1-16.

Armour, G. and Buffa, E. (1963). A Heuristic Algorithm and Simulation Approach to Relative Location of Facilities. Management Science, 9(2), 294-309.

Arnolds, I., and Nickel, S. (2015). "Layout planning problems in health care." Applications of Location Analysis, International Series in Operations Research & Management Science, H. A. Eiselt and V. Marianov, eds., Springer International Publishing, Cham, Switzerland, 109–152.

Arthur, P. and Passini, R. (1992). Wayfinding: people, signs, and architecture. McGraw-Hill (1st edition).

Asef-Vaziri, A. and Kazemi, M. (2018). Covering and connectivity constraints in loop-based formulation of material flow network design in facility layout. European Journal of Operational Research, 264(3), 1033-1044.

Asef-Vaziri, A., Jahandideh, H. and Modarres, M. (2017). Loop-based facility layout design under flexible bay structures. International Journal of Production Economics, 193, 713-725.

Bashiri, M. and Dehghan, E. (2010). Optimizing a multiple criteria dynamic layout problem using a simultaneous data envelopment analysis modeling. International Journal on Computer Science and Engineering, 2(1), 48–55.

Bate, P., and Robert, G. (2007). Bringing User Experience to Healthcare Improvement: The Concepts, Methods and Practices of Experience-Based Design. Radcliffe, Oxford, U.K

Benders, J. (1962). Partitioning procedures for solving mixed-variables programming problems. Numerische Mathematik, 4, 238-252.

Bernardi, S. and Anjos, M. (2013). A Two-Stage Mathematical-Programming Method for the Multi-Floor Facility Layout Problem. Journal of the Operational Research Society, 64(3), 352–364.

Bertsimas, D., Shioda, R. (2013). Restaurant revenue management. Operations Research, 51(3), 472-486.

Birgin, E., Lobato, R. (2010). Orthogonal packing of identical rectangles within isotropic convex regions. Computers and Industrial Engineering. 59(4), 595-602.

Bortolete, J., et al. (2021). A support planning tool for planning classrooms considering social distancing between students. Optimization Online. http://www.optimization-online.org/DB_FILE/2021/04/8343.pdf.

Bozer, Y. and Wang, C. (2012). A graph-pair representation and MIP-model-based heuristic for the unequal-area facility layout problem. European Journal of Operational Research, 218(2), 382–391.

Bozorgi, N., Abedzadeh, M. and Zeinali, M. (2015). Tabu search heuristic for efficiency of dynamic facility layout problem. International Journal of Advanced Manufacturing Technology, 77(1–4), 689–703.

Bukchin, Y. and Tzur, M. (2014). A New MILP Approach for the Facility Process-Layout Design Problem with Rectangular and L/T Shape Departments. International Journal of Production Research, 52(24), 7339–7359.

Caputo, A., Pelagagge, P., Palumbo, M. and Salini, P. (2015). Safety-based process plant layout using genetic algorithm. Journal of Loss Prevention in the Process Industry, 34, 139–150.

Caron, G., Hansen, P., and Jaumard, B. (1999) The assignment problem with seniority and job priority constraints. Operations Research, 47(3), 449-453.

Chae, J. and Regan, A. (2016). Layout Design Problems with Heterogeneous Area Constraints. Computers & Industrial Engineering, 102, 198–207.

Chae, J. and Regan, A. (2020). A mixed integer programming model for a double row layout problem. Computers and Industrial Engineering, 140, article 106244.

Chang et al. (2021) Mobility network models of COVID-19 explain inequities and inform reopening. Nature, 589, 82-87.

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

Chen, C., Chacón Vega, R. and Kong, T. (2020). Using genetic algorithm to automate the generation of an open-plan office layout. International Journal of Architectural Computing, DOI: https://doi.org/10.1177/1478077120943532

Chu, D., Akl, E., Duda, S., Solo, K., Yaacoub, S. Schünemann, H. (2020). Physical distancing, face masks, and eye protection to prevent person-to-person transmission of SARS-CoV-2 and COVID-19: a systematic review and meta-analysis. The Lancet.

Chung, J. and Tanchoco, J. (2010). The double row layout problem. International Journal of Production Research, 48(3), 709–727.

Coffman, E., Garey, M. and Johnson, D. (1997) Approximation Algorithms for Bin Packing: A Survey. In Approximation Algorithms for NP-Hard Problems, Dorit S. Hochbaum (editor), PWS Publishing Company, 46-93.

Cooper, R. Easing restrictions on travel, business operations, and mass gathering: Phase 2, Executive Order No. 141. North Carolina Office of the Governor. https://governor.nc.gov/documents/executive-order-no-141. Accessed 20 May 2020.

Cornu´ejols, G. (2008) Valid inequalities for mixed integer programs. Mathematical Programming. 112(1),3–44.

DaGue, A. How to Create a Perfect Restaurant Layout with Examples & Ideas. Eat App. https://restaurant.eatapp.co/blog/management/restaurant-floor-plan. Accessed 3 June 2020.

Datta, D., Amaral, A. and Figueira, J. (2011). Single row facility layout problem using a permutation-based genetic algorithm. European Journal of Operational Research., 213(2), 388–394.

Derakhshan, A. and Wong, K. (2017). Solving unequal-area static and dynamic facility layout problems using modified particle swarm optimization. Journal of Intelligent Manufacturing, 28(6), 1317–1336.

Designing Your Restaurant's Dining Room Layout. WebstaurantStore. https://www.webstaurantstore.com/article/118/dining-room-design.html. Accessed 3 June 2020.

Díaz-Ovalle,C., Vázquez-Román, R. and Mannan, M. (2010). An Approach to Solve the Facility Layout Problem Based on the Worst-Case Scenario. Journal of Loss Prevention in the Process Industries, 23(3), 385–392.

Dokeroglu, T. (2015). Hybrid teaching-learning-based optimization algorithms for the Quadratic Assignment Problem. Computers & Industrial Engineering, 85, 86–101.

Dube, K., Nhamo, G. and Chikodzi, D. (2021). COVID-19 cripples global restaurant and hospitality industry. Current Issues in Tourism, 24(11), 1487-1490.

Emami, S. and Nookabadi, A. (2013). Managing a new multi-objective model for the dynamic facility layout problem. International Journal of Advanced Manufacturing Technology, 68(9–12), 2215–2228.

Ergan, S., Zhuoya, S. and Yu, X. (2018). Towards quantifying human experience in the built environment: A crowdsourcing based experiment to identify influential architectural design features. Journal of Building Engineering, 20, 51.59.

Ertay, T., Ruan, D. and Tuzkaya, U. (2006). Integrating data envelopment analysis and analytic hierarchy for the facility layout design in manufacturing systems. Information Sciences, 176(3), 237–262.

Fabregas, K. Planning Your Restaurant Floor Plan – Step-by-Step Instructions. Fit Small Buiness. https://fitsmallbusiness.com/restaurant-floor-plan/. Accessed 3 June 2020.

Ferjani, A., Pierreval, H. and Frikha, A. (2019). Evaluation of Layout Designs Taking into Account Workers' Fatigue. Proceedings of the 2019 International Conference on Industrial Engineering and Systems Management, IESM 2019, 1–6.

Ficko, M. and Palcic, I. (2013). Designing a layout using the modified triangle method, and genetic algorithms. International Journal of Simulation Modelling, 12(4), 237–251.

Fischetti, M., Fischetti, M., Stoustrup, J. (2021). Safe distancing in the time of COVID-19. European Journal of Operational Research. DOI: https://doi.org/10.1016/j.ejor.2021.07.010.

Foulds, L. and Partovi, F. (1998). Integrating the analytic hierarchy process and graph theory to model facilities layout. Annals of Operations Research, 82, 435–451.

Foulds, L. and Robinson, D. (1978). Graph theoretic heuristics for the plant layout problem. International Journal of Production Research, 16(1), 27–37.

Friedrich, C., Klausnitzer, A. and 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.

Gao, H., Medjdoub, B., Luo, H., Zhong, H., Zhong, B. and Sheng, D. (2020). Building evacuation time optimization using constraint-based design approach. Sustainable Cities and Society, 52, 101839.

García-Hernández, L., Palomo-Romero, J., Salas-Morera, L., Arauzo-Azofra, A. and Pierreval, H. (2015). A novel hybrid evolutionary approach for capturing decision maker knowledge into the unequal area facility layout problem. Expert Systems with Applications, 42(10), 4697–4708.

Garcia-Hernandez, L., Pierreval, H., Salas-Morera, L. and Arauzo-Azofra, A. (2013). Handling qualitative aspects in Unequal Area Facility Layout Problem: An Interactive Genetic Algorithm. Applied Soft Computing Journal, 13(4), 1718–1727.

Garcia-Hernandez, L., Salas-Morera, L., Carmona-Muñoz, C., Abraham, A. and S. Salcedo-Sanz (2020) A novel multi-objective interactive coral reefs optimization algorithm for the unequal area facility layout problem. Swarm and Evolutionary Computation, article #100688.

Garey, M., Johnson, D. and Stockmeyer, L. (1976). Some simplified NP-complete graph problems. Theoretical Computer Science, 1(3), 237-267.

Ghassemi Tari, F. and Neghabi, H. (2015). A New Linear Adjacency Approach for Facility Layout Problem with Unequal Area Departments. Journal of Manufacturing Systems, 37(1), 93–103.

Ghassemi Tari, F. and Neghabi, H. (2018). Constructing an Optimal Facility Layout to Maximize Adjacency as a Function of Common Boundary Length. Engineering Optimization, 50(3), 499–515.

Glover, F., McMillian, C. and Novick, B. (1985). Interactive decision software and computer graphics for architectural and space planning. Annals of Operations Research. 5, 557-573.

Goetschalckx, M. (1992). An Interactive Layout Heuristic Based on Hexagonal Adjacency Graphs. European Journal of Operational Research, 63, 304-321.

Gomez, A., Fernandez, Q., De la Fuente Garcia, D. and Garcıa, P. (2003). Using genetic algorithms to resolve layout problems in facilities where there are aisles. International Journal of Production Economics, 84 (3), 271–282.

Gonçalves, J. and Resende, M. (2015). A biased random-key genetic algorithm for the unequal area facility layout problem. European Journal of Operational Research, 246(1), 86–107.

Grason, J. (1971). An approach to computerized space planning using graph theory. Proceedings of the 8th Design Automation Workshop. 170-178.

Guan, J. and Lin, G. (2016). Hybridizing variable neighborhood search with ant colony optimization for solving the single row facility layout problem. European Journal of Operational Research, 248 (3), 899–909.

Guan, J., Lin, G., Feng, H. and Ruan, Z. (2020). A decomposition-based algorithm for the double row layout problem. Applied Mathematical Modeling, 77(1), 963–979.

Guerriero, F., Miglionico, G. and Olivito, F. (2014). Strategic and operational decisions in restaurant revenue management. European Journal of Operational Research, 237, 1119-1132.

Gülşen, M., Murray, C. and Smith, A. (2019). Double-row facility layout with replicate machines and split flows. Computers & Operations Research, 108, 20–32.

Guo, Z. and Li, B. (2017). Evolutionary approach for spatial architecture layout design enhanced by an agent-based topology finding system. Collection of Frontiers of Architectural Research. 6(1), 53-62.

Gurobi Optimization, Inc. Gurobi Optimizer Reference Manual (8.1.1). Retrieved from https://www.gurobi.com/documentation/8.1/refman/index.html. (2018).

Hadi-Vencheh, A. and Mohamadghasemi, A. (2013). An integrated AHP-NLP methodology for facility layout design. Journal of Manufacturing Systems, 32(1), 40.

Halawa, F., Madathil, S., Gittler, A. and Khasawneh, M. (2020). Advancing evidence-based healthcare facility design: a systematic literature review. Health Care Management Science, 23(3), 453–480.

Hammad, A., Akbarnezhad, A. and Rey, D. (2016). A multi-objective mixed integer nonlinear programming model for construction site layout planning to minimise noise pollution and transport costs. Automation in Construction, 61, 73-85.

Hassan, M. and Hogg G. (1991). On constructing a block layout by graph theory. International Journal of Production Research, 29(6), 1263–1278.

Heragu, S. and Alfa, A. (1992). Experimental analysis of simulated annealing based algorithms for the layout problem. European Journal of Operational Research, 57(2), 190–202.

Heragu, S. and Kusiak, A. (1988). Machine Layout Problem in Flexible Manufacturing Systems. Operations Research, 36(2), 258-268.

Herran, A., Colmenar, J. and Duarte, A. (2021). An efficient variable neighborhood search for the Space-Free Multi-Row Facility Layout problem. European Journal of Operational Research, 295(3), 893-907.

Hoffman, A. and Kruskal, J. (1956). Integral Boundary Points of Convex Polyhedra. Linear Inequalities and Related Systems, Annals of Mathematics Studies, 38, Princeton (NJ): Princeton University Press, 223–246.

Hosseini, S., Al Khaled, A. and Vadlamani, S. (2014). Hybrid imperialist competitive algorithm, variable neighborhood search, and simulated annealing for dynamic facility layout problem. Neural Computing with Applications, 25(7–8), 1871–1885.

Hosseini, S., Yazdani, R. and Fuente, A. (2020). Multi-objective interior design optimization method based on sustainability concepts for post-disaster temporary housing units. Building and Environment, 173, 106742.

Hosseini-Nasab, H. and Emami, L. (2013). A hybrid particle swarm optimisation for dynamic facility layout problem. International Journal of Production Research, 51(14), 4325–4335.

Hu, R., Huang, Z., Tang, Y., Kaick, O., Zhang, H. and Huang, H. (2020). Graph2Plan: learning floorplan generation from layout graphs. ACM Transactions on Graphics. 39(4), 1-14.

Hua, H., Hovestadt, L., Tang, P. and Biao, L. (2019). Integer programming for urban design. European Journal of Operational Research, 274(3), 1125-1137.

Hwang, J. (2008). Restaurant table management to reduce customer waiting times. Journal of Foodservice Business Research. 11, 334-351.

Hwang, J., Gao, L., Jang, W. (2010). Joint demand and capacity management in a restaurant system. European Journal of Operational Research. 207(1), 465-472.

International Building Code. (2021). International Code Council, Inc. https://codes.iccsafe.org/content/IBC2021P1. Accessed 3 June 2020.

Izui, K., Murakumo, Y., Suemitsu, I., Nishiwaki, S., Noda, A. and Nagatani, T. (2013). Multiobjective layout optimization of robotic cellular manufacturing systems. Computers & Industrial Engineering, 64(2), 537–544.

Jankovits, I., Luo, C., Anjos, M. and Vannelli, A. (2011). A Convex Optimisation Framework for the Unequal-Areas Facility Layout Problem. European Journal of Operational Research, 214(2), 199–215.

Jiang, S. and Nee, A. (2013). A novel facility layout planning and optimization methodology. CIRP Annals, 62(1), 483–486.

Jolai, F., Tavakkoli-Moghaddam, R. and Taghipour, M. (2012). A multi-objective particle swarm optimisation algorithm for unequal sized dynamic facility layout problem with pickup/drop-off locations. International Journal of Production Research, 50(15), 4279–4293.

Kaku, B. and Rachamadugu, R (1992). Layout design for flexible manufacturing systems. European Journal of Operational Research, 57(2), 224-230.

Kalita, Z. and Datta, D. (2014). Solving the bi-objective corridor allocation problem using a permutation-based genetic algorithm. Computers & Operations Research, 52(PART A), 123–134.

Kamol, K. and Krung, S. (2005). Optimizing Architectural Layout Design via Mixed Integer Programming. Computer Aided Architectural Design Futures 2005. 175-184.

Kang, S. and Chae, J. (2017). Harmony search for the layout design of an unequal area facility. Expert Systems with Applications, 79, 269–281.

Kang, S., Kim, M. and Chae, J. (2018). A closed loop based facility layout design using a cuckoo search algorithm. Expert Systems with Applications: An International Journal, 93(C), 322-335.

Karlen, M. and Fleming, R. (2016). Space Planning Basics (4th edition). Wiley, Hoboken, New Jersey.

Karmarkar, S. and Dutta, G. (2011). Optimal table-mix and acceptance-rejection problems in restaurants. International Journal of Revenue Management, 5(1), 1-15.

Keller, B. and Buscher, U. (2015). Single row layout models. European Journal of Operational Research, 245(3), 629–644.

Khaksar-Haghani, F., Kia, R., Mahdavi, I. and Kazemi, M. (2013). A genetic algorithm for solving a multi-floor layout design model of a cellular manufacturing system with alternative process routings and flexible configuration. International Journal of Advanced Manufacturing Technology, 66(5–8), 845–865.

Kia, R., Khaksar-Haghani, F., Javadian, N. and Tavakkoli-Moghaddam, R. (2014). Solving a multi-floor layout design model of a dynamic cellular manufacturing system by an efficient genetic algorithm. Journal of Manufacturing Systems, 33(1), 218–232.

Kiaran, A and Tansel, BC. (1989). Optimal pickup point location in material handling networks. International Journal of Production Research, 27, 1475–1486.

Kim, J and Kim, Y. (1999). A branch and bound algorithm for locating I/O points of departments on the block layout. Journal of the Operational Research Society, 50, 517–525.

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

Kim, M. and Chae, J. (2019). Monarch Butterfly Optimization for Facility Layout Design Based on a Single Loop Material Handling Path. Mathematics, 7(2), 1-21.

Klausnitzer, A. and Lasch, R. (2019). Optimal facility layout and material handling network design. Computers and Operations Research, 103, 237-251.

Komarudin and Wong, K. (2010). Applying Ant System for solving Unequal Area Facility Layout Problems. European Journal of Operational Research, 202(3),730–746.

Koopmans, T. and Beckmann, M. (1957). Assignment problems and the location of economic activities. Econometric. 25(1), 53-76.

Kothari, R. and Ghosh, D. (2013). Tabu search for the single row facility layout problem using exhaustive 2-opt and insertion neighborhoods. European Journal of Operational Research, 224(1), 93–100.

Kothari, R. and Ghosh, D. (2014). A scatter search algorithm for the single row facility layout problem. Journal of Heuristics, 20(2), 125–142.

Krishnan, K., Mirzaei, S., Venkatasamy, V. and Pillai, V. (2012). A comprehensive approach to facility layout design and cell formation. International Journal of Advanced Manufacturing Technology, 59(5–8), 737–753.

Ku, M., Hu, M. and Wang, M. (2011). Simulated annealing based parallel genetic algorithm for facility layout problem. International Journal of Production Research, 49(6), 1801–1812.

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. (2017). The Zone-Based Dynamic Facility Layout Problem. Infor, 57(1), 1–31.

Kulturel-Konak, S. and 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. and Konak, A. (2011). Unequal area flexible bay facility layout using ant colony optimization. International Journal of Production Research, 49(7), 1877–1902.

Kulturel-Konak, S. and Konak, A. (2013). Linear programming based genetic algorithm for the unequal area facility layout problem. International Journal of Production Research, 51(14), 4302–4324.

Kulturel-Konak, S. and Konak, A. (2015). A large-scale hybrid simulated annealing algorithm for cyclic facility layout problems," Engineering Optimization, 47(7), 963–978.

Kumar, S. and Cheng, J. (2015). A BIM-based automated site layout planning framework for congested construction sites. Automation in Construction, 59, 24-37.

Kusiak, A. and Heragu, S. (1987). The facility layout problem. European Journal of Operational Research, 29(3), 229–251.

Laignel, G., Pozin, N., Geffrier, X., Delevaux, L., Brun, F. and Dolla, B. (2021). Floor plan generation through a mixed constraint programming-genetic optimization approach. Automation in Construction. 123, 1-21.

Latifi, S., Mohammadi, E. and Khakzad, N. (2017). Process Plant Layout Optimization with Uncertainty and Considering Risk. Computers & Chemical Engineering, 106, 224–242.

Lausnitzer, A. and Lasch, R. (2019). Optimal facility layout and material handling network design. Computers & Operations Research, 103, 237-251.

Lear. Safe Work Playbook. https://www.tesla.com/sites/default/files/blog_attachments/Tesla-Return-to-Work-Playbook.pdf?redirect=no. Accessed 20 May 2020.

Lee, K., Roh, M., and Jeong, H. (2005). An improved genetic algorithm for multi-floor facility layout problems having inner structure walls and passages. Computers & Operations Research, 32 (4), 879–899.

Lee, M., Kim, J. and Kim, Y. (2009). Linear programming and Lagrangian relaxation heuristics for designing a material flow network on a block layout. International Journal of Production Research, 47(18), 5185-5202.

Lenaghan, P. and Schwedhelm, M. (2015). Nebraska biocontainment unit design and operations. The Journal of Nursing Administration. 45(6), 298-301.

Lenin, N., Siva Kumar, M., Islam, M. and Ravindran, D. (2013). Multi-objective optimization in single-row layout design using a genetic algorithm. International Journal of Advanced Manufacturing Technology, 67(5–8), 1777–1790.

Leno, I., Sankar, S. and Ponnambalam, S. (2016). An elitist strategy genetic algorithm using simulated annealing algorithm as local search for facility layout design. International Journal of Advanced Manufacturing Technology, 84(5–8), 787–799.

Leno, I., Sankar, S., Raj, M. and Ponnambalam, S. (2013). An elitist strategy genetic algorithm for integrated layout design. International Journal of Advanced Manufacturing Technology, 66(9–12). 1573–1589.

Li, J., Tan, X. and Li, J. (2018). Research on Dynamic Facility Layout Problem of Manufacturing Unit Considering Human Factors. Mathematical Problems in Engineering, 2018, 1-13.

Lin, L. and Sharp, G. (1999). Quantitative and qualitative indices for the plant layout evaluation problem. European Journal of Operational Research, 116(1), 100–117.

Lin, Q. and Wang, D. (2019). Facility Layout Planning with SHELL and Fuzzy AHP Method Based on Human Reliability for Operating Theatre. Journal of Healthcare Engineering, 2019.

Lin, Y. and Lin, Y. (2015). Applying an immune ant colony system algorithm to solve an integrated flexible bay facility layout problem with input/output points design. Lecture Notes in Management Science, 7, 56–62.

Louveaux, F. and Birge, J. (1997). Introduction to Stochastic Programming. Springer, New York.

Margot, F. (2010). Symmetry in integer linear programming. In: Jünger, M., Liebling, T., Naddef, D., Nemhauser, G., Pulleyblank, W., Reinelt, G., Rinaldi, G., Wolsey, L. (eds.) 50 Years of Integer Programming 1958-2008. Springer, Heidelberg.

Margot, F. Symmetry in integer linear programming. In Jünger, M., Liebling, T. M., Naddef, D., Nemhauser, G. L., Pulleyblank, W. R., Reinelt, G., Rinaldi, G. and Wolsey, L. A. (ed. 50), 50 Years of Integer Programming 1958–2008. Springer, Berlin. 2010

Matai, R. (2015). Solving multi objective facility layout problem by modified simulated annealing. Applied Mathematics of Computation, 261, 302–311.

Matai, R., Singh, S. and Mittal, M. (2013). Modified simulated annealing based approach for multi objective facility layout problem. International Journal of Production Research, 51(14), 4273–4288.

Mazinani, M., Abedzadeh, M. and Mohebali, N. (2013). Dynamic facility layout problem based on flexible bay structure and solving by genetic algorithm. International Journal of Advanced Manufacturing Technology, 65(5–8), 929–943.

McKendall, A. and Hakobyan, A. (2010). Heuristics for the dynamic facility layout problem with unequal-area departments. European Journal of Operational Research, 201(1), 171–182.

McKendall, A. and Liu, W. (2012). New Tabu search heuristics for the dynamic facility layout problem. International Journal of Production Research, 50(3), 867–878.

Meller R. and Gau K. (1996). The facility layout problem: recent and emerging trends and perspectives, Journal of Manufacturing Systems, 15, 351-366.

Meller, R., Kirkizoglu, Z. and Chen, W. (2010). A New Optimization Model to Support a Bottom-up Approach to Facility Design. Computers & Operations Research, 37(1), 42-49.

Meller, R., Narayanan, V. and Vance, P. (1998). Optimal facility layout design. Operations Research Letters. 23(3-5), 117-127.

Merrell, P., Schkuzfza, E. and Koltun, V. (2010). Computer-Generated Residential Building Layouts. ACM Transactions on Graphics. 29(6), 1-12.

Michalek, J. and Papalambros, P. (2002). Interactive design optimization of architectural layouts. Engineering Optimization. 34(5), 484-501.

Michalek, J., Choudhary, R. and Papalambros, P. (2002). Architectural layout design optimization. Engineering Optimization. 343(5), 461-484.

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

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

Montreuil, B., Venkatadrim U. and Ratliff, D. (1989). Generating a layout from a design skeleton. IIE Transactions. 25(1), 3-15.

Moslemipour, G. and Lee, T. (2012). Intelligent design of a dynamic machine layout in uncertain environment of flexible manufacturing systems. Journal of Intelligent Manufacturing, 23(5), 1849–1860.

Murray, C., Smith, A. and Zhang, Z. (2013). An efficient local search heuristic for the double row layout problem with asymmetric material flow. International Journal of Production Research, 51(20), 6129–6139.

Murray, C., Zuo, X. and Smith, A. (2012).. An extended double row layout problem. proceedings of the 12th International Material Handling Research Colloquium, 535–550.

National Restaurant Association (2021). 2021 State of the Industry. https://shop.restaurant.org/2021-State-of-the-Industry. Accessed 15 September 2021.

Navidi, H., Bashiri, M. and Bidgoli, M. (2012). A heuristic approach on the facility layout problem based on game theory. International Journal of Production Research, 50(6), 1512–1527.

Neghabi, H. and Tari, F. (2016). A new concept of adjacency for concurrent consideration of economic and safety aspects in design of facility layout problems. Journal of Loss Prevention in the Process Industries. 40, 603-614.

Neghabi, H., Eshghi, K. and Salmani, M. (2014). A New Model for Robust Facility Layout Problem. Information Sciences, 278, 498–509.

Niroomand, S., Hadi-Vencheh, A., Şahin, R. and Vizvári, B. (2015). Modified migrating birds optimization algorithm for closed loop layout with exact distances in flexible manufacturing systems. Expert Systems with Applications, 42(19), 6586–6597.

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

Ou-Yang, C. and Utamima, A. (2013). Hybrid Estimation of Distribution Algorithm for solving Single Row Facility Layout Problem. Computers & Industrial Engineering, 66(1), 95–103.

Paes, F., Pessoa, A. and Vidal, T. (2017). A hybrid genetic algorithm with decomposition phases for the Unequal Area Facility Layout Problem. European Journal of Operational Research, 256(3), 742–756.

Palomo-Romero, J., Salas-Morera, L. and García-Hernández, L. (2017). An island model genetic algorithm for unequal area facility layout problems. Expert Systems with Applications, 68, 151–162.

Palubeckis, G. (2012). A branch-and-bound algorithm for the single-row equidistant facility layout problem. OR Spectrum, 34(1), 1–21.

Palubeckis, G. (2015). Fast local search for single row facility layout. European Journal of Operational Research, 246(3), 800–814.

Palubeckis, G. (2015). Fast simulated annealing for single-row equidistant facility layout. Applied Mathematics of Computation, 263, 287–301.

Palubeckis, G. (2017). Single row facility layout using multi-start simulated annealing. Computers & Industrial Engineering, 103, 1–16.

Park, K., Koo, J., Shin, D., Lee, C. and Yoon, E. (2011). Optimal Multi-Floor Plant Layout with Consideration of Safety Distance Based on Mathematical Programming and Modified Consequence Analysis. Korean Journal of Chemical Engineering, 28(4), 1009–1018.

Peng, C., Yang, Y., Bao, F., Fink, D., Yan, D., Wonka, P. and Mitra, N. (2016). Computational network design from functional specifications. ACM Transactions on Graphics, 35(4), 1-12.

Peng, Y, Zeng, T., Fan, L., Han, Y. and Xia, B. (2018). An Improved Genetic Algorithm Based Robust Approach for Stochastic Dynamic Facility Layout Problem. Discrete Dynamics in Nature and Society, 1-8, 2018.

Pillai, V., Hunagund, I. and Krishnan, K. (2011). Design of robust layout for Dynamic Plant Layout Problems. Computers & Industrial Engineering, 61(3), 813–823.

Pourvaziri, H, Pierreval, H. and Marian, H. (2021) Integrating facility layout design and aisle structure in manufacturing systems: Formulation and exact solution. European Journal of Operational Research, 290, 499-513.

Pourvaziri, H. and Naderi, B. (2014). A hybrid multi-population genetic algorithm for the dynamic facility layout problem. Applied Soft Computing Journal, 24, 457–469.

Pourvaziri, H. and Pierreval, H. (2017). Dynamic facility layout problem based on open queuing network theory. European Journal of Operational Research, 259(2), 538–553.

Pourvaziri, H. Pierrerval, H. and Marian, H. (2021). Integrating facility layout design and aisle structure in manufacturing systems: Formulation and exact solution. European Journal of Operational Research, 290(2), 499-513.

Python Software Foundation. Python 3.7.7 documentation. Retrieved from https://docs.python.org/3.7/.

Raman, A., Roy, D. (2015). An analytical modeling framework for determining the optimal table mix in restaurants. IFAC-PapersOnLine. 48(3), 225-230.

Restaurants and bars during the Covid-19 pandemic. Centers for Disease and Control. https://www.cdc.gov/coronavirus/2019-ncov/downloads/community/restaurants-and-bars-decision-tree.pdf. Accessed 25 May 2020.

Río-Cidoncha, G., Palacios, J. and Iglesias, J. (2007). A multidisciplinary model for floorplan design. International Journal of Production Research. 45, 3457-3476.

Ripon, K., Glette, K., Khan, K., Hovin, M. and Torresen, J. (2013). Adaptive variable neighborhood search for solving multi-objective facility layout problems with unequal area facilities. Swarm and Evolutionary Computation, 8, 1–12.

Rodrigues, E., Gaspar, A. and Gomes, Á. (2013). An approach to the multi-level space allocation problem in architecture using a hybrid evolutionary technique. Automation in Construction, 35, 482–498.

Ruch, J. (1978). Interactive space layout: a graph theoretical approach. Proceedings of the 15th Design Automation Conference. 152-157.

Saaty, T. (1994). How to Make a Decision: The Analytic Hierarchy Process. Interfaces, 24(6), 19–43

Sadrzadeh, A. (2012). A genetic algorithm with the heuristic procedure to solve the multi-line layout problem. Computers & Industrial Engineering, 62(4), 1055–1064.

Sagnak, M., Ada, E. and Kazancoglu, Y. (2019). A new holistic conceptual framework for layout performance assessment. Journal of Manufacturing Technology Management, 30(1), 233–260.

Şahin, R., Ertoral, K. and Türkbey, O. (2010). A simulated annealing heuristic for the dynamic layout problem with budget constraint. Computers & Industrial Engineering, 59(2), 308–313.

Sahinidis, N. (2004). Optimization under uncertainty: state-of-the-art opportunities. Computers & Chemical Engineering, 28, 971-983.

Sahni, S. and Gonzales, T. (1976). P-Complete Approximation Problems. Journal of the Association for Computing Machinery. 23(2), 555-565.

Samarghandi, H. and Eshghi, K. (2010). An efficient tabu algorithm for solving the single row facility layout problem. European Journal of Operational Research, 205(1), 98–105.

Samarghandi, H., Taabayan, P. and Jahantigh, F. (2010). A particle swarm optimization for the single row facility layout problem. Computers & Industrial Engineering, 58(4), 529–534.

Scholz, D., Jaehn, F. and Junker, A. (2010). Extensions to STaTS for practical applications of the facility layout problem. European Journal of Operational Research, 204(3), 437–472.

Secchin, L and Amaral, A. (2019). An improved mixed-integer programming model for the double row layout of facilities. Optimization Letters, 13, 193–199.

Seehof, J., Evans, W., Friederichs. J. and Quigley, J. (1966). Automated facilities Layout Programs. Proceedings of the 1966 21st National Conference. 191-199.

Sherali, H., Fraticelli, B. and Meller, R. (2003). Enhanced model formulations for optimal facility layout. Operations Research, 51(4), 629–644.

Simmons, D. (1969). One-Dimensional Space Allocation: An Ordering Algorithm. Operations Research, 17(5), 812-826.

Singh, S. and Singh, V. (2011). Three-level AHP-based heuristic approach for a multi-objective facility layout problem. International Journal of Production Research, 49(4), 1105–1125.

Singh, S., Sharma, R. (2005). A review of different approaches to the facility layout problems. International Journal of Advanced Manufacturing Technology. 30, 425–433.

Tanchoco, J and Sinreich, D. (1992). OSL-optimal single loop guide paths for AGVS. International Journal of Production Research, 30, 665–681.

Tari, F. and Neghabi, H. (2015). A new linear adjacency approach for facility layout problem with unequal area departments. Journal of Manufacturing Systems. 37(1), 93-103.

Tayal, A. and Singh, S. (2018). Integrating big data analytic and hybrid firefly-chaotic simulated annealing approach for facility layout problem. Annals of Operations Research, 270,(1–2), 489–514.

Tayal, A., Kose, U., Solanki, A., Nayyar, A. and Saucedo, J. (2020). Efficiency analysis for stochastic dynamic facility layout problem using meta-heuristic, data envelopment analysis and machine learning. Computational Intelligence, 36(1), 172–202.

Tesla. Return to Work Playbook. https://www.tesla.com/sites/default/files/blog_attachments/Tesla-Return-to-Work-Playbook.pdf?redirect=no. Accessed 20 May 2020.

Thompson, G. (2002). Optimizing a restaurant's seating capacity: Use dedicated or combinable tables? Cornell Hotel and Restaurant Administration Quarterly 43(4), 48-57.

Tosun, U. (2015). On the performance of parallel hybrid algorithms for the solution of the quadratic assignment problem. Engineering Applications of Artificial Intelligence, 39, 267–278.

Tosun, U., Dokeroglu, T. and Cosar, A. (2013). A robust Island Parallel Genetic Algorithm for the Quadratic Assignment Problem. International Journal of Production Research, 51(14), 4117–4133.

Tubaileh, A. and Siam, J. (2017). Single and multi-row layout design for flexible manufacturing systems. International Journal of Computer Integrated Manufacturing, 30(12), 1316–1330.

Ugail, H., et al. (2021). Social distancing enhanced automated optimal design of physical spaces in the wake of the COVID-19 pandemic. Sustainable Cities and Society. 68, id 102791.

Ugurlu, C., Chatzikonstantinou, I., Sariyildiz, S., Tasgetiren, M. (2015). Evolutionary computation for architectural design of restaurant layouts. IEEE – Congress on Evolutionary Computation. 2279–2286.

Ulutas, B. and Islier, A. (2015). Dynamic facility layout problem in footwear industry. Journal of Manufacturing Systems, 36, 55–61.

Ulutas, B. and 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.

Vázquez-Román, R., Díaz-Ovalle, C., Jung, S. and Castillo-Borja, F. (2019). A Reformulated Nonlinear Model to Solve the Facility Layout Problem. Chemical Engineering Communications, 206(4), 476–487.

Verma, M. and Thakur, M. (2010). Architectural Space Planning using Genetic Algorithms. The 2nd International Conference on Computer and Automation Engineering. 268-275.

Vitayasak, S., Pongcharoen, P. and Hicks, C. (2017). A tool for solving stochastic dynamic facility layout problems with stochastic demand using either a Genetic Algorithm or modified Backtracking Search Algorithm. International Journal of Production Economics., 190, 146–157.

Vollmann, T. and Buffa, E. (1966). The Facilities Layout Problem in Perspective. Management Science, 12(10), B450-B468.

Wang, P., Zhu, Z. and Wang, Y. (2016). A novel hybrid MCDM model combining the SAW, TOPSIS and GRA methods based on experimental design. Information Sciences, 345, 27–45.

Wang, S., Zuo, X., Liu, X., Zhao, X. and Li, J. (2015). Solving dynamic double row layout problem via combining simulated annealing and mathematical programming. Applied Soft Computing Journal, 37, 303–310.

Wang, S., Zuo, X., Liu, X, Zhao, X. and Li, J. (2015). Solving dynamic double row layout problem via combined simulated annealing and mathematical programming. Applied Soft Computing, 37, 303–310.

Wong, M. et al. (2019). Design strategies for biocontainment units to reduce risk during doffing of high-level personal protective equipment. Clinical Infectious Diseases. 69 (Supplement_3). S241-S247.

Wu, W., Fan, L., Liu, L. and Wonka, P. (2018). MIQP-Based Layout Design for Building Interiors. Computer Graphics Forum, 37(2), 511–521.

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

Xiao, Y., Seo, Y. and Seo, M. (2013). A two-step heuristic algorithm for layout design of unequal-sized facilities with input/output points. International Journal of Production Research, 51(14), 4200–4222.

Xiao, Y., Xie, Y., Kulturel-Konak, S. and Konak, A. (2017). A problem evolution algorithm with linear programming for the dynamic facility layout problem—A general layout formulation. Computers & Operations Research, 88 , 187–207.

Xie, Y., Zhou, S., Xiao, Y., Kulturel-Konak, S. and Konak, A. (2018). A β-Accurate Linearization Method of Euclidean Distance for the Facility Layout Problem with Heterogeneous Distance Metrics. European Journal of Operational Research, 265(1), 26–38.

Yang, C., Chuang, S. and Hsu, T. (2011). A genetic algorithm for dynamic facility planning in job shop manufacturing. International Journal of Advanced Manufacturing Technology, 52(1–4), 303–309.

Yang, T. and Kuo, C. (2003). A hierarchical AHP/DEA methodology for the facilities layout design problem. European Journal of Operational Research, 147(1), 128–136.

Yang, X., Cheng, W., Smith, A. and Amaral, A. (2020). An improved model for the parallel row ordering problem. Journal of the Operational Research Society, 71(3), 475–490.

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

Yu-Hsin Chen, G. (2013). A new data structure of solution representation in hybrid ant colony optimization for large dynamic facility layout problems. International Journal of Production Economics., vol. 142, no. 2, pp. 362–371, 2013.

Zawidski, M., Tateyama, K. and Nisikawa, I. (2010). The constraints satisfaction problem approach in the design of an architectural functional layout. Engineering Optimization. 43, 943-966.

Zhang, Z. and Murray, C. (2012). A corrected formulation for the double row layout problem. International Journal of Production Research, 50(15), 4220–4223.

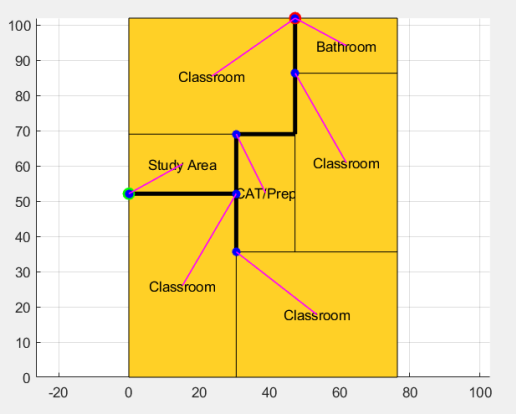
Zimring, C. et al. (2018) Making the invisible visible: Why does design matter for safe doffing of personal protection equipment?. Infection Control & Hospital Epidemiology. 39(11), 1375-1377.

Zouein, P., Harmanani, H. and Hajar, A. (2002). Genetic algorithm for solving site layout problem with unequal-size and constrained facilities. Journal of Computing in Civil Engineering, 16(2), 143-151.

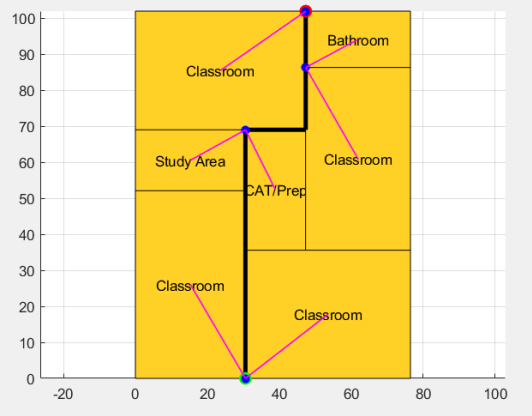
Zuo, X., Murray, C. and Smith, A. (2014). Solving an extended double row layout problem using multiobjective tabu search and linear programming. IEEE Transactions on Automation Science and Engineering. 11(4), 1122–1132.

Zuo, X., Murray, C. and Smith, A. (2016). Sharing clearances to improve machine layout. International Journal of Production Research. 54(14), 4272–4285.

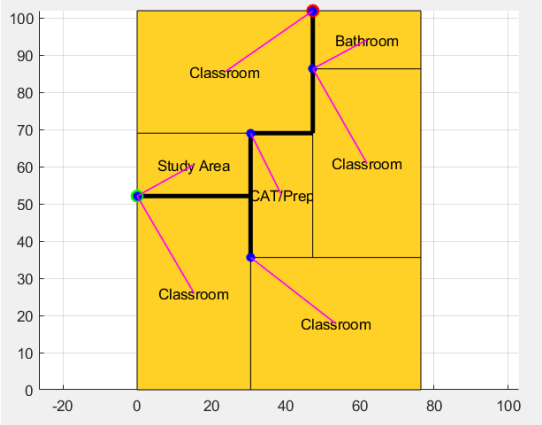
APPENDIX A: LAYOUT EVALUATIONS



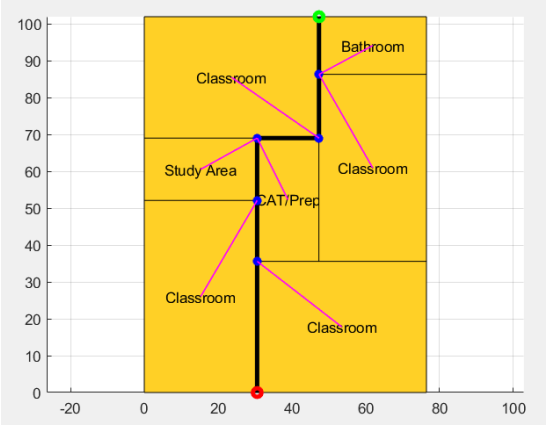
Layout 1
Distance between accessways: 68.79 ft.
Number of turns along egress paths: 3
Aisle network length: 113.78 ft.
Score: 4.5



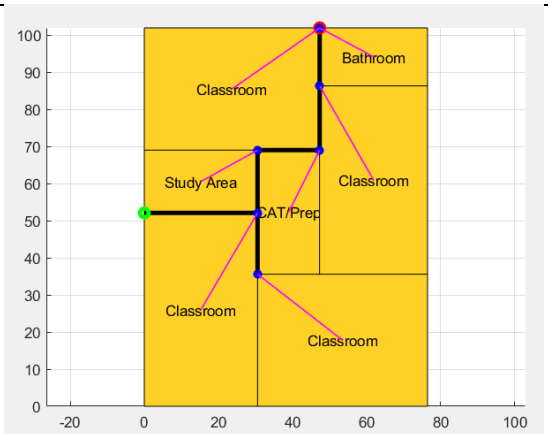
Layout 2
Distance between accessways: 103.37 ft.
Number of turns along egress paths: 2
Aisle network length: 118.77 ft.
Score: 8



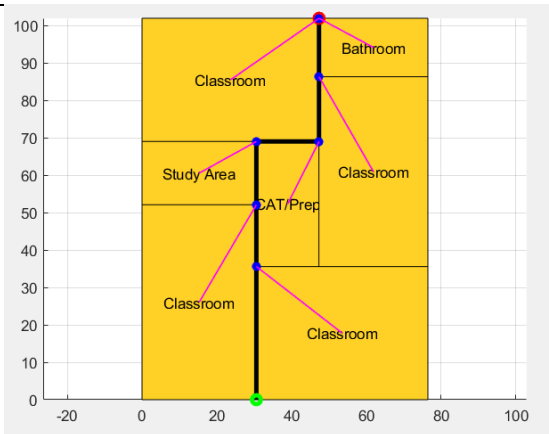
Layout 3
Distance between accessways: 68.79 ft.
Number of turns along egress paths: 3
Aisle network length: 113.78 ft.
Score: 7



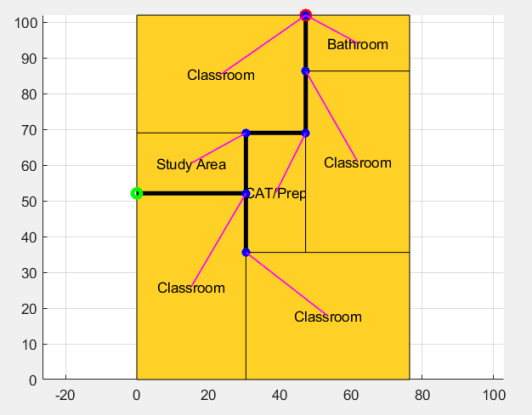
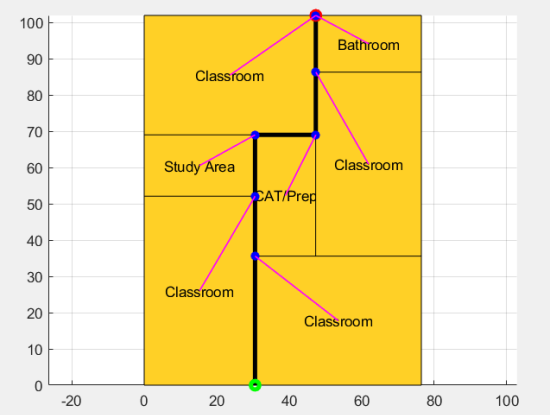
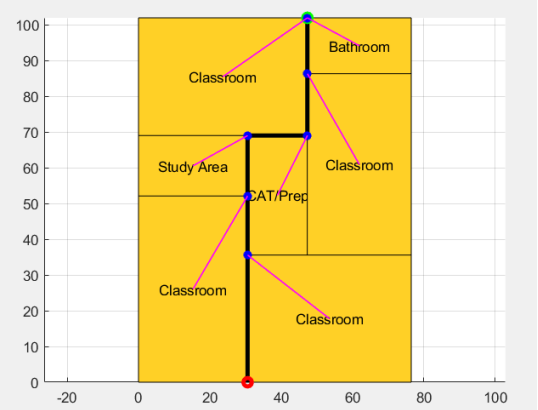
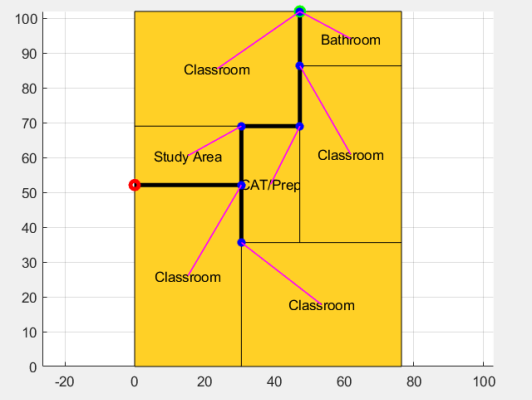
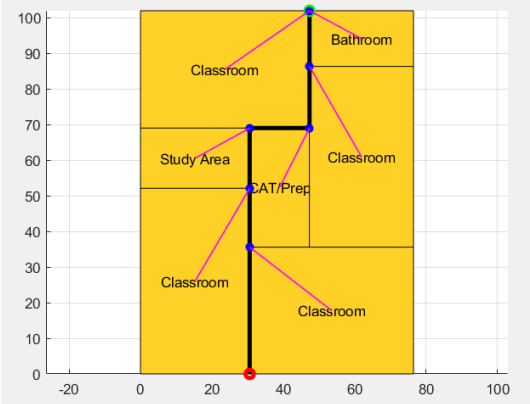
Layout 4
Distance between accessways: 103.37 ft.
Number of turns along egress paths: 2
Aisle network length: 118.77 ft.
Score: 10



Layout 5



Layout 6

Distance between accessways: 68.79 ft. Number of turns along egress paths: 3 Aisle network length: 113.78 ft. Score: 5	Distance between accessways: 103.37 ft. Number of turns along egress paths: 2 Aisle network length: 118.77 ft. Score: 8
 Layout 7 Distance between accessways: 68.79 ft. Number of turns along egress paths: 3 Aisle network length: 113.78 ft. Score: 5	 Layout 8 Distance between accessways: 103.37 ft. Number of turns along egress paths: 2 Aisle network length: 118.77 ft. Score: 8
 Layout 9 Distance between accessways: 68.79 ft. Number of turns along egress paths: 3 Aisle network length: 113.78 ft. Score: 4.5	 Layout 10 Distance between accessways: 103.37 ft. Number of turns along egress paths: 2 Aisle network length: 118.77 ft. Score: 8
 Layout 11	 Layout 12

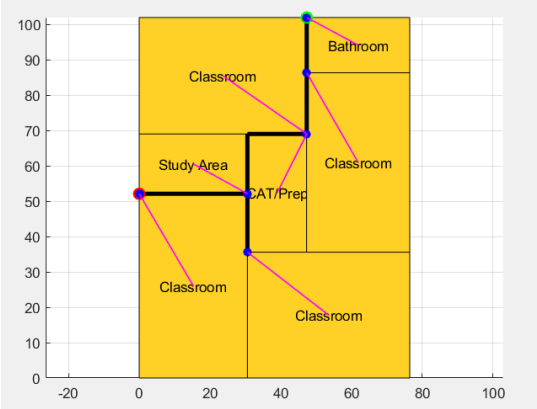
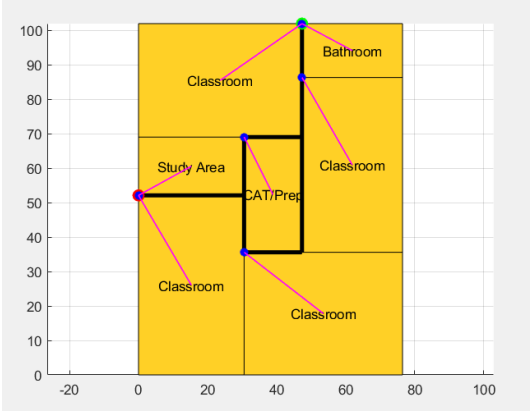
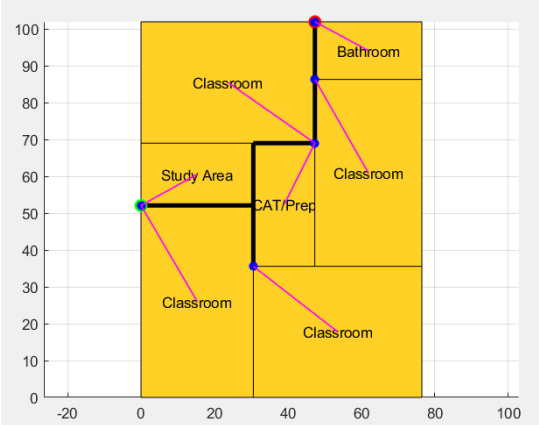
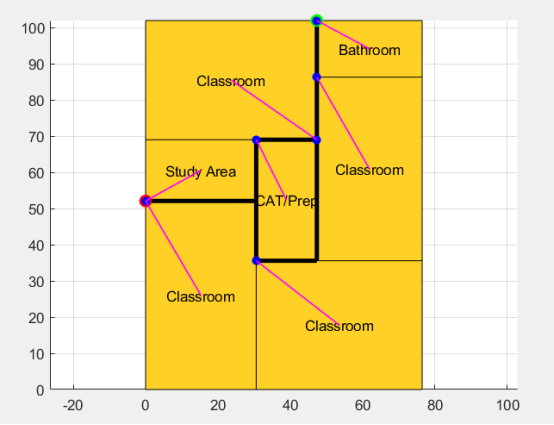
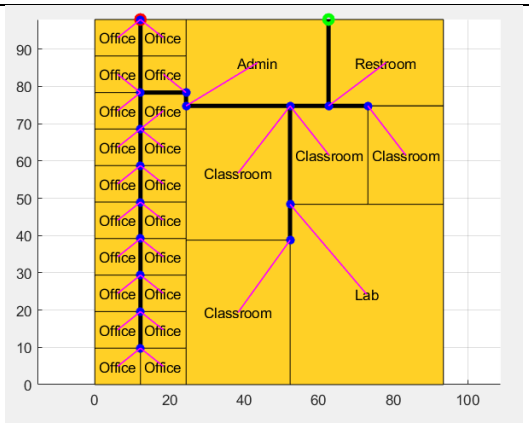
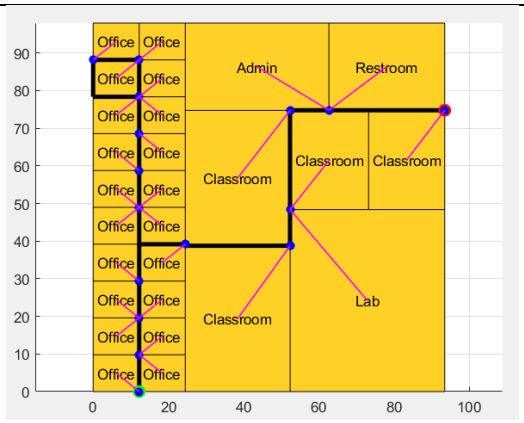
Distance between accessways: 68.79 ft. Number of turns along egress paths: 3 Aisle network length: 113.78 ft. Score: 4.5	Distance between accessways: 103.37 ft. Number of turns along egress paths: 2 Aisle network length: 118.77 ft. Score: 8
 Layout 13 Distance between accessways: 68.79 ft. Number of turns along egress paths: 3 Aisle network length: 113.78 ft. Score: 7	 Layout 14 Distance between accessways: 68.79 ft. Number of turns along egress paths: 4 Aisle network length: 164 ft. Score: 3
 Layout 15 Distance between accessways: 68.79 ft. Number of turns along egress paths: 3 Aisle network length: 113.78 ft. Score: 7	 Layout 16 Distance between accessways: 68.79 ft. Number of turns along egress paths: 4 Aisle network length: 164 ft. Score: 4

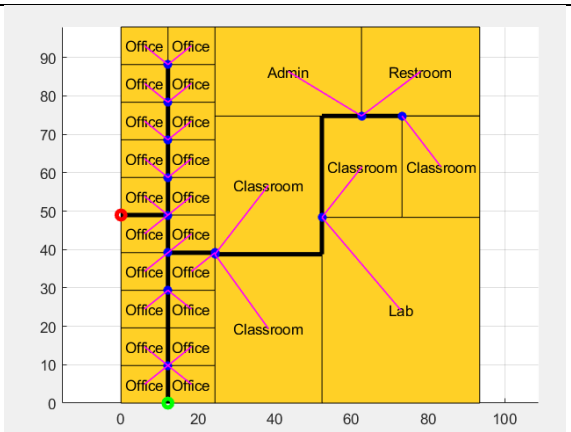
Figure A1: Small-scale layout alternatives from the experimental design



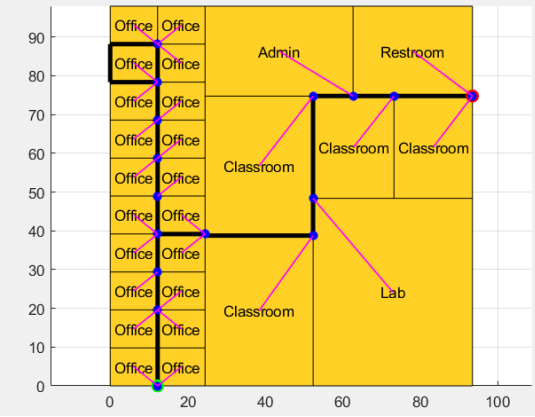
Layout 1
Distance between accessways: 50.45 ft.
Number of turns along egress paths: 5
Aisle network length: 211.91 ft.
Score: 3



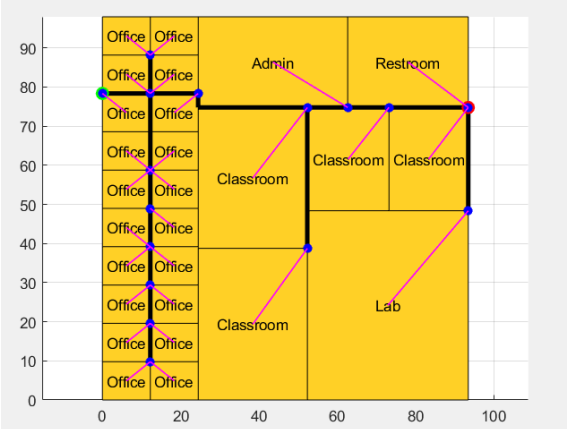
Layout 2
Distance between accessways: 110.42 ft.
Number of turns along egress paths: 7
Aisle network length: 240.13 ft.
Score: 8



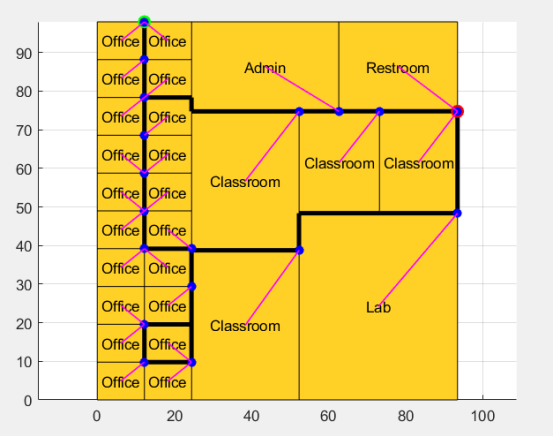
Layout 3
Distance between accessways: 50.48 ft.
Number of turns along egress paths: 6
Aisle network length: 197.82 ft.
Score: 2



Layout 4
Distance between accessways: 110.42 ft.
Number of turns along egress paths: 7
Aisle network length: 240.13 ft.
Score: 7



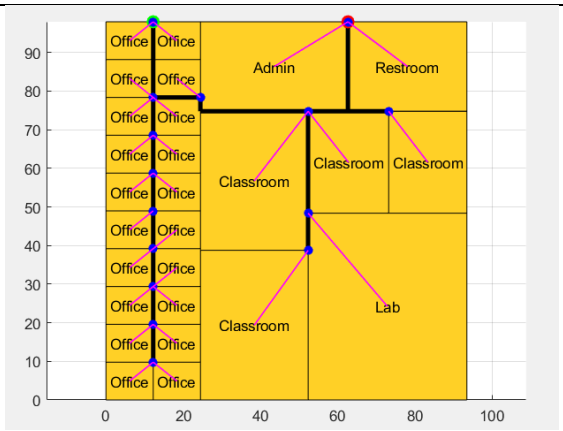
Layout 5



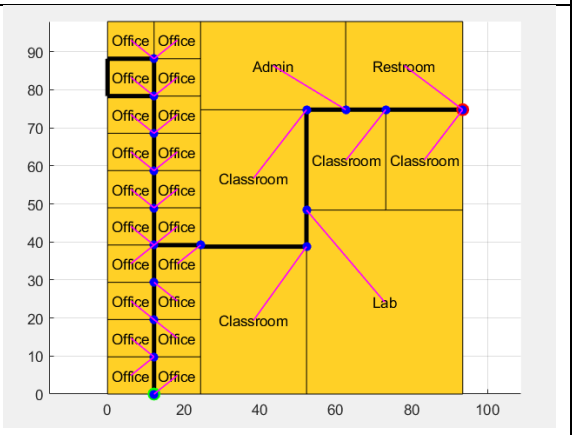
Layout 6

Distance between accessways: 93.56 ft.
Number of turns along egress paths: 5
Aisle network length: 237.85 ft.
Score: 2

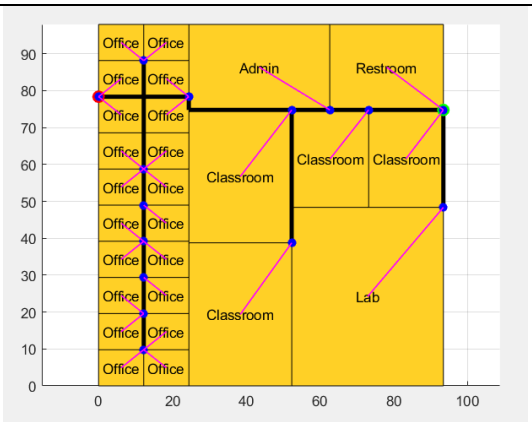
Distance between accessways: 84.49 ft.
Number of turns along egress paths: 11
Aisle network length: 324.6 ft.
Score: 6



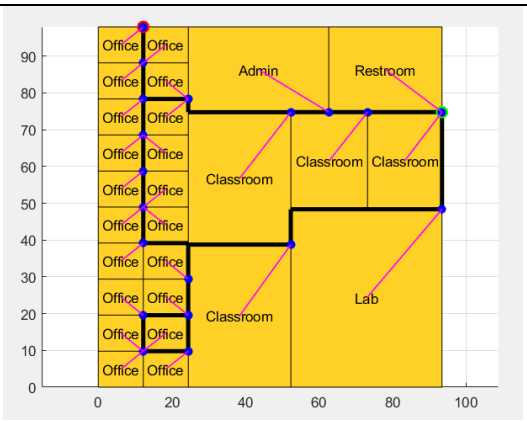
Layout 7
Distance between accessways: 50.45 ft.
Number of turns along egress paths: 5
Aisle network length: 211.91 ft.
Score: 2



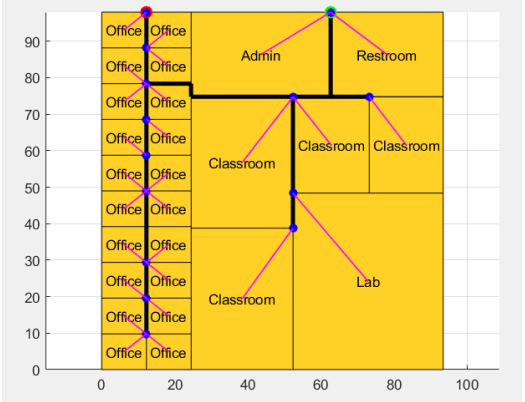
Layout 8
Distance between accessways: 110.42 ft.
Number of turns along egress paths: 7
Aisle network length: 240.13 ft.
Score: 7



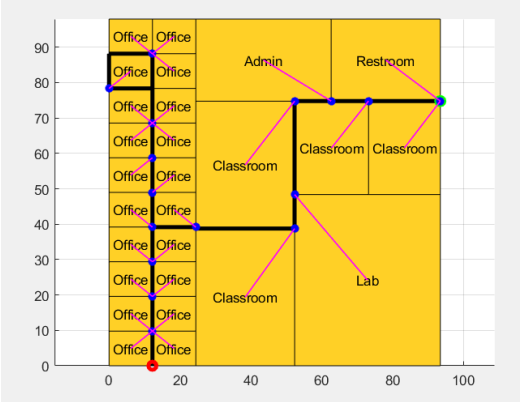
Layout 9
Distance between accessways: 93.57 ft.
Number of turns along egress paths: 5
Aisle network length: 237.85 ft.
Score: 3



Layout 10
Distance between accessways: 84.49 ft.
Number of turns along egress paths: 11
Aisle network length: 324.6 ft.
Score: 5



Layout 11



Layout 12

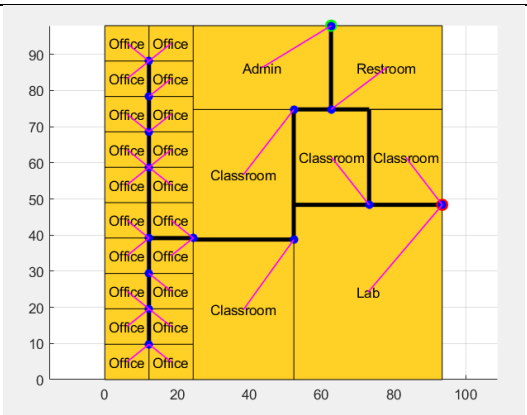
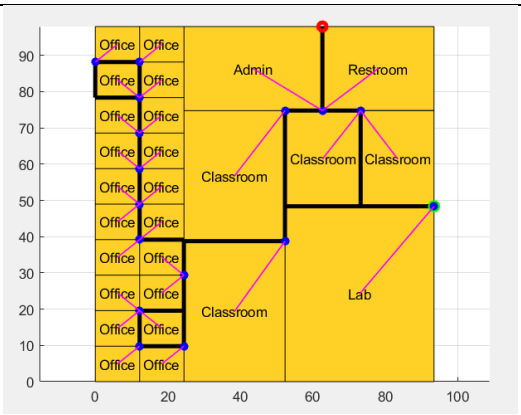
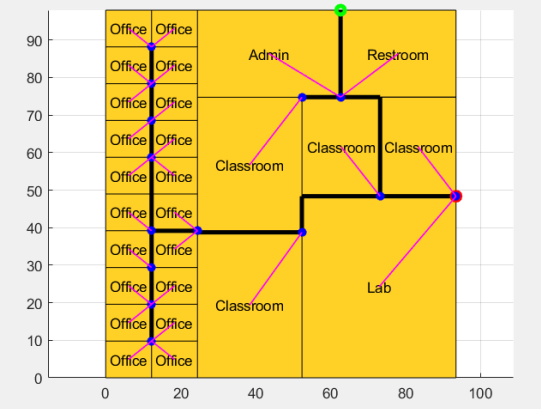
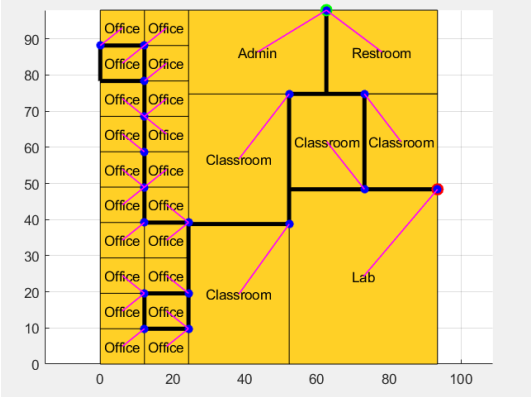
Distance between accessways: 50.45 ft. Number of turns along egress paths: 5 Aisle network length: 211.91 ft. Score: 3	Distance between accessways: 110.42 ft. Number of turns along egress paths: 7 Aisle network length: 240.13 ft. Score: 7
 Layout 13 Distance between accessways: 58.36 ft. Number of turns along egress paths: 8 Aisle network length: 266.49 ft. Score: 3	 Layout 14 Distance between accessways: 58.36 ft. Number of turns along egress paths: 13 Aisle network length: 334.62 ft. Score: 5
 Layout 15 Distance between accessways: 58.36 ft. Number of turns along egress paths: 8 Aisle network length: 240.12 ft. Score: 1	 Layout 16 Distance between accessways: 58.36 ft. Number of turns along egress paths: 13 Aisle network length: 334.62 ft. Score: 4

Figure A2: Large-scale layout alternatives from the experimental design