A Three-Stage Mathematical-Programming Method for the Multi-Floor Facility Layout Problem

by

Sabrina Bernardi

A research paper
presented to the University of Waterloo
in partial fulfillment of the requirements for the degree of
Master of Mathematics
in
Computational Mathematics

Supervisor: Dr. Miguel Anjos

Waterloo, Ontario, Canada, 2010

I hereby declare that I am the sole author of this report. This is a true copy of the report, including any required final revisions, as accepted by my examiners.

I understand that my report may be made electronically available to the public.

Abstract

The purpose of this research paper is to present a three-stage method using mathematical-programming techniques that finds high-quality solutions to the multi-floor facility layout problem. The first stage is a linear mixed-integer program that assigns departments to floors such that the total of the departmental interaction costs between floors is globally minimized. Subsequent stages find a locally optimal layout for each floor. Two versions of the proposed approach are considered. The first solves the layout of each floor independently of the other floors, allowing up to one elevator location. The second solves the layout of all floors simultaneously, allowing for multiple elevator locations. Variations to the problem and to the basic method are also investigated. The two versions are tested and compared to each other through computational experiments and also to existing results in the literature. It is clear that the proposed method can provide several high-quality layouts for medium and large-scale problem instances. Not only does it achieve competitive results compared to previous methods, but it also overcomes some of their limitations.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

In general, *facility layout problems* involve finding the optimal arrangement of departments within a facility. Interaction costs between departments of given areas are minimized in the optimal arrangement. Many applications of this general problem exist and include arranging departments in production facilities, in hotels, in office buildings, and in hospitals, to name a few. There are several variations to the problem, all of which are $\mathcal{NP}$-hard [3]. Even the *quadratic assignment problem*, which is the special case of assigning N departments to N fixed locations with departments of fixed, equal shapes is $\mathcal{NP}$-hard [12].

A particular case of the general facility layout problem is the *multi-floor facility layout problem* which, as the name suggests, involves finding the optimal arrangement of departments in a facility having multiple floors. More constraints arise in the multi-floor problem in addition to those already present in the single floor case; this adds to the complexity of the problem. Not only must the interaction between departments on the same floor be considered, but also the interaction between departments that are on different floors of the facility. This requires the use and placement of elevators and/or stairwells to facilitate the movement of material between floors. Due to the complexity of the problem, many multi-floor approaches have several limitations. These may include the inability to accommodate multiple elevator locations, the need to split departments across floors, and computational times that are too high for practical use.

In general, the objective function for the multi-floor facility layout problem can be given as

$$\min \sum_i \sum_j (c_{ij}^H d_{ij}^H + c_{ij}^V d_{ij}^V) f_{ij},$$

where f_{ij} is a parameter denoting the flow between departments i and j, c_{ij}^H (c_{ij}^V) is a parameter denoting the horizontal (vertical) cost per unit distance between departments

i and j, and d_{ij}^H (d_{ij}^V) is a variable denoting the horizontal (vertical) distance between departments i and j [23]. It is assumed here that the material is transported between departments on different floors using the elevator that minimizes the distance between the two departments. In other words, if departments i and j are located on different floors, $d_{ij}^H = \min_e (d_{ie} + d_{ej})$, where d_{ie} is the distance between department i and elevator e and d_{ej} is the distance between elevator e and department j [23]. The costs and flows are given by the user and the position of the departments within the facility is determined in the optimal layout.

There are many different forms of this objective function, taking into account other important aspects of facility layouts. In certain applications, information on corridors, multiple elevator locations or stairwells, and even their capacities are helpful or even required. Given the complexity of these problems, exact solutions may be difficult to find and global optimal algorithms work, in general, only for small problem cases. Heuristics are often needed for larger, more complex problems. The latter is the approach taken here.

In this research paper a three-stage method is presented that uses mathematical-programming techniques to provide good solutions to the multi-floor facility layout problem. In particular, this method extends the framework for the single-floor facility-layout problem by Anjos and Vannelli [3] to the multi-floor case. The first stage is a linear mixed-integer program equivalent to FAF, which was introduced by Meller and Bozer [23] to assign departments to floors while minimizing vertical interaction costs between departments. Each department remains fixed to the floor it is assigned in the first stage. Subsequent stages find a locally optimal layout for each floor using an approach based on the single-floor framework of Anjos and Vannelli [3]. This new multi-floor model was implemented and solved using the CPLEX solver for the first stage and MINOS for the remaining stages through the GAMS modeling language. Variations to the problem and to the basic method are also investigated.

Two versions of the problem are considered. The first solves the layout for each floor independently of the other floors. One consequence is that no more than one elevator location can be considered. The second version solves the layouts on all floors simultaneously, allowing for multiple elevator locations. These versions are compared to each other through computational experiments and also to existing methods in the literature. It is clear that both versions can provide several high-quality layouts even for large problem instances.

The report is structured as follows. Chapter 2 is a literature review of the methods for solving the multi-floor facility layout problem. Chapter 3 gives the background necessary for the proposed three-stage method. Chapter 4 presents both versions of the three-stage multi-floor layout model and the results of computational experiments can be found in Chapter 5. The conclusions and specifics of the problem data constitute the remainder of the report.

Chapter 2

Literature Review

This chapter outlines several methods in the literature used to solve the multi-floor facility layout problem and its variations. Advantages and disadvantages, limitations and strengths, as well as the quality of their results are summarized.

2.1 Single-Stage Approaches

2.1.1 Exchange-Based Heuristics

Exchange-based heuristics, in general, begin with an initial layout and exchange departments within/across floors in order to find a lower cost layout. Some early heuristics place restrictions on the departments that can be exchanged and others involve splitting departments in the final layout. This is not acceptable for many practical applications and more recent heuristics improve upon these limitations.

CRAFT

CRAFT [6] is a single-floor improvement-type heuristic that influenced subsequent methods for solving multi-floor facility layout problems. Using a steepest descent approach, CRAFT begins with an initial layout and exchanges the locations of two or three departments that are either adjacent or equal in area. The effect of every possible exchange on the material-handling cost is recorded and the exchange which will most reduce the cost is selected. This process is repeated and is terminated when no exchange that reduces the objective function value can be found.

As a result of using the steepest descent approach, it is possible that CRAFT will arrive at a solution that is a local minimum rather than the global minimum. Since there is likely more than one local minimum, the final solution can vary depending on the initial solution and the path taken, i.e., the exchanges that are made [24].

SPACECRAFT

Presented in 1982 by Johnson [13], SPACECRAFT is a method influenced by CRAFT for solving the multi-floor facility layout problem. It was the first method of arranging departments in a multi-floor building known to Johnson at the time. The procedure itself begins with an initial layout and attaches the separate floors to each other in a two-dimensional layout grid, before attempting to improve the solution iteratively. An improved solution is obtained by exchanging the two or three departments which will result in the greatest savings. Similar to CRAFT, these departments must either be adjacent pairs or triplets in the layout and/or department pairs of equal size. The procedure repeats until no improved solution can be found by performing these exchanges or until the procedure has reached its maximum number of iterations allowed. SPACECRAFT allows for elevator and stairwell locations in any area of the building. However, due to the way in which SPACECRAFT evaluates its exchanges, departments may be split across floors; the floors are attached to each other in a two-dimensional layout grid, the exchanges are made, and then it is transformed back into multiple floors [5].

MULTIPLE

The next improvement type algorithm, which is also an extension of CRAFT, overcomes some of the above limitations. Bozer et al.[5] present MULTIPLE which stands for Multi-Floor Plant Layout Evaluation. This algorithm uses spacefilling curves and a two-dimensional layout grid to represent the layout of each floor. The area that each department will occupy is known and is represented by the number of grid squares it occupies within this grid.

Using a similar example to the one given in [5], with department areas given by the number of grid squares in Table 2.1 and layout sequence $1-2-3$, one can see that on

Table 2.1: Department Areas

Department Number	1	2	3
Number of Grid Squares	5	11	4

the layout for the floor, the first 5 grid squares following the path of the spacefilling curve belong to department 1, the next 11 belong to department 2, and the next 4 belong to department 3. The path of the spacefilling curve passes through every usable grid square that is not allocated to a fixed department.

MULTIPLE begins with an initial layout and considers all exchanges that are area-feasible between any two departments located on the same floor or across different floors, even if they are not adjacent or equal in size. In each iteration, the algorithm then selects the best feasible exchange, the one which minimizes the cost, and repeats this process with the new layout. When no exchanges that improve the layout can be found, the process terminates.

Within a floor, the exchange is straight-forward; the layout sequence is simply rearranged and the grid squares are assigned to departments as before. The order in which the grid squares are assigned to a department follows the path of the constructed spacefilling curve, which is a continuous function, so the departments will never be split on the same floor and the department shapes will not worsen with each iteration. The areas assigned to each department consist of a range of acceptable values rather than a specific number. This, along with the fact that there is a separate spacefilling curve for each floor, will increase the number of exchanges that can be made within and across floors without splitting departments.

Software: LayOPT

LayOPT [11] is a software for use in Windows that can find optimal solutions to single and multiple floor layout problems. The algorithm used is based on the one used in [5]. LayOPT is able to run the optimization algorithm automatically or interactively. This software allows the user to specify constraints and spacefilling curves. It allows any department shape and the user may modify these shapes in order to better suit his/her purpose. The user can also specify the flows and the costs associated with them.

2.1.2 Simulated-Annealing Based Algorithms

The heuristics mentioned above are path-dependent whose final solutions may settle at a local minimum since they do not consider any departmental exchanges that might temporarily increase the value of the objective function. Simulated annealing is used in heuristics to attempt to reduce path dependency and should yield better solutions by reducing the bias associated with the initial layout and by removing some of the exchange restrictions [22].

SABLE

SABLE, introduced in [22] by Meller and Bozer, uses simulated annealing and spacefilling curves to solve the multi-floor facility layout problem. Similar to MULTIPLE, the layout can be represented with grids and can be uniquely defined by a sequence of numbers with dividers and a spacefilling curve.

SABLE begins with an annealing schedule of temperatures upon which the quality of the final solution is dependent. Each department is assigned an address that determines the initial layout and its cost. A number b between 0 and 1 is uniformly sampled for each department and is compared to a specified critical value β. If $b < \beta$, then a new department address is generated. Otherwise, the address remains as it is. The departments are re-sorted according to their new addresses to determine the new layout sequence and checked for feasibility. If not feasible, the process is repeated. Generating layouts in this way, the procedure does more than just exchange two or three departments as in previous layouts, but many exchanges can occur and may even change the number of departments on each floor. If the changed layout, the candidate layout, reduces the value of the objective function, it becomes the new current representation. If not, it is accepted with a certain probability. This allows the algorithm to visit layouts even if they are worse, with a certain probability, to overcome the problem of settling at a local minimum [22].

Experimental results conclude that, on average, SABLE outperforms MULTIPLE, especially in the case when the value of vertical cost per distance unit to horizontal cost per distance unit is high. This result makes sense given that SABLE is more flexible with departmental exchanges across floors and may even change the number of departments on each floor [22]. The largest of the test problems considered is a 40-department and 4-floor problem with an average running time of 305.3 seconds.

2.1.3 Genetic-Based Algorithms

Several genetic-based algorithms exist and are useful for including other important aspects of facility layout problems. Some present variations to the layout problem that may accommodate many practical problems. Several of these algorithms are presented here.

MULTI-HOPE

Kochhar and Heragu [16] introduce a genetic algorithm-based heuristic to solve the multi-floor layout problem called the Multi-Floor Heuristically Operated Placement Evolution (MULTI-HOPE) technique. Each floor is represented by a grid of unit squares. The number of unit squares assigned to each department corresponds to the area of each department.

The location of lifts are given in advance and are indicated on the grid. The lift with the lowest transportation cost is selected to transport materials. This algorithm does not allow departments to be split across floors.

Experimental results in [16] show that MULTI-HOPE resulted in better average final solutions in most of the tested cases than both MULTIPLE and SABLE, however, it does so with larger computational times. The largest of the test problems considered is also a 40-department and 4-floor problem.

MUSE

Matsuzaki, Irohara, and Yoshimoto [20] introduce MUSE (MUlti-Story layout algorithm with consideration of Elevator utilization), which is a heuristic, improvement-type algorithm that considers the capacity of elevators and optimizes their number and location. It is assumed that the area and shape of every floor and that the capacity of each elevator is equal. However, the areas of elevators and aisles are not considered. Vertical and horizontal material handling costs are included in the objective function as well as the installation costs of each elevator. They conclude that their proposed algorithm is effective by testing it on the 15-department, 3-floor, and 6-elevator problem that was used to test MULTIPLE in [5].

An Improved Genetic Algorithm for Multi-floor Facility Layout Problems Having Inner Structure Walls and Passages

Lee, Roh, and Jeong [17] present an improved genetic algorithm for multi-floor facility layout problems having inner structure walls and passages. The boundary of the facility can be a curve such as the boundary of a ship. It is assumed that the number and position of the inner structure walls and lifts are specified. Also specified are the number of passages, their widths and the bounds of their locations. Experimental results on test problems with between 11 and 40 departments, between 2 and 3 floors, and between 2 and 6 elevators, show that this algorithm performed better than STAGES, which is presented in Section 2.2.

Multiple-Floor Facility Layout Design with Aisle Construction

Chang et al. [7] consider aisle construction in the multi-floor facility layout problem. The departments must be rectangularly shaped and may not be split by any space. The departments' size and shape remain unchanged throughout the procedure. This procedure can apply to problems where the floors have different areas by assuming that each floor has the same area and then forbidding certain areas of each floor. The numbers and locations of doors and elevators must be specified.

The procedure includes a construction stage that groups the departments using the K-means clustering algorithm. Reference departments are selected and are assigned to floors. The remaining departments are then assigned to floors individually. The result is that each of the groups are allocated to a floor so that departments of the same group occupy the same floor.

In the improvement stage, a genetic algorithm is used to improve the initial layout. Multiple chromosomes are used to represent departments in this multi-floor facility. This is combined with a heuristic decode function in order to generate a layout with doors and aisles.

Simulations show that the algorithm efficiently constructs layouts while constructing door and aisle structures automatically. Their simulations consist of 2 to 5 floors, 10 to 30 departments, and 1 to 4 elevators.

2.1.4 Mathematical Programming Techniques

Computer Aided Design Group's Space Planning System

Liggett and Mitchell [18] describe a software for space planning problems called Computer Aided Design Group's Space Planning System. This software system attempts to optimize operating efficiency by allocating "activities" to "facilities". In fact, three different types of problems can be handled by this system and are noted in [18]. These include the stacking or zone plan optimization problem (that optimizes the assignment of activities to parts of a facility), the block plan optimization problem (that optimizes the spatial arrangement of activities on a floor), and the move optimization plan (that optimizes the number of moves made within a facility).

The Space Planning System handles the above problems using a specialized form of the general quadratic assignment problem which is $\mathcal{NP}$-hard and is solved using a constructive initial placement strategy [18]. In this specialized form, fixed costs, interactive or communication costs, and move costs are considered. Each activity is composed of modules of equal size and each part of the facility is partitioned into location modules of the same size. The system assigns the activity modules to the location modules. Using modules in this way allows the system to handle problems in which different activities have different areas that do not necessarily match the areas of specific locations.

It also gives the user the ability to supply shape constraints by specifying minimum values for ratios involving a bounding rectangle drawn around the shape. Split penalties, large interaction values, are associated with pairs of modules from the same activity, so that the parts of the activity will be located as close as possible to each other in the case that activities must be split. They also assume that there is only one lift location which may be a group of centrally located elevators [23].

Multi-Floor Facility Layout Problem with Elevators

In 2007, Goetschalckx and Irohara [10] developed two formulations for the continuous facility layout problem with elevators; one with full-service elevators and one allowing partial-service elevators. This problem is known as the Multi-Floor Facility Layout Problem with Elevators (MFFLPE). Both formulations include, as decision variables, where to locate each department and elevator, the number of elevators, and which elevator to assign transportation operations. Elevators and travel aisle space are included in the areas of the departments. An elevator is given by a point that remains the same on any floor it services and must be located on the boundary of departments. Departments cannot be split on multiple floors, they are all rectangular, and have the same height equal to that of the floor. The shape and area of the departments are given, but the location and orientation of each department are decision variables.

In [10], symmetry-breaking techniques and valid inequalities are also presented to reduce computational times. The problem is solved using a combination of computer software including AMPL, a modeling language for mathematical programming and the CPLEX solver. The largest of the problems solved in this paper consists of 15 departments, 3 floors, and 6 elevators.

The Multi-Story Space Assignment Problem

The Multi-Story Space Assignment Problem (MSAP) is presented in [12] by Hahn, Smith, and Zhu. Here the multi-floor facility layout problem is modeled as a Generalized Quadratic 3-dimensional Assignment Problem (GQ3AP) and also includes an evacuation plan for the facility. The main objective is to assign the departments to floors so as to minimize the evacuation time given the number of people per department and the size restrictions of each department. A secondary objective is to minimize transportation costs given the flows and distances between the departments.

The GQ3AP is a $\mathcal{NP}$-hard problem and is applied to problems concerning a pair of independent simultaneous one-to-one assignments, which is why is it of interest for this type of problem; one must assign departments to locations while simultaneously assigning these same departments to escape exits. The footprint of the facility is a rectilinear polygon or can be closely approximated by a closed and bounded polygon. Each department can be subdivided, can be assigned separately to different stairwells, and no department may be split across floors.

The authors of this paper solve the MSAP using an exact solution method. Experiments consider between 7 and 8 floors, between 10 and 13 departments, and 2 to 3 stairwells. They recognize from the experimental results that the algorithm quickly provides solutions, however the run times are exponential in the problem size.

2.2 Multi-Stage Approaches Combining Exact and Heuristic Procedures

Several approaches for solving the multi-floor layout problem consist of two stages. In the first stage, departments are assigned to floors in order to reduce the vertical interaction costs and in the second stage, the layouts are optimized within each floor. Some two-stage approaches fix the departments to the floor they were assigned in the first stage throughout the second stage. The idea is that because the vertical interaction cost is usually more expensive than the horizontal interaction cost, minimizing the interaction between floors should reduce the solution space in the second stage while still including good solutions [23]. Other approaches do not restrict the departments to the floor they were assigned in the first stage rather they are allowed to move between floors in the second stage.

A heuristic method for the multi-story layout problem

Kaku, Thompson, and Baybars [15] present a heuristic procedure capable of producing solutions to large multi-floor layout problems with as many as 150 departments in reasonable times. The multi-story layout problem (MSLP) considered in this paper consists of two stages. The first stage groups the departments and assigns these groups to floors. The second stage uses the department to floor assignment from the first stage to determine the layout of each floor. The problem is broken up into subproblems, allowing the entire problem to be solved in a more reasonable time which is important, especially as the size of the problem increases.

This heuristic method assumes that the building has only one elevator or a group of elevators at one location and that all departments are interchangeable, thus requiring the departments to occupy the same floor space. On the other hand, modifying this method to handle unequal area departments is problematic and can increase the complexity and size of the problem [16].

In the first stage, a K-median heuristic is used to form groups of departments with the goal of including departments having a high interaction in the same group, minimizing inter-group interaction. These K groups need to contain an equal number of departments. An elevator can then be added to each group that will break up inter-group flows into three separate flows. So, a flow from department i in group I to a department j in group J is broken up into a flow from department i to the elevator belonging to group I, a flow from group I to group J, and a flow from the elevator of group J to department j. This is done for the purpose of reducing the problem into $K + 1$ QAPs. Included is one QAP that determines the group's floor number by considering the flow between groups. The remaining K QAPs find a layout on each floor by considering intra-group flows. Here, the flows between the elevator and departments help to ensure that those departments that have a high interaction with departments belonging to other groups are located near the elevator [15]. These QAPs are solved using heuristics presented in [14].

A simplified exchange procedure can then improve the entire solution through departmental exchanges across floors. It is a "simplified" version because instead of computing the change in the value of the objective function exactly, it is estimated at several steps of the process. Of course an exchange of this sort will change the groups and the flows. In effect, groups may have to be reassigned to floors and the layout of each floor will have to be determined again given the changes.

ALDEP

The Automated Layout Design Program (ALDEP) is a construction-type algorithm introduced in [25] by Seehof, Evans, Friederichs, and Quigley. It is a two-step program that assigns each department to a floor in the first stage and assigns the departments to locations within each floor in the second stage. Unfortunately, ALDEP can only work with a maximum of three floors at a time.

First the planner must specify the building and department requirements. Any area can be fixed which may represent aisles, bathrooms, and stairs, to name a few. A preference table is then constructed indicating the preferences for departments to be located near one another. The first department is randomly selected using a modified random-selection technique. To select another department, the preference table is searched for the department with the highest preference of being located near the already chosen department. If such a department is found, it is chosen next. Otherwise, another department is chosen randomly. This process continues until a complete layout, consisting of all the departments, is formed. This layout is then evaluated by adding together the preference values for bordering departments. Many layouts can be found and evaluated and the best of these can be considered further by the planner.

Meller and Bozer [4] mentioned three issues concerning ADELP: it ignores the vertical flow between departments after they have been assigned to floors; it is unclear how this assignment is made; and departments may be split across floors.

SABASS

Meller and Bozer [4] present a construction-type layout algorithm for manufacturing facilities with multiple floors and capacitated lifts. This algorithm consists of several stages. The first stage optimally assigns departments to floors without considering which lift is used and without splitting departments across floors. To do this, a mixed-integer linear programming formulation is used and is referred to as the Floor Assignment Formulation (FAF). It is $\mathcal{NP}$-hard and a branch-and-bound algorithm is used to solve this problem [23].

The second stage of the algorithm determines the layout of each floor simultaneously using the fixed floor assignments from FAF in stage one. It is assumed that the locations of existing or potential lifts are specified in advance and that the vertical flow will use the lift, l, that minimizes $d_{ij}^H = \min_l(d_{il}^H + d_{lj}^H)$. For the case where there is one lift whose location is known, an approach in which the floor layouts are constructed independently can be used. However, it is necessary to use another method when there is more than one lift. This is because in an algorithm such as this, where a department has not yet been assigned a location, it is not known in advance if that department will interact with a particular lift.

For the case of multiple lifts, this paper presents an improvement-type algorithm. Again, the departments are restricted to the floors to which they were assigned in stage one. Each floor is given a space filling curve and a layout sequence with dividers as in SABLE, above. An address consisting of a fixed component and a variable component are assigned to each department which indicates where it will be located in the layout sequence. The generated variable components are values between 0 and 1 and determine new layouts using an algorithm similar to SABLE described above. This algorithm is called the Simulated-Annealing Based Algorithm for the Second Stage (SABASS).

The third stage solves the Lift Location-Allocation Problem (LLAP). The LLAP is the problem of deciding which lifts to open and which lift to assign each vertical flow while not exceeding the throughput capacity of the lift [4]. It is assumed that every vertical flow can be assigned to only one lift and that only one lift can be at each location. The loads arrive at a lift according to a Poisson process and are served one at a time on a first-come-first-served basis. Specified and fixed are the distance between floors, the travel speeds of each lift, and the pick-up and deposit times of loads. The costs to minimize in the objective function include the amortized cost of the open lifts, the cost to wait for lifts, and the horizontal travel costs between departments to lifts. A simulated-annealing based heuristic algorithm similar to SABASS is used to solve this problem. However, instead of departments, flows and lifts are used.

Computational results, using the first two stages of this algorithm, show that it achieves results that are better than or equal to SABLE's for most problems [4]. A comparison of the runtimes depend on the problem size since FAF is used in the first stage and has difficulty solving large problem sizes such as the 40-department problem tested [4].

STAGES & FLEX

Meller and Bozer [23] introduce STAGES, a two-stage approach, combining mathematical-programming and simulated annealing. Here, the departments are assigned to floors with the goal of reducing the vertical handling cost in the first stage, and in the second stage, the layouts are determined on each floor.

FAF, the mixed-integer linear programming problem, is used for the first stage. Using a modified version of SABLE in which exchanges across floors are not allowed, the procedure attempts to improve the departmental layout of each floor in the second stage. This leaves the departments fixed to the floor to which they were assigned by FAF in the first stage. This step attempts to minimize the horizontal handling cost, while the vertical handling cost remains minimized [23].

Since this algorithm does not consider exchanges across floors, it does not consider the case when an exchange across a floor will decrease the horizontal handling cost more than the resulting gain in the vertical handling cost. So, Meller and Bozer [23] present another two-step procedure named FLEX for comparison purposes. FLEX uses FAF in the first stage followed by SABLE in the second. The difference here is that the departments are not fixed to the floor to which they were originally assigned.

Computational evidence concludes that STAGES outperforms both SABLE and FLEX [23]. The largest data set used to test these methods consists of 40 departments, 4 floors, and 3 lifts. Again, the run times vary and as noted in [23], they are between 0.5 and 2.5 times as long as SABLE's run times.

GRASP/TS and FAF/TS

Abdinnour-Helm and Hadley [1] present two heuristics. The first of these is GRASP/TS which is a two-stage heuristic. A modified version of GRASP is used in the first stage and a tabu search method is used in the second stage.

GRASP (Greedy Randomized Adaptive Search Procedures) is defined in [9] and consists of a construction phase and a local search phase. In the construction phase, a solution is constructed iteratively and a local search phase is used at each iteration that attempts to improve the solution.

In the first stage of GRASP/TS, a modified version of GRASP is used in which the assignment of departments to floors can be modeled as a graph partitioning problem. It finds an initial assignment of departments to floors while minimizing the inter-floor flow. In this modification of GRASP, the local search phase is not performed at each iteration, which is sufficient since the objective of the first stage is to find a good solution for the second stage, not to solve the whole problem.

The second stage uses Tabu Search (TS). Like simulated annealing, TS is a method that attempts to overcome the problem of getting stuck at local optimal solutions [1]. Spacefilling curves are used and one curve is generated for each floor. It begins with the initial layout obtained from stage one. At each iteration, TS finds the best feasible move by evaluating all possible feasible pairwise exchanges or shifts including those that occur across different floors. These are the only two types of moves that are allowed. A shift move is important because it allows a department to move to another location on the same floor and also to move to a different floor, allowing the number of departments on a floor to vary. Each move is then evaluated. Once the move is made, it is added to a tabu list which means that this move is forbidden, at least for some given number of future iterations. This method maintains a separate tabu list for exchange moves and shift moves. It is still possible to make a move that is on this list if the resulting layout is better than the best layout determined up to that point. There is a chance that the move that is selected does not actually improve the layout and this can help prevent TS from getting stuck at a local optimal solution. The process is repeated until it has reached its maximum number of iterations allowed.

The second heuristic presented in this paper is FAF/TS. This heuristic also consists of two stages. In the first stage, FAF which is described above, is used to obtain an exact solution to the above graph partitioning problem, and in the second stage, TS is applied as just described.

FAF/TS found some solutions that are the best known solutions to date on a few data sets. FAF/TS was also shown to outperform STAGES, indicating that tabu search performs better than simulated annealing in this situation [1]. The fact that FAF/TS and STAGES outperform SABLE, which was the best known single stage approach at the time, suggests that approaching the problem in two stages is a good approach [1].

Chapter 3

Background for New Mathematical Programming Approach

Next, I will present the mathematical-programming framework by Anjos and Vannelli [3] upon which the proposed three-stage multi-floor layout model is based. First, two methods are briefly introduced that are important to the development of the ideas in their framework.

It is assumed that there are N departments. The center of department i is (x_i, y_i). We have that c_{ij} is the cost per unit distance between departments i and j and that $c_{ij} = c_{ji}$. The distance between departments i and j, d_{ij}, is measured from the center of department i to the center of department j.

3.1 Dispersion Concentration Method (DISCON)

In 1980, Drezner [8] solved a version of the facility layout problem based on a non-convex mathematical-programming method named DISpersion-CONcentration in which each department i is approximated by a circle of radius r_i and center (x_i, y_i). The distance between two circles i and j is measured as $d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$.

The method determines the location of each department by solving the following formulation using a penalty-based algorithm:

$$\min_{(x_i, y_i)} \sum_{1 \le i < j \le N} c_{ij} d_{ij}$$
$$\text{s.t. } d_{ij} \ge r_i + r_j \text{ for all } 1 \le i < j \le N.$$

Inspired by the Big Bang theory, the algorithm has two phases. The first is the *DIS*persion phase for which the center of the circles are placed at one point (the origin) and are allowed to disperse. This phase provides good starting points for the second phase, the *CON*centration phase, where the departments are once again densely arranged achieving a local minimum and arriving at a final solution [8].

3.2 Nonlinear Optimization Layout Technique (NLT)

van Camp, Carter and Vannelli [26] introduced new heuristics that help find good solutions to the layout problem. They presented the Nonlinear Optimization Layout Technique (NLT) allowing for rectangular departments of any area with heights and widths determined throughout the optimization procedure.

The following model is the basic nonlinear optimization model used in the NLT method to approximate the real layout problem. The model will be denoted by vCCV to be consistent with the paper of Anjos and Vannelli [3]:

$$
\begin{array}{lll}
\min\limits_{(x_i,y_i),h_i,w_i,h_F,w_F} & \sum\limits_{1\le i<j\le N} c_{ij}d_{ij} & \\
\text{s.t.} & \mid x_i - x_j \mid -\frac{1}{2}(w_i + w_j) \ge 0 & \text{if } \mid y_i - y_j \mid -\frac{1}{2}(h_i + h_j) < 0 \\
& \mid y_i - y_j \mid -\frac{1}{2}(h_i + h_j) \ge 0 & \text{if } \mid x_i - x_j \mid -\frac{1}{2}(w_i + w_j) < 0 \\
& \frac{1}{2}w_F - (x_i + \frac{1}{2}w_i) \ge 0 & \text{for } i = 1, \ldots, N \\
& \frac{1}{2}h_F - (y_i + \frac{1}{2}h_i) \ge 0 & \text{for } i = 1, \ldots, N \\
& (x_i - \frac{1}{2}w_i) + \frac{1}{2}w_F \ge 0 & \text{for } i = 1, \ldots, N \\
& (y_i - \frac{1}{2}h_i) + \frac{1}{2}h_F \ge 0 & \text{for } i = 1, \ldots, N \\
& \min(w_i, h_i) - l_i^{\min} \ge 0 & \text{for } i = 1, \ldots, N \\
& l_i^{\max} - \min(w_i, h_i) \ge 0 & \text{for } i = 1, \ldots, N \\
& \min(w_F, h_F) - l_F^{\min} \ge 0 & \\
& l_F^{\max} - \min(w_F, h_F) \ge 0, &
\end{array}
$$

where $d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$. Here (x_i, y_i) is the center of department i, w_i and h_i represent the width and height of department i, w_F and h_F represent the width and height of the facility, and l_i^{min}, l_i^{max}, l_F^{min} and l_F^{max} are the minimum and maximum allowable lengths for the shortest side of department i and the facility.

The NLT method adopts a three-stage approach that uses penalty function methods. Stage One is a relaxation that attempts to distribute the centers of the departments evenly throughout the floor space, completely ignoring the boundaries of the departments. In Stage Two, each department is represented by a circle whose diameter equals the square root of the area, such that the circle is inscribed in a square having the same area of the department. Using a relaxation of the vCCV model, a layout is determined where the circles do not overlap and are contained within the boundaries of the facility. The solution of this stage can be used as initial values for the vCCV model. The Stage Two model is:

$$
\begin{aligned}
\min_{(x_i,y_i),h_F,w_F} \quad & \sum_{1\le i<j\le N} c_{ij} d_{ij} \\
\text{s.t. } & d_{ij} - (r_i + r_j) \ge 0 && \text{for all } i,\ j = 1,\dots,N \\
& \frac{1}{2} w_F - (x_i + r_i) \ge 0 && \text{for all } i = 1,\dots,N \\
& \frac{1}{2} h_F - (y_i + r_i) \ge 0 && \text{for all } i = 1,\dots,N \\
& \frac{1}{2} w_F + (x_i - r_i) \ge 0 && \text{for all } i = 1,\dots,N \\
& \frac{1}{2} h_F + (y_i - r_i) \ge 0 && \text{for all } i = 1,\dots,N \\
& \min(w_F, h_F) - l_F^{\min} \ge 0 \\
& l_F^{\max} - \min(w_F, h_F) \ge 0,
\end{aligned}
$$

where all the parameters and variables are as defined before.

Finally, in Stage Three the departments are modeled as rectangles and using the solution of Stage Two as the initial layout, the final solution is determined by solving the vCCV model.

3.3 The Anjos-Vannelli Facility-Layout Design

Anjos and Vannelli [3] present a framework which consists of two stages. They combine two new mathematical-programming models to find solutions for the facility-layout problem. The purpose of the first stage is to find a solution that provides good initial values for the next stage. The second stage attempts to find a locally optimal layout. In addition, the framework incorporates aspect-ratio constraints that prevent unrealistically shaped departments in the final layout.

3.3.1 Stage One: ModCoAR

Stage One uses an attractor-repeller (AR) model, which is a relaxation of the layout problem that improves upon the first two stages of the NLT method. In this model, each department is also approximated by a circle of radius r_i and center (x_i, y_i). Its purpose is to find good initial values for the next stage in which the final layout is determined.

AR model

The AR model introduced by Anjos and Vannelli [3] is given as follows:

$$
\begin{aligned}
\min_{(x_i,y_i),h_F,w_F} \sum_{1\le i<j\le N} & c_{ij}D_{ij} + f(\frac{D_{ij}}{t_{ij}}) \\
s.t. \frac{1}{2}w_F &\ge x_i + r_i \quad \text{for } i = 1,\dots,N \\
\frac{1}{2}w_F &\ge r_i - x_i \quad \text{for } i = 1,\dots,N \\
\frac{1}{2}h_F &\ge y_i + r_i \quad \text{for } i = 1,\dots,N \\
\frac{1}{2}h_F &\ge r_i - y_i \quad \text{for } i = 1,\dots,N \\
w_F^{max} &\ge w_F \ge w_F^{min} \\
h_F^{max} &\ge h_F \ge h_F^{min},
\end{aligned}
$$

where $f(z) = (1/z) - 1$ for $z > 0$ and $t_{ij} = \alpha(r_i + r_j)^2$ for a given $\alpha > 0$ and for $1 \le i < j \le N$. Furthermore $D_{ij} = (x_i - x_j)^2 + (y_i - y_j)^2$ denotes the square distance between departments i and j and $w_F^{max}, w_F^{min}, h_F^{max}, h_F^{min}$ denote the maximum and minimum widths and heights of the facility. The first four constraints ensure that all the circles remain completely inside the bounds of the facility and the remaining provide a bound for the shape of the facility.

As mentioned, the AR model improves upon the first two stages of the NLT method. Both methods are non-convex, but the AR model has only linear constraints, which is a major advantage over the NLT method. The main difference here is that in the AR model, the non-overlap constraints are enforced through the use of *target distances* instead of the constraints $d_{ij} \ge r_i + r_j$, for all $1 \le i < j \le N$. The concept of target distances is introduced next.

Notice that since the costs are nonnegative, the term $\sum_{i,j} c_{ij}D_{ij}$ would achieve a minimum when D_{ij}, the square of the distance between departments i and j, is as small as possible. This acts as an *attractor* because it causes the distances between each pair of circles to decrease. It can be seen that without a constraint of the form $d_{ij} \geq r_i + r_j$, or the second term in the objective function, the minimum value would be achieved when $D_{ij} = 0$, that is when the circles i and j completely overlap each other. Instead of using the constraints $d_{ij} \geq r_i + r_j$, for all $1 \leq i < j \leq N$, as in the NLT method, a *repeller term* $f(\frac{D_{ij}}{t_{ij}})$ is added to the objective function and works to prevent the circles from overlapping [3].

The AR model aims to ensure an ideal separation between two circles at optimality. Theoretically, this occurs when $\frac{D_{ij}}{t_{ij}} = \frac{(x_i-x_j)^2+(y_i-y_j)^2}{\alpha(r_i+r_j)^2} = 1$. Here, $\sqrt{t_{ij}}$ is the *target distance* between two pairs of circles i and j and t_{ij} is the target value for D_{ij}. When $\alpha = 1$ and when $\frac{D_{ij}}{t_{ij}} = 1$ at optimality, the circles should intersect at exactly one point. Then, of course, having $\alpha < 1$ in $t_{ij} = \alpha(r_i + r_j)^2$ would be a relaxation in that some overlap would be allowed between the circles i and j. Choosing $\alpha > 1$ would enforce a greater separation between circles [2]. This concept is illustrated in Figure 3.1.

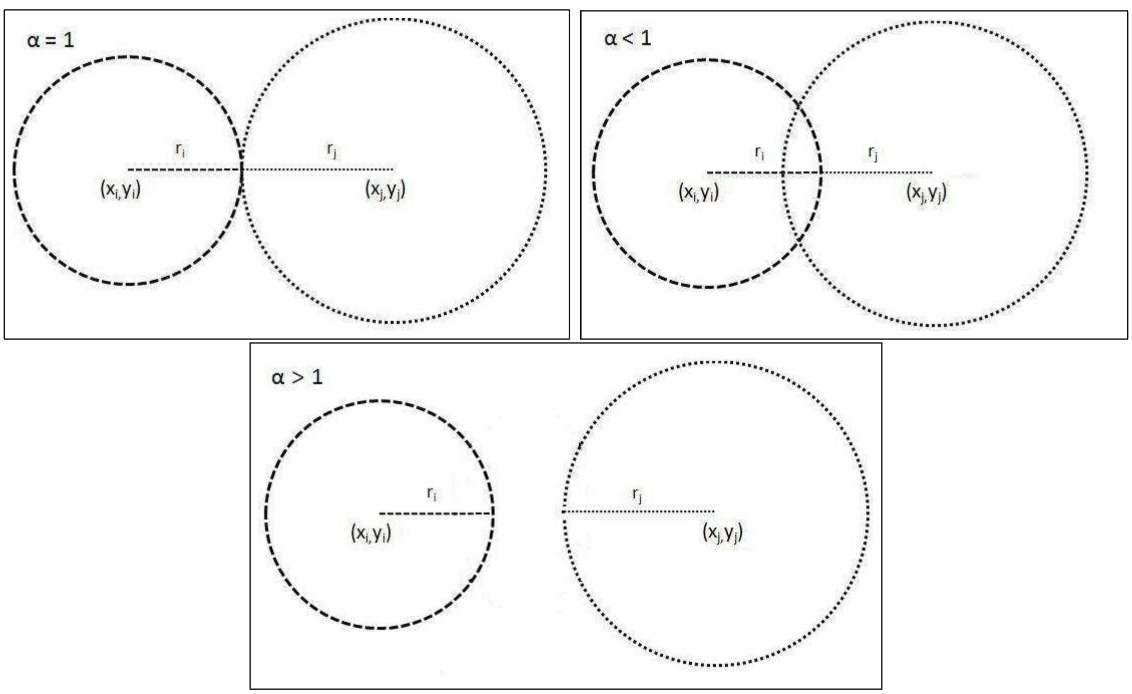

Figure 3.1: The theoretical concept of target distances.

Initially, the circles are arranged in a layout in which the squares of the distances between the circles are much larger than the corresponding target distances, implying that $\frac{D_{ij}}{t_{ij}} > 1$. The attractor-repeller effect is accomplished by penalizing overlap through the inclusion of the *repeller term* in the objective function. Recall that the repeller term is $f(\frac{D_{ij}}{t_{ij}}) = \frac{t_{ij}}{D_{ij}} - 1$, where $\frac{D_{ij}}{t_{ij}} > 0$. The attractor term in the objective function aims to decrease $\frac{D_{ij}}{t_{ij}}$, while the repeller term aims to increase this ratio until it has reached an equilibrium. The goal is to achieve this equilibrium when $\frac{D_{ij}}{t_{ij}} \approx 1$ by adjusting the parameter α [2]. The choice of this parameter is discussed in Section 5.1.

Convexified AR Model

Since the objective function of the AR model is not convex, a new convex version, CoAR, is presented in [3]. Assuming that $c_{ij} > 0$ and $t_{ij} > 0$ for all i, j, the following piecewise function is convex and continuously differentiable [2]:

$$f_{ij}(x_i, x_j, y_i, y_j) := \begin{cases} c_{ij}z + \frac{t_{ij}}{z} - 1, & z \geq \sqrt{\frac{t_{ij}}{c_{ij}}} \\ 2\sqrt{c_{ij}t_{ij}} - 1, & 0 \leq z < \sqrt{\frac{t_{ij}}{c_{ij}}} \end{cases}$$

where $z = (x_i - x_j)^2 + (y_i - y_j)^2$.

Anjos and Vannelli [3] present CoAR as:

$$\begin{aligned} \min_{(x_i,y_i),h_F,w_F} \quad & \sum_{1 \leq i < j \leq N} f_{ij}(x_i, x_j, y_i, y_j) \\ s.t. \quad & \frac{1}{2}w_F \geq x_i + r_i \quad \text{for } i = 1, \ldots, N \\ & \frac{1}{2}w_F \geq r_i - x_i \quad \text{for } i = 1, \ldots, N \\ & \frac{1}{2}h_F \geq y_i + r_i \quad \text{for } i = 1, \ldots, N \\ & \frac{1}{2}h_F \geq r_i - y_i \quad \text{for } i = 1, \ldots, N \\ & w_F^{max} \geq w_F \geq w_F^{min} \\ & h_F^{max} \geq h_F \geq h_F^{min}. \end{aligned}$$

Generalized Target Distances

The CoAR model motivates the discussion of *generalized target distances*. It can be shown that the minimum of the function f_{ij} occurs when $D_{ij} \leq \sqrt{t_{ij}/c_{ij}}$ [3]. However, the situation where the circles completely overlap (when $D_{ij} = 0$) also satisfies this inequality.

It is desirable to seek a layout where the overlap of circles is minimized and this occurs when $D_{ij} \approx \sqrt{t_{ij}/c_{ij}}$ [3].

The generalized target distance T_{ij} is defined in [3] as

$$T_{ij} = \sqrt{\tfrac{t_{ij}}{c_{ij}+\epsilon}}, \text{ for all } 1 \leq i < j \leq N.$$

A small number $\epsilon > 0$ is included to enforce the assumption that $c_{ij} > 0$ made for the CoAR model. Intuitively, it is desirable to seek a layout in which $D_{ij} \approx T_{ij}$. First, D_{ij} is proportional to the corresponding target distance, t_{ij}. In addition, D_{ij} is inversely proportional to c_{ij}. If the cost between circles i and j is high, then T_{ij} is small and since $D_{ij} \approx T_{ij}$, the two circles will probably be close to each other. On the other hand, if the cost c_{ij} is low, then T_{ij} is high and by the same reasoning the circles will probably be located farther away from each other in the layout.

ModCoAR Model

Applying generalized target distances to the CoAR model would require a specialized algorithm that is not very practical. So, a new model, although not convex, is used so that (theoretically) $D_{ij} \approx T_{ij}$ at optimality. This new model is ModCoAR and is presented in [3] as follows:

$$\begin{aligned}
\min_{(x_i,y_i),h_F,w_F} \quad & \sum_{1\leq i<j\leq N} F_{ij}(x_i,x_j,y_i,y_j) - K_{MOD}\ln(\frac{D_{ij}}{T_{ij}}) \\
\text{s.t.} \quad & \frac{1}{2}w_F \geq x_i + r_i \quad \text{for } i = 1,\ldots,N \\
& \frac{1}{2}w_F \geq r_i - x_i \quad \text{for } i = 1,\ldots,N \\
& \frac{1}{2}h_F \geq y_i + r_i \quad \text{for } i = 1,\ldots,N \\
& \frac{1}{2}h_F \geq r_i - y_i \quad \text{for } i = 1,\ldots,N \\
& w_F^{max} \geq w_F \geq w_F^{min} \\
& h_F^{max} \geq h_F \geq h_F^{min},
\end{aligned}$$

where K_{MOD} is a scaling factor and

$$F_{ij}(x_i,x_j,y_i,y_j) := \begin{cases} c_{ij}z + \frac{t_{ij}}{z} - 1, & z \geq T_{ij} \\ 2\sqrt{c_{ij}t_{ij}} - 1, & 0 \leq z < T_{ij}. \end{cases}$$

3.3.2 Stage Two: BPL

In stage two, the Bilinear Penalty Layout Model (BPL) uses the solution of ModCoAR as initial values in order to solve the layout problem. In fact, BPL is an exact formulation of the facility layout problem and is modeled as:

$$
\begin{aligned}
\min_{(x_i,y_i),h_i,w_i,h_F,w_F} \quad & \sum_{1 \le i < j \le N} c_{ij}\delta(x_i, y_j, x_j, y_j) + K_{BPL} X_{ij} Y_{ij} & \\
\text{s.t. } & X_{ij} \ge \frac{1}{2}(w_i + w_j) - |x_i - x_j| & \text{for all } 1 \le i < j \le N \\
& Y_{ij} \ge \frac{1}{2}(h_i + h_j) - |y_i - y_j| & \text{for all } 1 \le i < j \le N \\
& X_{ij} \ge 0, \quad Y_{ij} \ge 0, \quad \text{and } \; X_{ij}Y_{ij} = 0 & \text{for all } 1 \le i < j \le N \\
& \frac{1}{2}w_F - (x_i + \frac{1}{2}w_i) \ge 0 & \text{for } i = 1, \ldots, N \\
& (x_i - \frac{1}{2}w_i) + \frac{1}{2}w_F \ge 0 & \text{for } i = 1, \ldots, N \\
& \frac{1}{2}h_F - (y_i + \frac{1}{2}h_i) \ge 0 & \text{for } i = 1, \ldots, N \\
& (y_i - \frac{1}{2}h_i) + \frac{1}{2}h_F \ge 0 & \text{for } i = 1, \ldots, N \\
& w_i h_i = a_i & \text{for } i = 1, \ldots, N \\
& w_i^{\max} \ge w_i \ge w_i^{\min} & \text{for } i = 1, \ldots, N \\
& h_i^{\max} \ge h_i \ge h_i^{\min} & \text{for } i = 1, \ldots, N \\
& w_F^{\max} \ge w_F \ge w_F^{\min} \quad \text{and} \quad h_F^{\max} \ge h_F \ge h_F^{\min}. &
\end{aligned}
$$

Here, K_{BPL} is a penalty constant and $\delta(x_i, y_j, x_j, y_j)$ is the distance function, which may be measured with several different norms [3]. The first three constraints are non-overlap constraints, replacing the more intuitive, but disjunctive non-overlap constraints that can be expressed as $\frac{1}{2}(w_i+w_j)-|x_i-x_j| \le 0$ or $\frac{1}{2}(h_i+h_j)-|y_i-y_j| \le 0$. The constraints $X_{ij} \ge 0$, $Y_{ij} \ge 0$, and $X_{ij}Y_{ij} = 0$ for all $1 \le i < j \le N$, make the model a mathematical program with equilibrium constraints (MPEC) [19]. Anjos and Vannelli [3] penalize $X_{ij}Y_{ij} = 0$ for all i, j in the objective function since MINOS, which is used to solve this problem, would otherwise fail when applied to BPL. Handling the problem in this way often successfully leads to solutions where $X_{ij}Y_{ij} = 0$ for all i,j [3].

3.3.3 Aspect-Ratio Constraints

Anjos and Vannelli [3] also incorporate *aspect-ratio constraints* into the BPL model that allow the user to have control over the shape of the departments. Without these constraints, it is possible that the final layout will contain one or a few very long and narrow departments which is not always practical. The aspect ratio for department i is defined as $\beta_i = \max\{h_i, w_i\}/\min\{h_i, w_i\}$. Figure 3.2 demonstrates a layout obtained using the Armour and Buffa 20-department problem without using any aspect ratio constraints and another with aspect ratio constraints of $\beta_i \leq 3$. It can be seen that the narrowness of some departments without any aspect ratio constraints is not practical for many applications.

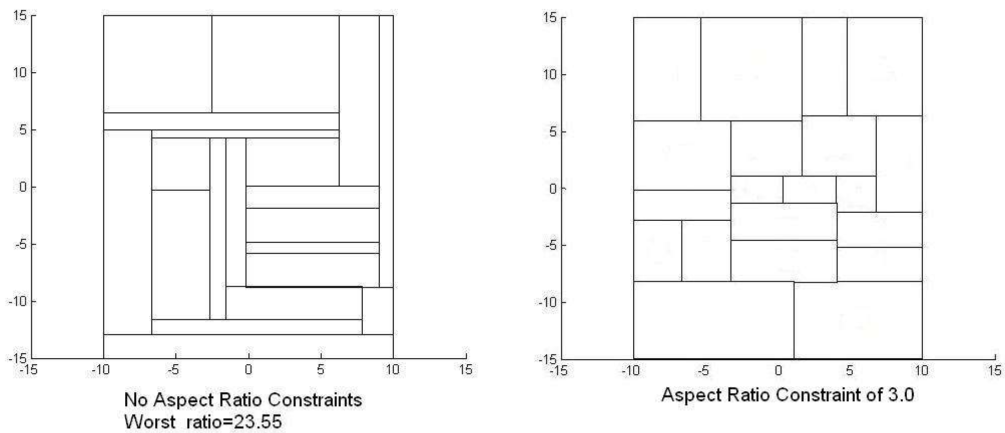

Figure 3.2: Final layouts of Armour and Buffa 20-department problem using Anjos-Vannelli Method [3].

The ModCoAR and BPL models given above constitute the two stages of the mathematical-programming framework for facility-layout design by Anjos and Vannelli [3]. Results of computational experiments in [3] indicate that they improve on previous results obtained from other single-floor methods in the literature. The methods of this single-floor framework are modified and extended for the proposed three-stage multi-floor layout method presented next which is the main contribution of this report.

Chapter 4

Three-Stage Multi-Floor Facility Layout Method

From the literature review in Chapter 2, it can be seen that methods for solving the multi-floor layout problem consist of one or several stages. Methods such as MULTIPLE [5] and SABLE [22] that approach the problem in a single stage use special techniques in order to ensure that departments are not split across floors and that the problem remains feasible when departments are moved across floors. It can also be seen in the literature review that multi-stage approaches such as STAGES [23] and GRASP/TS [1], in which the departments are assigned to floors in the first stage and the layout is optimized on each floor in the second stage, perform just as well, if not better, than the best single stage approaches in many practical cases.

It has also been seen that some multi-stage approaches such as FLEX [23] allow departmental exchanges across floors after they have already been assigned in the first stage, while others such as STAGES do not [23]. Experimental results by Meller and Bozer [23] suggest that allowing departments to be exchanged across floors is not necessarily advantageous even though it is possible that an exchange of this type will result in a reduction of horizontal costs greater than that of the increase in vertical costs. They found that this is true, in general, as long as the ratio, c_{ij}^V/c_{ij}^H, is greater than or equal to 1 and that the floor layouts are solved simultaneously [23]. If the ratio is greater than 1, it is desirable that departments with a high level of interaction be located on the same floor because of the greater cost for vertical travel than horizontal travel [23]. They hypothesize that the success of the two-stage method in which departments are fixed to a floor is due to the fact that it operates over a smaller solution space, that is, a low cost portion of the whole solution space. If there were no limits on the running time of the procedure, this would probably not be true, but for practical purposes, smaller computational effort is important. Hence, for this research project we consider a method with multiple stages.

In the first stage of the proposed three-floor multi-floor facility layout method, the task is to assign departments to floors minimizing the vertical interaction cost. The second and third stages use an extension of the mathematical-programming framework by Anjos and Vannelli given in Section 3.3 in order to solve the multi-floor problem. The layout of each floor can be solved simultaneously or independently. Theoretically, in the case of a single elevator location, the two versions of this method should be equivalent, however both have their advantages and disadvantages.

4.0.4 Solving Each Floor Independently vs. All Floors Simultaneously

The first version solves the layout of each floor independently of the other floors and will be denoted FBF (Floor-By-Floor). This case allows several smaller problems to be solved separately as opposed to solving the one larger, more complex problem of solving for the layout of all floors simultaneously. However, this version can only handle up to one elevator location. This is because since the layout on another floor is not known until the end of the procedure, the elevator that will minimize the travel distance between two departments on different floors cannot be determined throughout the optimization procedure.

As will be seen, the models in stages two and three of the simultaneous version, denoted AFS (All-Floors-Simultaneously), require there to be a penalty term in the objective function for each floor. In practice, several penalty terms in the objective function may be more difficult for the solver, but these models have the advantage of allowing for multiple elevator locations.

If one elevator location is sufficient, then FBF presented in Section 4.3.1 can be used. It is capable of providing good quality solutions in a short time and can also solve large problems with many departments and several floors. If multiple elevator locations are required, then AFS that solves each floor simultaneously would be required. This version is presented in Section 4.3.2 and has also proven to be able to solve large problems.

4.1 Notation for the Three-Stage Multi-Floor Model

We consider N departments and K floors where the areas of the departments are given and the lengths, widths, and positions of the departments are optimized in the layout. To be consistent with the notation in [23], the area needed for department i is a_i and A_k is the maximum floor space that can be used on floor k. The distance between any two adjacent floors is given by δ and y_i is the variable denoting the floor number of department i.

In the three-stage multi-floor model, if two departments i and j are on the same floor, the horizontal distance, d_{ij}^H, is simply the distance between the two departments which can be measured with various norms. On the other hand, the distance between two departments on different floors must include both the horizontal and vertical distance between them. As in [23], the horizontal distance, d_{ij}^H, is $d_{ij}^H = \min_e (d_{ie} + d_{ej})$, where d_{ie} is the distance between department i and elevator e on one floor and d_{ej} is the distance between elevator e and department j on the other floor. The vertical distance, d_{ij}^V, between two departments not on the same floor is $d_{ij}^V = \delta|y_i - y_j|$.

It is important to note that the costs c_{ij} and flows f_{ij} are not necessarily symmetric, which allows for the case when the interaction between departments i and j is not necessarily the same as the interaction between departments j and i. It is also necessary to distinguish between vertical costs c_{ij}^V and horizontal costs c_{ij}^H seeing that the cost to transport materials in the vertical direction is likely more costly than transporting materials to another area on the same floor.

4.2 Stage One: Assigning Departments to Floors

In the first stage, the task is to assign departments to floors, minimizing the vertical interaction cost using a mathematical programming formulation. A binary variable x_{ij} is 1 if department i is placed on floor j, and 0 otherwise. The constraints need to guarantee that the area capacity of each floor is not exceeded and that each department is assigned to exactly one floor. Then the vertical cost and distance must be included in the objective function when two departments are not on the same floor. This is modeled by Meller and Bozer [23] as:

$$\begin{aligned}
\min \sum_{i=1}^{N}\sum_{j=1}^{N}\sum_{k=1}^{K}\sum_{g=1}^{K} & f_{ij}c_{ij}^V\hat{d}_{kg}x_{ik}x_{jg} & \\
\text{s.t.} \sum_{k=1}^{K} & x_{ik} = 1 & \text{for } i = 1,\dots,N \\
\sum_{i=1}^{N} & a_i x_{ik} \le A_k & \text{for } k = 1,\dots,K,
\end{aligned}$$

where

$$x_{ik} := \begin{cases} 1, & \text{if department } i \text{ is assigned to floor } k \\ 0, & \text{otherwise} \end{cases}$$

and $\hat{d}_{kg}$ is the distance between floor k and floor g.

As pointed out by Meller and Bozer in [23], this problem is simply a generalization of the traditional quadratic assignment problem, which is known to be $\mathcal{NP}$-hard and generally difficult to solve with more than 20 departments. They improve on this formulation by considering the structure of the inter-floor distance function, creating a model with a linear objective function, allowing for larger problem sizes to be solved [23]. This improved model is FAF [23], which has already been mentioned in the above literature review. With the exception of some changes in notation, the following is equivalent to FAF and is used in Stage One of the three-stage multi-floor layout model:

$$\begin{aligned}
\min \sum_{i=1}^{N} \sum_{j=1}^{N} V_{ij} & \\
\text{s.t. } \sum_{k=1}^{K} k x_{ik} = y_i & \quad \text{for } i = 1, \ldots, N \\
V_{ij} \geq (y_i - y_j)\delta c_{ij}^{V} f_{ij} & \quad \text{for } i,\ j = 1, \ldots, N \\
V_{ij} \geq (y_j - y_i)\delta c_{ij}^{V} f_{ij} & \quad \text{for } i,\ j = 1, \ldots, N \\
\sum_{k=1}^{K} x_{ik} = 1 & \quad \text{for } i = 1, \ldots, N \\
\sum_{i=1}^{N} a_i x_{ik} \leq A_k & \quad \text{for } k = 1, \ldots, K,
\end{aligned}$$

where

$$x_{ik} := \begin{cases} 1, & \text{if department } i \text{ is assigned to floor } k \\ 0, & \text{otherwise.} \end{cases}$$

4.3 Stages Two and Three: Optimizing the Layout of Each Floor

Stages Two and Three use the mathematical-programming framework of Anjos and Vannelli for the single floor case and extend its ideas in order to solve the multi-floor problem. The layout of each floor is solved independently in FBF and simultaneously for all floors in AFS.

4.3.1 FBF: Optimizing the Layout of Each Floor Independently

Each floor uses modified versions of the ModCoAR and BPL models presented in Section 3.3. I will denote ModCoAR_l as the modified version of ModCoAR for floor l and similarly, BPL_l as the modified BPL model for floor l.

An elevator can be given a fixed position on each floor and, of course, must be located in the same position on each floor. In other words, the center of the elevator (x_E, y_E) must be fixed at the same coordinates on each floor. In addition to the horizontal interaction costs present in the single floor model, vertical interaction costs must be considered as well. This means that, for example, a department i on floor l that interacts heavily with department j located on another floor, will ideally be located closer to the elevator (in order to minimize costs) than another department on floor l that does not interact with any departments on another floor. It is this reasoning that motivates the three cases included in my model. On each floor, l, the layouts are optimized independently by considering three cases; each case must be included in the objective functions of both the ModCoAR_l and BPL_l models:

On each floor l, consider:
Case 1: Departments i and j are both on floor l.
Let $\mathcal{A}_E^l$={ $(i,j)|$ departments i and j are both on floor l }. This case can be handled in a way equivalent to the single floor problem presented by Anjos and Vannelli [3], as will be seen.
Case 2: Department i is on floor l and department j is on another floor.
Let $\mathcal{B}_E^l$={ $(i,j)|$ department i is on floor l and department j is on another floor }. On floor l, we calculate the portion of the cost that is between department i and the elevator.
Case 3: Department i is on another floor and department j is on floor l.
Let $\mathcal{C}_E^l$={ $(i,j)|$ department i is on another floor and department j is on floor l }. On floor l, we calculate the portion of the cost that is between the elevator and department j.

Additional Notation

Recall that in this new approach, the costs and flows are not necessarily symmetric. Since the model by Anjos and Vannelli [3] does not distinguish between costs and flows, but most of the multi-floor models in the literature do, the costs are redefined as

$$c_{ij}^H := c_{ij}^H f_{ij} \text{ and } c_{ij}^V := c_{ij}^V f_{ij}.$$

The radius of department i, r_i, and the radius of the elevator E, r_E, are defined as

$$r_i = \sqrt{\frac{a_i}{\pi}} \text{ and } r_E = \sqrt{\frac{a_E}{\pi}}.$$

The target distances are modified for each of the above cases:

$$t_{ij} = \alpha_l (r_i + r_j)^2, \text{ if } (i,j) \in \mathcal{A}_E^l$$

$$t\_E_{ij} := \begin{cases} \alpha_l (r_i + r_E)^2, & \text{if } (i,j) \in \mathcal{B}_E^l \\ \alpha_l (r_E + r_j)^2, & \text{if } (i,j) \in \mathcal{C}_E^l, \end{cases}$$

where α_l is a parameter for floor l.

The generalized target distances include

$$T_{ij} = \sqrt{\frac{t_{ij}}{c_{ij} + \epsilon}} \text{ if } (i,j) \in \mathcal{A}_E^l \quad \text{and} \quad T\_E_{ij} = \sqrt{\frac{t\_E_{ij}}{c_{ij} + \epsilon}} \text{ if } (i,j) \in \mathcal{B}_E^l \cup \mathcal{C}_E^l.$$

These next variables are equivalent to the square distances D_{ij}, D_{iE} and D_{Ej}:

$$z = (x_i - x_j)^2 + (y_i - y_j)^2,$$

$$z\_E := \begin{cases} (x_i - x_E)^2 + (y_i - y_E)^2, & \text{if } (i,j) \in \mathcal{B}_E^l \\ (x_E - x_j)^2 + (y_E - y_j)^2, & \text{if } (i,j) \in \mathcal{C}_E^l. \end{cases}$$

K_{MOD_l} and K_{BPL_l} are parameters for ModCoAR_l and BPL_l, respectively.

Stage Two: ModCoAR_l

The ModCoAR_l model is given next:

$$\min_{(x_i,y_i),h_F,w_F} \sum_{\substack{i,j\in\mathcal{A}_E^l\\ i\neq j}} [F_{ij}(x_i,x_j,y_i,y_j) - K_{MOD_l}\ln(\frac{D_{ij}}{T_{ij}})]$$
$$+ \sum_{i,j\in\mathcal{B}_E^l} [F\_E_{ij}(x_i,x_j,y_i,y_j) - K_{MOD_l}\ln(\frac{D_{iE}}{T\_E_{ij}})]$$
$$+ \sum_{i,j\in\mathcal{C}_E^l} [F\_E_{ij}(x_i,x_j,y_i,y_j) - K_{MOD_l}\ln(\frac{D_{Ej}}{T\_E_{ij}})]$$
$$\text{s.t. } \frac{1}{2}w_F \geq x_i + r_i \text{ for all } i \text{ on floor } l$$
$$\frac{1}{2}w_F \geq r_i - x_i \text{ for all } i \text{ on floor } l$$
$$\frac{1}{2}h_F \geq y_i + r_i \text{ for all } i \text{ on floor } l$$
$$\frac{1}{2}h_F \geq r_i - y_i \text{ for all } i \text{ on floor } l$$
$$w_F^{\max} \geq w_F \geq w_F^{\min}$$
$$h_F^{\max} \geq h_F \geq h_F^{\min},$$

where if $(i,j)\in\mathcal{A}_E^l$ then

$$F_{ij}(x_i,x_j,y_i,y_j) := \begin{cases} c_{ij}^H z + \frac{t_{ij}}{z} - 1, & z \geq T_{ij} \\ 2\sqrt{c_{ij}^H t_{ij}} - 1, & 0 \leq z < T_{ij}, \end{cases}$$

and if $(i,j)\in\mathcal{B}_E^l \cup \mathcal{C}_E^l$ then

$$F\_E_{ij}(x_i,x_j,y_i,y_j) := \begin{cases} c_{ij}^H z\_E + \frac{t\_E_{ij}}{z\_E} - 1, & z\_E \geq T\_E_{ij} \\ 2\sqrt{c_{ij}^H t\_E_{ij}} - 1, & 0 \leq z\_E < T\_E_{ij}. \end{cases}$$

Stage Three: BPL_l

BPL_l is formulated as:

$$\begin{aligned}
\min_{(x_i,y_i),h_F,w_F} & \sum_{\substack{i,j\in\mathcal{A}_E^l \\ i\neq j}} [c_{ij}^H \delta(x_i,y_i,x_j,y_j) + K_{BPL_l} X_{ij}Y_{ij}] \\
& + \sum_{i,j\in\mathcal{B}_E^l} [c_{ij}^H \delta(x_i,y_i,x_E,y_E)] \\
& + \sum_{i,j\in\mathcal{C}_E^l} [c_{ij}^H \delta(x_E,y_E,x_j,y_j)] \\
\text{s.t. } & X_{ij} \geq \frac{1}{2}(w_i+w_j) - |x_i - x_j| && \text{for all } (i,j)\in\mathcal{A}_E^l,\ i\neq j \\
& Y_{ij} \geq \frac{1}{2}(h_i+h_j) - |y_i - y_j| && \text{for all } (i,j)\in\mathcal{A}_E^l,\ i\neq j \\
& X_{ij}\geq 0, \quad Y_{ij}\geq 0, \quad \text{and} \quad X_{ij}Y_{ij}=0, && \text{for all } (i,j)\in\mathcal{A}_E^l,\ i\neq j \\
& \frac{1}{2}w_F - (x_i + \frac{1}{2}w_i) \geq 0 && \text{for all } i \text{ on floor } l \\
& (x_i - \frac{1}{2}w_i) + \frac{1}{2}w_F \geq 0 && \text{for all } i \text{ on floor } l \\
& \frac{1}{2}h_F - (y_i + \frac{1}{2}h_i) \geq 0 && \text{for all } i \text{ on floor } l \\
& (y_i - \frac{1}{2}h_i) + \frac{1}{2}h_F \geq 0 && \text{for all } i \text{ on floor } l \\
& w_i h_i = a_i, && \text{for all } i \text{ on floor } l \\
& w_i^{\max} \geq w_i \geq w_i^{\min} && \text{for all } i \text{ on floor } l \\
& h_i^{\max} \geq h_i \geq h_i^{\min} && \text{for all } i \text{ on floor } l \\
& w_F^{\max} \geq w_F \geq w_F^{\min} \text{ and } \ h_F^{\max} \geq h_F \geq h_F^{\min}.
\end{aligned}$$

Outline of FBF

The general outline of FBF is

```
Solve FAF;
TotalCost=0;
for l = 1 to K do
   Solve ModCoAR_l;
   Solve BPL_l;
   TotalCost=TotalCost+BPLCost;
```

end for
VerticalCost=$\sum_{ij} c_{ij}^V * \delta * |y_i - y_j|$;
TotalCost=TotalCost+VerticalCost;

4.3.2 AFS: Optimizing the Layout of Each Floor Simultaneously

Optimizing the layout of each floor simultaneously has the advantage of allowing for multiple elevator locations. AFS uses an extension of the ModCoAR and BPL methods denoted Multi-ModCoAR and Multi-BPL. Several notation modifications are made.

Additional Notation

The target distances are modified for the simultaneous case and are given as

$$t\_l_{ij} = \alpha_l (r_i + r_j)^2, \text{ for all } i, j \text{ on floor } l \text{ and for all } 1 \leq l \leq K.$$

The generalized target distances for each floor l are given as

$$T\_l_{ij} = \sqrt{\frac{t\_l_{ij}}{c_{ij} + \epsilon}}, \text{ for all } i, j \text{ on floor } l.$$

The M elevators are denoted $E_1, \ldots, E_m$.

Stage 2: Multi-ModCoAR Model

$$
\begin{aligned}
\min_{(x_i,y_i),h_F,w_F} & \sum_{\substack{i,j \text{ on floor 1} \\ i\neq j}} [F\_1_{ij}(x_i,x_j,y_i,y_j) - K_{MOD_1}\ln(\frac{D_{ij}}{T\_1_{ij}})] \\
& + \sum_{\substack{i,j \text{ on floor 2} \\ i\neq j}} [F\_2_{ij}(x_i,x_j,y_i,y_j) - K_{MOD_2}\ln(\frac{D_{ij}}{T\_2_{ij}})] \\
& + \ldots + \sum_{\substack{i,j \text{ on floor } K \\ i\neq j}} [F\_K_{ij}(x_i,x_j,y_i,y_j) - K_{MOD_K}\ln(\frac{D_{ij}}{T\_K_{ij}})] \\
& + \sum_{\substack{i,j \text{ on} \\ \text{different floors}}} c^H_{ij}\bar{D}_{ij} \\
\text{s.t. } & \frac{1}{2}w_F \geq x_i + r_i \text{ and } \frac{1}{2}w_F \geq r_i - x_i \text{ for all } i \text{ on floor 1} \\
& \frac{1}{2}h_F \geq y_i + r_i \text{ and } \frac{1}{2}h_F \geq r_i - y_i \text{ for all } i \text{ on floor 1} \\
& \\
& \frac{1}{2}w_F \geq x_i + r_i \text{ and } \frac{1}{2}w_F \geq r_i - x_i \text{ for all } i \text{ on floor 2} \\
& \frac{1}{2}h_F \geq y_i + r_i \text{ and } \frac{1}{2}h_F \geq r_i - y_i \text{ for all } i \text{ on floor 2} \\
& \vdots \\
& \frac{1}{2}w_F \geq x_i + r_i \text{ and } \frac{1}{2}w_F \geq r_i - x_i \text{ for all } i \text{ on floor } K \\
& \frac{1}{2}h_F \geq y_i + r_i \text{ and } \frac{1}{2}h_F \geq r_i - y_i \text{ for all } i \text{ on floor } K \\
& \\
& w_F^{\max} \geq w_F \geq w_F^{\min} \\
& h_F^{\max} \geq h_F \geq h_F^{\min} \\
& \\
& \bar{D}_{ij} \geq \min\left(D_{iE_1} + D_{E_1j},\ D_{iE_2} + D_{E_2j}, \ldots, D_{iE_m} + D_{E_mj}\right) \text{ for all } i,j \text{ not on same floor } (\star),
\end{aligned}
$$

where if departments i and j are both on floor l

$$
F\_l_{ij}(x_i,x_j,y_i,y_j) := \begin{cases} c^H_{ij}z + \frac{t\_l_{ij}}{z} - 1, & z \geq T\_l_{ij} \\ 2\sqrt{c^H_{ij}t\_l_{ij}} - 1, & 0 \leq z < T\_l_{ij} \end{cases}
$$

and recall that $D_{ij} = (x_i - x_j)^2 + (y_i - y_j)^2$ and thus D_{iE_1}, for example, is the square distance between department i and elevator E_1.

The constraints ($\star$) are non-convex. Alternatively, the following convex constraints can be used:

$$\begin{aligned}
&\bar{d}_{ij} \geq \sqrt{D_{iE_1} + D_{E_1 j}} \text{ for all } i, j \text{ on different floors} \\
&\bar{d}_{ij} \geq \sqrt{D_{iE_2} + D_{E_2 j}} \text{ for all } i, j \text{ on different floors} \\
&\vdots \\
&\bar{d}_{ij} \geq \sqrt{D_{iE_K} + D_{E_m j}} \text{ for all } i, j \text{ on different floors,}
\end{aligned}$$

with the objective function modified as:

$$\begin{aligned}
\min_{(x_i,y_i),h_F,w_F} & \sum_{\substack{i,j \text{ on floor } 1 \\ i \neq j}} [F_1_{ij}(x_i, x_j, y_i, y_j) - K_{MOD_1} \ln(\frac{D_{ij}}{T_1_{ij}})] \\
& + \sum_{\substack{i,j \text{ on floor } 2 \\ i \neq j}} [F_2_{ij}(x_i, x_j, y_i, y_j) - K_{MOD_2} \ln(\frac{D_{ij}}{T_2_{ij}})] \\
& + \ldots + \sum_{\substack{i,j \text{ on floor } K \\ i \neq j}} [F_K_{ij}(x_i, x_j, y_i, y_j) - K_{MOD_K} \ln(\frac{D_{ij}}{T_K_{ij}})] \\
& + \sum_{\substack{i,j \text{ on} \\ \text{different floors}}} c_{ij}^H \left(\bar{d}_{ij}\right)^2
\end{aligned}$$

Note that the model with convex constraints would probably yield a solution in which each circle will be not too far away from any elevator. The solution to Multi-ModCoAR need not be exact and although this may not give a uniform distribution of circles, it has the advantage of having convex constraints. The AFS model using the Multi-ModCoAR model with convex constraints, denoted AFS-C, and the one using the Multi-ModCoAR model with non-convex constraints, denoted AFS-NC, are tested and compared in Chapter 5.

Stage 3: Multi-BPL Model

$$\min_{(x_i,y_i),h_F,w_F} \sum_{i\neq j} \left(c_{ij}^H d_{ij}^H + c_{ij}^V d_{ij}^V\right) + \sum_{\substack{i,j \text{ on floor 1} \\ i\neq j}} K_{BPL_1} X_{ij} Y_{ij}$$

$$+ \sum_{\substack{i,j \text{ on floor 2} \\ i\neq j}} K_{BPL_2} X_{ij} Y_{ij} + \ldots + \sum_{\substack{i,j \text{ on floor K} \\ i\neq j}} K_{BPL_K} X_{ij} Y_{ij}$$

$$\begin{aligned}
\text{s.t. } & X_{ij} \geq \frac{1}{2}(w_i + w_j) - |x_i - x_j| \text{ and } Y_{ij} \geq \frac{1}{2}(h_i + h_j) - |y_i - y_j| && \text{for all } i,j \text{ on floor 1} \\
& X_{ij} \geq 0, \quad Y_{ij} \geq 0, \quad \text{and} \quad X_{ij}Y_{ij} = 0 && \text{for all } i,j \text{ on floor 1} \\
& \frac{1}{2}w_F - (x_i + \frac{1}{2}w_i) \geq 0 \text{ and } (x_i - \frac{1}{2}w_i) + \frac{1}{2}w_F \geq 0 && \text{for all } i \text{ on floor 1} \\
& \frac{1}{2}h_F - (y_i + \frac{1}{2}h_i) \geq 0 \text{ and } (y_i - \frac{1}{2}h_i) + \frac{1}{2}h_F \geq 0 && \text{for all } i \text{ on floor 1} \\
\\
& X_{ij} \geq \frac{1}{2}(w_i + w_j) - |x_i - x_j| \text{ and } Y_{ij} \geq \frac{1}{2}(h_i + h_j) - |y_i - y_j| && \text{for all } i,j \text{ on floor 2} \\
& X_{ij} \geq 0, \quad Y_{ij} \geq 0, \quad \text{and} \quad X_{ij}Y_{ij} = 0 && \text{for all } i,j \text{ on floor 2} \\
& \frac{1}{2}w_F - (x_i + \frac{1}{2}w_i) \geq 0 \text{ and } (x_i - \frac{1}{2}w_i) + \frac{1}{2}w_F \geq 0 && \text{for all } i \text{ on floor 2} \\
& \frac{1}{2}h_F - (y_i + \frac{1}{2}h_i) \geq 0 \text{ and } (y_i - \frac{1}{2}h_i) + \frac{1}{2}h_F \geq 0 && \text{for all } i \text{ on floor 2} \\
& \vdots \\
& X_{ij} \geq \frac{1}{2}(w_i + w_j) - |x_i - x_j| \text{ and } Y_{ij} \geq \frac{1}{2}(h_i + h_j) - |y_i - y_j| && \text{for all } i,j \text{ on floor } K \\
& X_{ij} \geq 0, \quad Y_{ij} \geq 0, \quad \text{and} \quad X_{ij}Y_{ij} = 0 && \text{for all } i,j \text{ on floor } K \\
& \frac{1}{2}w_F - (x_i + \frac{1}{2}w_i) \geq 0 \text{ and } (x_i - \frac{1}{2}w_i) + \frac{1}{2}w_F \geq 0 && \text{for all } i \text{ on floor } K \\
& \frac{1}{2}h_F - (y_i + \frac{1}{2}h_i) \geq 0 \text{ and } (y_i - \frac{1}{2}h_i) + \frac{1}{2}h_F \geq 0 && \text{for all } i \text{ on floor } K \\
\\
& w_i h_i = a_i \text{ and } w_i^{\max} \geq w_i \geq w_i^{\min} \text{ and } h_i^{\max} \geq h_i \geq h_i^{\min} && \text{for all } i \\
\\
& d_{ij}^H = \min_{E_m : m=1\ldots,M} (d_{iE_m} + d_{E_m j}) && \text{for all } i,j \text{ on different floors} \\
& d_{ij}^H = d_{ij} && \text{for all } i,j \text{ on same floor} \\
\\
& w_F^{\max} \geq w_F \geq w_F^{\min} \text{ and } \ h_F^{\max} \geq h_F \geq h_F^{\min},
\end{aligned}$$

where d_{ij} is simply the horizontal distance between departments i and j and, for example, d_{iE_m} is the horizontal distance between department i and elevator E_m.

Outline of Three-Stage Multi-Floor Layout Model (AFS)

The general outline of AFS is simply

Solve FAF;
Solve Multi-ModCoAR;
Solve Multi-BPL;

TotalCost=BPLCost;

Chapter 5

Computational Experiments

In this Chapter, we study the computational behavior of the proposed models. We also investigate the choice of parameters, the effect of slack space, and the case of a narrow facility. Both versions of the three-stage multi-floor layout method are tested on several problems using the CPLEX 12.1.0 solver for the first stage and MINOS 5.4 for the remaining stages through the GAMS modeling language. In addition, AFS using Multi-ModCoAR with the convex constraints (AFS-C) will be compared to AFS with the non-convex constraints (AFS-NC). The test data used for the experiments are included in the Appendix.

Each problem requires an initial configuration of departments. The center of each department, (x_i, y_i), is placed at equal intervals around a circle of radius $r = w_F^{max} + h_F^{max}$. So, if there are M departments on a floor, then $x_i = r\cos\theta_i$ and $y_i = r\sin\theta_i$ where $\theta_i = 2\pi(i-1)/M$.

5.1 Choosing Parameters

In this report, the choice of parameters was investigated in an attempt to find a correlation between these values and the quality of the optimal layout. The choice of α_l and the penalty values K_{MOD_l} and K_{BPL_l} are important to the performance of the modified ModCoAR and BPL models in both versions of the proposed three-stage method described in Section 4.3:

- K_{BPL_l} must be chosen large enough so that the method will converge to a feasible layout.
- K_{MOD_l} is a scaling parameter and affects the position of the circles in the layout.
  - A small K_{MOD_l} results in a layout where the circles gather closer to the center of the floor.

  - A large K_{MOD_l} results in a layout where the circles are pushed further against the walls of the floor.

- The parameter α_l appears in the target distances of the modified ModCoAR models ($t_{ij} = \alpha_l(r_i + r_j)^2$), and can be viewed as a fine tuning parameter.

The effect of these parameters is illustrated in Figure 5.1 which is associated with Table 5.1.

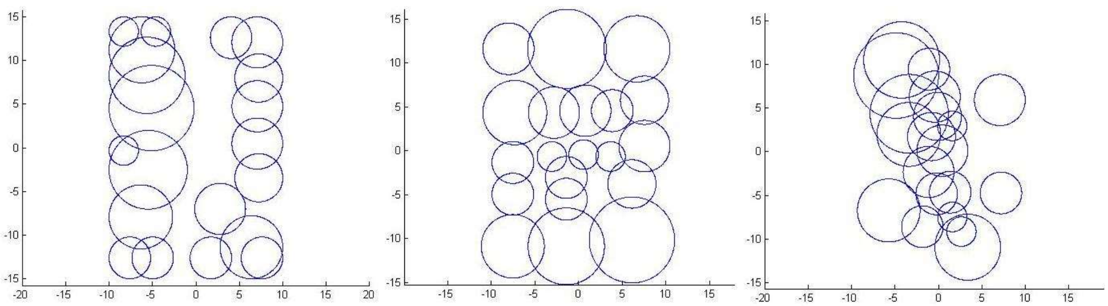

Figure 5.1: Solutions to ModCoAR for the Armour and Buffa 20-department problem.

Table 5.1: Parameters and Costs for Figure 5.1

Figure	K_{MOD}	α	Cost of Final Layout
Left	$5 * \sum_{1 \le i < j \le N} c_{ij}$	7	4772.76
Center	$\sum_{1 \le i < j \le N} c_{ij}$	1.8	4743.20
Right	$0.5 * \sum_{1 \le i < j \le N} c_{ij}$	0.5	4653.63

The method for finding several good layouts in [3] consists of a two step approach. First, K_{MOD} and a large enough K_{BPL} are chosen, then ModCoAR is run for several values of α. The solutions are inspected and the one or two values, say $\bar{\alpha}$, for which a good separation of circles are observed, are chosen. The algorithm is then executed using values $\bar{\alpha} \pm 0.1, 0.2, \ldots, 0.5$, usually resulting in several good layouts.

In the multi-floor case, different parameter values for each floor are often necessary to provide good solutions. Due to the increased number of possible parameter combinations it is even more desirable to have a mathematical method that can determine which penalty values are likely to result in a good final layout. Anjos and Vannelli [3] observed that α had a significant impact on the layouts obtained and, therefore, that the role of α should be the subject of future research.

5.1.1 Investigating The Choice of Parameters K_{MOD} and α

The aim of the ModCoAR model is for $D_{ij} \approx T_{ij}$ at optimality. My hypothesis was that a good approach for choosing parameters would be to choose K_{BPL} large enough, and to adjust K_{MOD} and α in such a way that the average of D_{ij}/T_{ij} would be approximately 1 at optimality in hopes of discovering a correlation between T_{ij} and the quality of the final solution. More precisely, the goal was to be able to adjust the parameters in such a way that, on average, $D_{ij} \approx T_{ij}$, say for parameter values $\bar{\alpha}$ and $\bar{K}_{MOD}$, and then execute the code for the values $\alpha = \bar{\alpha} \pm 0.1, 0.2, \ldots, 0.5$.

Although this method may find good layouts, having the average of D_{ij}/T_{ij} approximately equal to 1 does not appear to be correlated to the best solutions. In fact, often the best solutions are found far away from this point, demonstrating how difficult the multi-floor layout problem really is.

The results obtained using the Armour and Buffa 20-department problem and the 15-department and 3-floor problem were recorded for several values of K_{MOD}. In particular, for each K_{MOD}, the results for every value of alpha between 0.5 and 10 in intervals of 0.1 were recorded. They clearly show that there seems to be no correlation between the generalized target distances and optimal costs. Figure 5.2 illustrates the best solution I found for the Armour and Buffa 20-department problem with aspect ratio constraints of 3.0. Here, $\alpha = 8.0$ and $K_{MOD} = 3 * \sum_{1 \le i < j \le N} c_{ij}$. The average of $\frac{D_{ij}}{T_{ij}}$ is $10.46 \gg 1$ and it can be seen that the circles are not well separated.

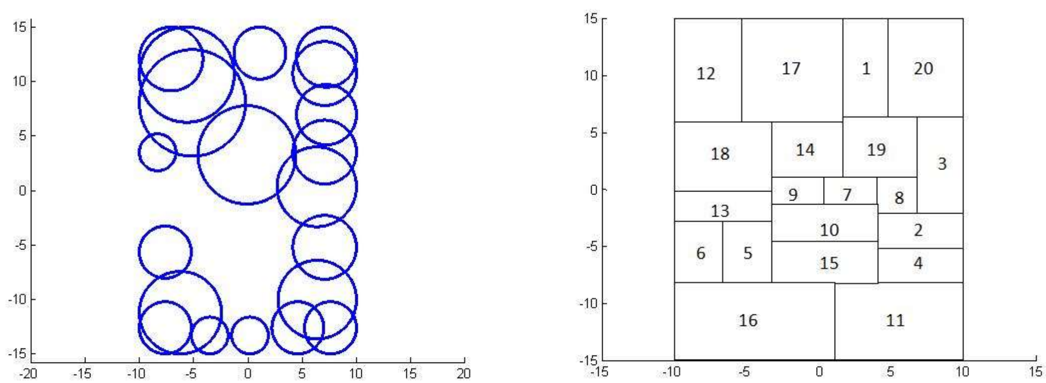

Figure 5.2: Results of ModCoAR and BPL for the lowest cost solution found.

Table 5.2 displays only some of the results obtained where it can be seen that often the best solutions are found far away from the point at which $D_{ij} \approx T_{ij}$ on average. The results in the table are ordered by the average of D_{ij}/T_{ij}, from largest to smallest, so that it can also be seen that as the average of D_{ij}/T_{ij} decreases, the total cost does not necessary decrease as well.

Table 5.2: Generalized Target Distances vs. Total Costs: Armour and Buffa 20-Department Problem

$\beta_i^\star$	K_{MOD}	K_{BPL}	α	Average of D_{ij}/T_{ij}	Total Cost
3	$5 * \sum_{1 \le i < j \le N} c_{ij}$	$\left(\sum_{1 \le i < j \le N} c_{ij}\right)^3$	2.0	24.62	4649.31
3	$\sum_{1 \le i < j \le N} c_{ij}$	$\left(\sum_{1 \le i < j \le N} c_{ij}\right)^3$	1.8	14.36	4743.20
3	$5 * \sum_{1 \le i < j \le N} c_{ij}$	$\left(\sum_{1 \le i < j \le N} c_{ij}\right)^3$	7	13.44	4772.76
3	$3 * \sum_{1 \le i < j \le N} c_{ij}$	$\left(\sum_{1 \le i < j \le N} c_{ij}\right)^3$	5.5	12.96	4531.80
3	$3 * \sum_{1 \le i < j \le N} c_{ij}$	$\left(\sum_{1 \le i < j \le N} c_{ij}\right)^3$	8.0	10.46	4127.80
3	$\sum_{1 \le i < j \le N} c_{ij}$	$\left(\sum_{1 \le i < j \le N} c_{ij}\right)^3$	6	7.7	5003.579
3	$3 * \sum_{1 \le i < j \le N} c_{ij}$	$\left(\sum_{1 \le i < j \le N} c_{ij}\right)^3$	6.54	1.98	5920.42

5.2 Slack Space

The purpose of this section is to investigate the impact of the amount of slack space on the quality of the solution. In particular, would even a small amount of slack increase the quality of the final solution?

To investigate this, I again used the Armour and Buffa 20-department problem, in which there is no slack space to begin with, and varied the amount of slack by expanding and contracting the area of the floor. Some results are given in Tables 5.3 and 5.4. The entries of the table are the costs of the final layout and the columns, labeled 30/20, for example, mean that the facility has a height of 30 and a width of 20. In Table 5.3, $K_{BPL} = (\sum_{1 \le i < j \le N} c_{ij})^2$, $K_{MOD} = 10 * \sum_{1 \le i < j \le N} c_{ij}$, and there are no aspect ratio constraints. In Table 5.4, $K_{BPL} = (\sum_{1 \le i < j \le N} c_{ij})^3$, $K_{MOD} = \sum_{1 \le i < j \le N} c_{ij}$, and the aspect ratio constraint for each department is set to allow for a ratio less than or equal to 3.00. In both tables, the heights and widths of each department are bounded below by 2.

In general, it can be seen that the more slack there is, the more likely it is that the solver will find a feasible solution. It is also clear that there is a relationship between the amount of slack space and the layout cost. With more slack space, even though the solver sometimes yields solutions that are much worse, often it does yield better solutions. For example, in Tables 5.3 and 5.4, the lowest cost solution for each (highlighted in bold) are in the last column, meaning that they were found when there was the greatest amount of slack space.

Table 5.3: Investigating Slack Space on Quality of Solution (no aspect ratio constraints)

α	30/20	30.1/20	30/20.1	30.1/20.1	30.1/20.2	30.2/20.1	30/21	31/21
1	-	-	-	-	-	5207.33	-	4240.29
1.5	4524.04	-	5520.07	-	4440.69	6005.1	4924.23	5023.78
2	-	5403.57	-	5247.08	5488.05	4981.41	4193.11	4720.79
2.5	5199.54	4978.23	5102.46	-	5055.52	5616.95	4878.7	4365.88
3	-	5238.6	4887.29	-	4621.24	5002.43	4625.01	**3755.83**
3.5	-	-	-	4864.42	4721.43	4528.89	5292.97	4458.93
4	5098.8	-	5145.08	-	4199.08	4923.42	4369.46	4836.86
4.5	-	-	5238.37	-	4714.07	4252.66	4033.56	4483.48
5	-	-	5468.68	-	5465.58	5137.25	4280.07	5151.61
slack	0	2	3	5.01	8.02	7.02	30	51

Table 5.4: Investigating Slack Space on Quality of Solution (aspect ratio constraints of 3.0)

α	30/20	30.1/20	30/20.1	30.1/20.1	30.1/20.2	30.2/20.1	30/21	31/21
1	-	-	-	-	4677.23	4677.23	4902.85	4290.77
1.8	4743.196	-	-	-	4559.69	4559.69	4884.44	4850.74
1.5	-	-	-	-	4663.04	4663.04	5326.51	**4282.7**
2	-	-	4829.83	-	-	-	4966.86	4583.9
2.5	-	-	-	-	5427.67	5427.67	5202.12	4607.42
4	-	-	5391.84	-	5092.3	5092.3	5172.92	5007.43
5.1	5585.781	-	4283.31	-	-	-	4470.69	5128.82
5.2	5007.864	-	-	4998.45	4735.8	4735.8	4297.3	4472.14
4.2	5019.872	-	4995.01	5639.81	-	-	4735.48	5241.47
slack	0	2	3	5.01	8.02	7.02	30	51

5.3 Case of a Narrow Facility

When the shape of the facility is narrow, it can occur that the circles are too large to fit in the narrow facility causing the solution of ModCoAR (or of the modified ModCoAR models) to be infeasible. In this case, it is possible to scale all of the radii by the same amount so that the circles fit within the facility and a reasonable layout can be obtained. This is illustrated in Figures 5.3 and 5.4. The figures illustrate the layout of one floor of the 15-department and 3-floor problem solved using the proposed AFS-NC method. The data for this problem is given in Appendix A.2.

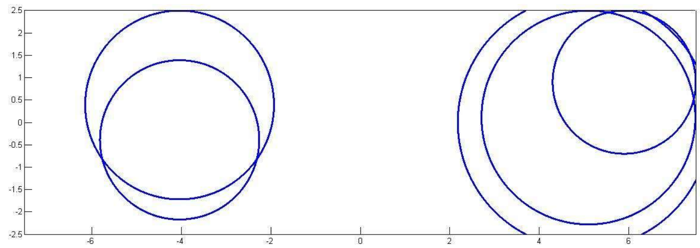

Figure 5.3: Infeasible solution of Multi-ModCoAR before scaling radii.

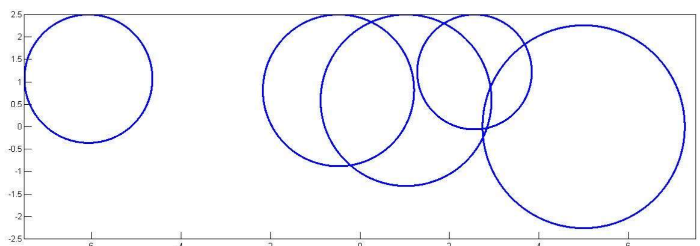

Figure 5.4: Solution of Multi-ModCoAR after radii is scaled by 0.8.

It is also sometimes helpful to scale the radii even when the solution is not infeasible, but the circles are too large for the narrowness of the facility.

5.4 Test Problem 1: 15-Departments and 3-Floors

The first test problem is a 15-department and 3-floor problem used in [5] to demonstrate MULTIPLE. It includes 6 potential lift locations that, from my understanding, are located on the perimeter of the floor and do not take up space. Department 15 is the receiving/shipping department and is thus fixed to the first floor as a 5×5 square in the bottom right hand corner. Also, because of the method used by MULTIPLE, a range of areas are allowed and the departments are not necessarily rectangular in shape. The final layout cost determined by MULTIPLE is \$125, 822.50 with compression and without any shape

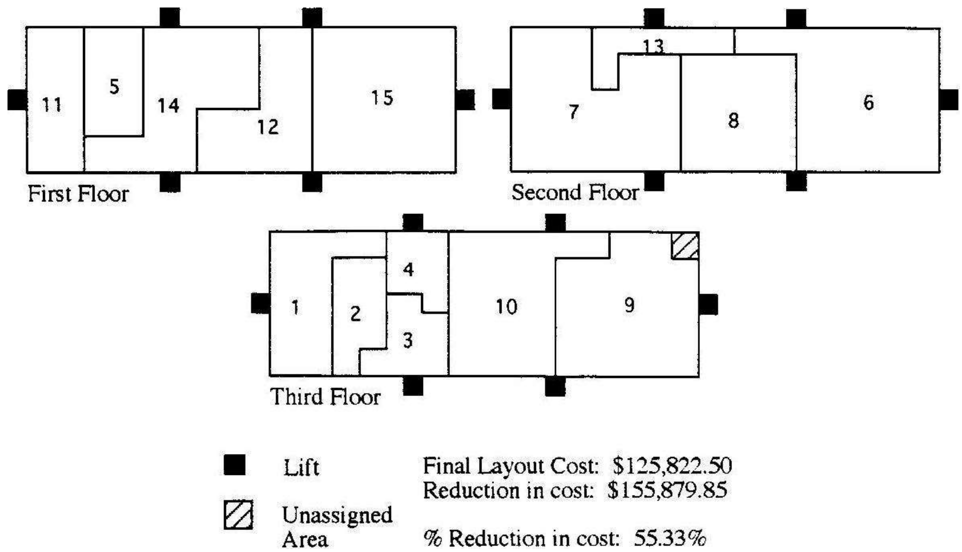

Figure 5.5: Final layout obtained by MULTIPLE. The figure comes from [5].

constraints and is shown in Figure 5.5. This layout was found in 37.9 seconds. With shape constraints and further restrictions, the cost increases.

In order to be able to compare my results with theirs in the best way possible, I determined the areas of each department from MULTIPLE's final layout. To do this, the final layout for each floor shown in Figure 5.5 was divided into a grid of unit squares. The area of a new rectangular department is precisely the number of unit squares occupied by the irregularly shaped department. The exact problem data used is given Appendix A.2.

5.4.1 Application of FBF to Test Problem 1

In order to be able to apply FBF, I first considered the case where only one elevator location is permitted. This location is fixed to the center of the floor and it is assumed that the elevator does not take up space, even though it can easily be made to do so. With the exception of the elevators, the specific data for this 15-department and 3-floor problem is the same as the data given in Appendix A.2.

As is the case for FBF, the layout of each floor l was solved independently. The parameters K_{MOD_l} and α_l were varied to find a reasonable separation of circles in ModCoAR_1, following the method described in Section 5.1. Table 5.5 summarizes the best results of 20 different layouts, including the largest aspect ratios found on each floor, the cost of the final layout on each floor, the vertical cost, the total cost of the final layout, and also the running time over all floors. It is possible to find lower cost solutions more easily if the aspect ratio constraints are relaxed, but they were kept low to ensure realistically shaped departments. Note that the running time that is recorded is the running time for one run of the FBF code and does not take into consideration the time taken to tune the parameters before arriving at the combination of parameters that yields the solution.

Table 5.5: Results of FBF on Test Problem 1

Largest Aspect Ratios (Floor1, Floor2, Floor3)	Cost Floor 1	Cost Floor 2	Cost Floor 3	Vertical Cost	Total Cost	Running Time (sec)
3.56, 3.00, 4.17	9958.91	15226.24	9736.50	86250	121,171.65	4.381
3.13, 3.00, 4.17	7270.63	15226.24	9736.50	86250	118,483.37	4.459
3.13, 3.00, 3.55	7270.63	15226.24	13939.16	86250	122,686.02	4.480
2.50, 3.00, 3.55	10413.64	15226.24	13939.16	86250	125,829.03	4.353

Figure 5.6, illustrates one of the final layouts obtained. It can be seen that the proposed three-stage multi-floor method found a low cost layout with realistic department shapes.

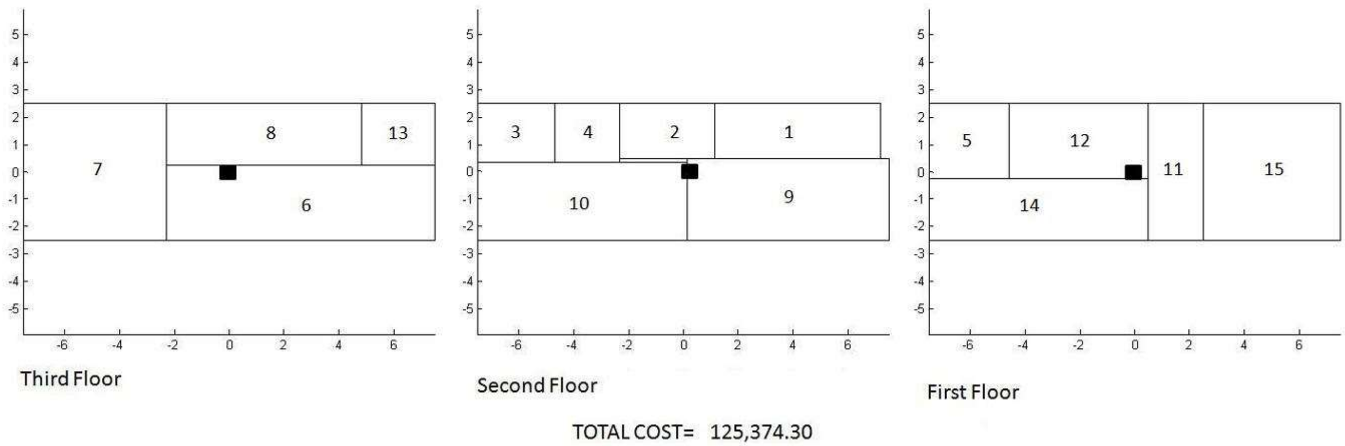

Figure 5.6: Final layout obtained by FBF.

5.4.2 Application of AFS to Test Problem 1

The same problem data was used, again with the exception that only one elevator location is permitted, so that the AFS and FBF methods can be compared. Note that since there is only one elevator location, there is no need to distinguish between AFS-C and AFS-NC.

The proposed AFS method was tested on this data for 20 different combinations of parameter values. Slightly larger aspect ratio constraints were used than in Section 5.4.1 in order to be able to find a few more feasible solutions in 20 trials. One reason for this is that the method for choosing parameters as described in Section 5.1 works differently when the layout of all floors are solved simultaneously because changing the parameters on one floor affects the layout of all floors. The feasible results of these trials are displayed in Table 5.6 and one of these layouts is illustrated in Figure 5.7.

Table 5.6: Results of AFS on Test Problem 1

Largest Aspect Ratios (Floor1, Floor2, Floor3)	Cost Floor 1	Cost Floor 2	Cost Floor 3	Vertical Cost	Total Cost	Running Time (sec)
2.92, 5.2, 6.81	10341.97	15680.99	9033.54	86250	121,306.50	0.523
5.12, 4.35, 4.6	11233.70	14168.26	13452.38	86250	125,104.35	0.569
5.12, 4.52, 6.01	11233.70	14235.26	14494.89	86250	126,213.86	0.552
5.64, 5.66, 4.33	13670.40	15586.55	15400.68	86250	130,907.63	0.516

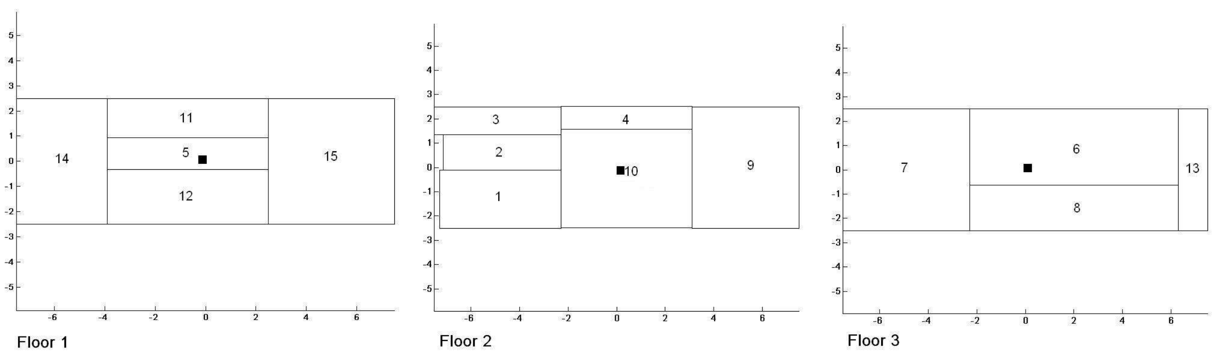

Figure 5.7: Final Layout using AFS with a cost of 125,104.35.

5.4.3 Application of AFS to Test Problem 1- AFS-C vs. AFS-NC

This section compares the proposed AFS-C and AFS-NC methods. Both methods were tested on this 15-department and 3-floor problem (with all 6 elevators) using tight aspect ratio constraints. First, K_{BPL_l}, K_{MOD_l}, and α_l were chosen in such a way that a reasonable solution to Multi-ModCoAR was achieved. Then α_l was increased and decreased by the same increments for both methods to find 30 different layouts. Table 5.7 displays the number of feasible solutions of the 30 trials and also the lowest cost found for each version.

The 4 lowest cost solutions for each method of the 30 trials are given in Tables 5.8 and 5.9. It can be seen that both methods find good solutions without much difference in computational times.

Table 5.7: AFS-C vs. AFS-NC

	AFS-C			AFS-NC		
Aspect Ratio Constraints Floor1, Floor2, Floor3	# of Solutions Found of 30	Lowest Cost Layout	Running Time for Lowest Cost Layout	# of Solutions Found of 30	Lowest Cost Layout	Running Time for Lowest Cost Layout
5,5,5	3	126,936.07	0.929	12	133,181.78	0.930
6,6,6	5	129,866.20	0.908	2	129,393.76	0.912
8,8,8	12	126,754.52	0.926	13	123,501.51	0.870

Table 5.8: Results: AFS-C

Largest Aspect Ratios (Floor1, Floor2, Floor3)	Total Cost	Running Time (sec)
2.66,8.00,4.32	126,754.52	0.926
3.14,4.26,4.32	126,936.07	0.929
7.14,7.15,4.32	128,982.77	0.934
3.14,5.83,4.32	129,866.20	0.908

Table 5.9: Results: AFS-NC

Largest Aspect Ratios (Floor1, Floor2, Floor3)	Total Cost	Running Time (sec)
7.06,4.42,5.76	123,501.53	0.870
8.00,7.05,4.32	124,763.39	0.869
7.14,4.63,3.54	125,269.47	0.848
5.60,8.00,7.26	125,423.63	0.876

5.4.4 Comparing Proposed Methods to MFFLPE

Goetschalckx et al. [10] also use the 15 department, 3 floor, and 6 elevator problem to test their mathematical-programming method, MFFLPE, summarized in Section 2.1.4 of the literature review. They converted the irregularly shaped departments of the problem into rectangular departments of the same area using one of the final layouts obtained from MULTIPLE in [5]. Using their algorithm, they were able to find a “distance score” of 121,419 in a time of 3,911 seconds.

However, there are a few issues with this. First of all, it is not clear what is meant by a “distance score”. Also, their algorithm requires the length of the long and short sides of the departments as input parameters rather than variables. In addition, it is clear from their final layout shown in Figure 5.8 that department 15 is not fixed to the first floor as it should be.

I tested the proposed methods of this report using the same data that Goetschalckx

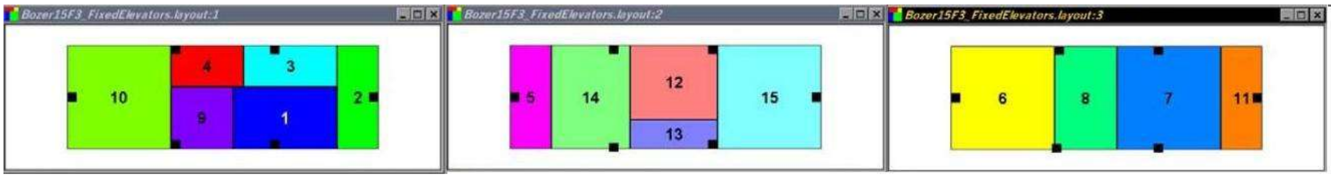

Figure 5.8: The optimal floor layout obtained by Goetschalckx et al. [10]. The figure was taken from [10].

et al. used in [10]. The data is the same as in Section A.2 of the Appendix except that department 15 is not fixed to the first floor and the areas are as given in Table 5.10.

Table 5.10: Fixed Area of Departments

Department	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Area	15	10	9	7	10	25	25	15	9	25	10	15	6	19	25

First, FBF was applied to this data. In this case, it is assumed that there is only one elevator location in the center of the floor. The largest aspect ratio in their results is 2.94. So, the aspect ratio constraints for each department in this experiment were set to allow for ratios less than or equal to 2.94. The final layout is illustrated in Figure 5.9 with a total cost of 106,830.08 and a total running time of 4.534 seconds.

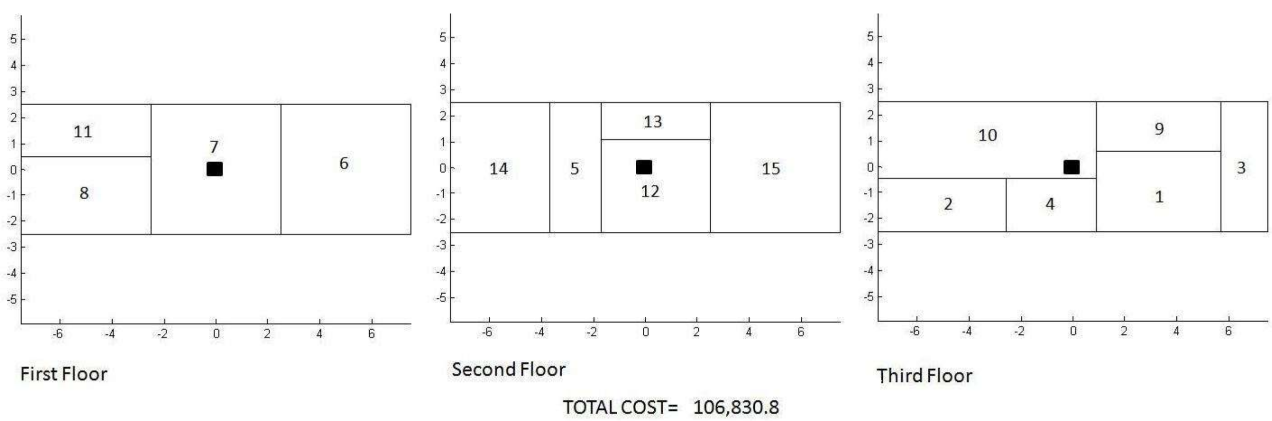

Figure 5.9: The final layout obtained from FBF.

AFS-NC and AFS-C were also applied to this problem, taking all 6 elevators into consideration. The aspect ratio constraints were relaxed since it is more difficult to choose parameters that will yield a feasible solution when the layout of all floors are solved simultaneously. This is partly because changing a parameter on one floor affects the layout of the other floors as well. Another factor could be the multiple penalty terms in the objective function. AFS-NC obtained a layout with a cost of 111,985.07 in 0.893 seconds and with aspect ratios of 4.00, 3.04, and 3.86. AFS-C obtained a layout with a cost of 113,803.39 in 0.950 seconds and with aspect ratios of 4.00, 2.58, and 3.10.

From these results, it is clear that the proposed three-stage multi-floor method yields good solutions and with small running times.

5.4.5 Comparison of Results for Test Problem 1

Although the results of the proposed three-stage multi-floor method cannot be compared directly to the final solution obtained by MULTIPLE because of differences in assumptions and in the solution method, it is clear that FBF, AFS-C, and AFS-NC are able to find low cost layouts with realistically shaped departments. MULTIPLE arrived at the solution illustrated in Figure 5.5 in 37.9 seconds. All versions of the proposed method of this report find several high-quality layouts with running times of at most a few seconds. In particular, the AFS-C and AFS-NC methods find solutions to this problem in a running time of less than 1 second. One must keep in mind, however, that the time that is recorded here is the running time for one run of the FBF, AFS-C, or AFS-NC code and does not take into consideration the time taken to tune the parameters before arriving at the combination of parameters that yields the solution. Since, on average, only a few parameter combinations need to be tested before arriving at a solution, the proposed method can find solutions in a short amount of time. Of course this depends on the tightness of the constraints used.

It is clear that both the FBF and AFS methods can provide good quality solutions in a short amount of time. Although FBF has the disadvantage of allowing for only one elevator location, and it has slightly larger running times, often a high-quality layout can be found in a more systematic way. This is because the layout of each floor is solved independently, so if the user finds that the layout on only one floor is not desirable, for example, the parameters for that floor can be changed without affecting the layout of the other floors.

Both the AFS-C and AFS-NC methods find good solutions without much difference in running times. From the results of Section 5.4.3, it may seem that AFS-NC yields lower cost solutions in general and that AFS-C finds lower cost solutions with lower aspect ratio constraints. However, this may not be the case because these results were highly dependent on the parameters chosen for the particular experiment. It is likely that changing the parameters will find other high-quality solutions.

5.5 Test Problem 2: 40-Departments and 4-Floors

The second data set used is a 40-department, 4-floor, and 3-elevator problem found in [21]. This test problem was designed for methods using grids and spacefilling curves, which is not the case in the proposed method of this report. The grid size is 4.0 square distance units and I interpreted this to mean that each grid square of the floor is of size 2×2. There are 26 squares on the grid for each floor, implying a total area of $4 \times 26 = 104$ square distance units. The inter-floor distance is 2.5 units and it seems as though department 40 is fixed to the first floor as an irregularly shaped department. This is illustrated in Figure 5.10.

There are several issues in comparing my results to the other methods in the literature that also used this test data, explaining why the costs I obtain in my results are much lower than the costs of the others methods. It is my opinion that they cannot be compared directly. I fixed department 40 to the first floor, but I could not fix it in the same irregular shape as can be seen in Figure 5.10. In addition, the shape of the floor itself is irregular, but I assumed that it was a square floor of the same area. Since the grid size is 4.0 and the shape of each department must follow the shape of the grid squares, there is less flexibility in the department shapes. The method of this report allows for rectangular shapes of any length and width, allowing for more flexibility. The specific data used for the experiment is given in Appendix A.3.

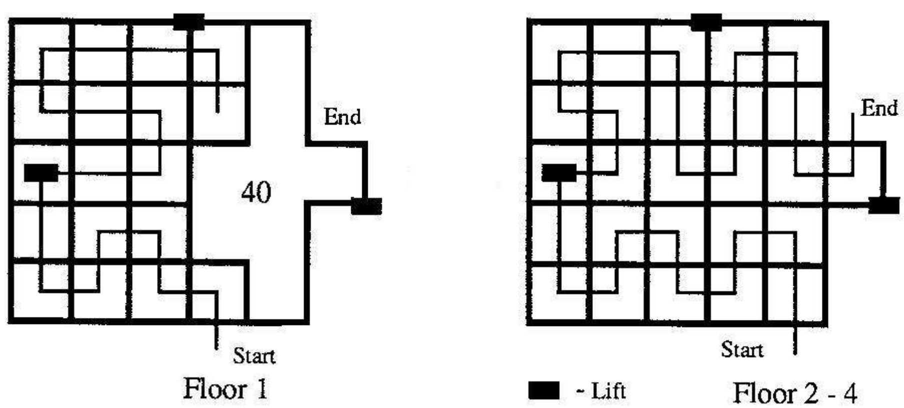

Figure 5.10: The spacefilling curve and fixed department locations for the 40-department and 4-floor problem. Figure comes from [21].

5.5.1 Application of FBF to Test Problem 2

FBF was tested on this problem for the purpose of testing its effectiveness in solving large-scale problem instances. Since FBF can only accommodate one elevator location, the problem data given in Appendix A.3 differs only in that there is one elevator location fixed to the center of the floor.

Results of this experiment found a low cost layout of 14,229.03 in a time of 178.7 seconds and the largest aspect ratios for each floor were 6.50 for floor 1, 2.80 for floor 2, 2.91 for floor 3, and 3.00 for floor 4. Solving ModCoAR_l for each floor accounts for most of the time taken to solve the problem.

Figure 5.11 illustrates another layout obtained that had a cost of 14,622.53. The largest aspect ratios for each floor were 3.31, 2.73, 3.01, 3.01 and this solution was found in a time of 178.9 seconds.

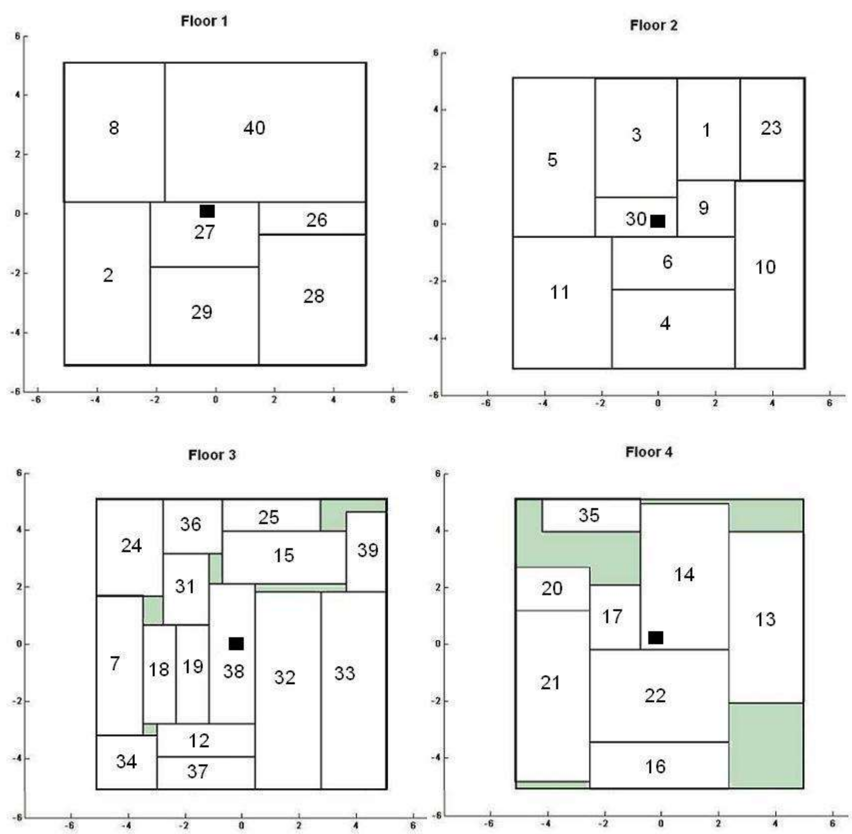

Figure 5.11: Layout obtained using FBF.

5.5.2 Application of AFS to Test Problem 2- AFS-C vs. AFS-NC

AFS-C and AFS-NC are tested on this problem, including all 3 elevators. Some results are given below in Tables 5.11 and 5.12, demonstrating that both methods are capable of finding good solutions to large problems very quickly.

Table 5.11: Results: AFS-C

Largest Aspect Ratios (Floor1, Floor2, Floor3)	Horizontal Cost	Vertical Cost	Total Cost	Running Time (sec)
4.65, 6.50, 10.71, 11.00	9092.27	5562.50	14,654.77	2.82
4.15, 6.50, 5.90, 6.28	9369.99	5562.50	14,932.49	2.80
2.43, 5.78, 6.00, 2.69	9453.50	5562.50	15,016.00	2.73
4.50, 4.96, 5.34, 6.00	9501.68	5562.50	15,064.18	2.72

Table 5.12: Results: AFS-NC

Largest Aspect Ratios (Floor1, Floor2, Floor3)	Horizontal Cost	Vertical Cost	Total Cost	Running Time (sec)
4.68, 2.77, 4.00, 5.00	8975.10	5562.50	14,537.60	2.68
4.48, 4.39, 4.00, 4.00	8959.16	5562.50	14,521.66	2.68
2.74, 3.61, 6.00, 3.69	8815.29	5562.50	14,377.79	2.66
3.47, 7.02, 6.38, 4.84	9633.63	5562.50	15,196.13	2.66

Note that choosing the initial configuration of departments (specifically for this data) using the method given in Chapter 5 results in the Multi-ModCoAR method being badly scaled for some choices of parameters. However, both AFS-C and AFS-NC are still able to find solutions, showing the strength of the methods. To avoid this problem, changes were made to the initial configuration by randomly changing the position of some circles. The initial configuration and the specifics of the problem data are given in Appendix A.3. To avoid the problem of scaling for the application of the AFS-C method, the initial configuration was scaled by 0.3.

5.5.3 Comparison of Results for Test Problem 2

It is clear that each version of the proposed three-stage multi-floor method is capable of finding good layouts even for large problems. Also, both AFS-C and AFS-NC arrive at a solution in only a few seconds compared to the 200 seconds it takes FBF to arrive at a solution using the same problem data, except with only one elevator location. It seems that the difference in running time is due to the formulation of the objective function of ModCoAR_l versus the objective function of Multi-ModCoAR.

5.6 Summary of Computational Experiments

First, the choice of parameters was investigated and although the aim of the ModCoAR model is for $D_{ij} \approx T_{ij}$ at optimality, there seems to be no correlation between T_{ij} and the quality of the final solution. However, the method for choosing parameters by Anjos and Vannelli [3] presented in Section 5.1 resulted in many good layouts for the above computational experiments.

It was seen that scaling the radii of the circles in the ModCoAR and the modified ModCoAR models helps to achieve a reasonable layout of circles in the case of a narrow facility.

The effect of slack space on the quality of the solution was also investigated. In general, the more slack there is, the more likely it is that the solver will find a feasible solution. It is also clear that there is a relationship between the amount of slack space and the layout cost; increasing the slack space often does yield better solutions.

From these test problems, it is clear that both the FBF and AFS methods can provide good quality solutions in a short amount of time. FBF has the disadvantage of allowing for only one elevator location since the layout of the floors are solved independently of one another. It also has larger running times than the AFS methods, but it seems that this is due to the formulation of the objective function of ModCoAR_l versus the objective function of Multi-ModCoAR. On the other hand, FBF has the advantage of allowing the user to find a desirable layout in a more systematic way, often making it easier to find low cost solutions with low aspect ratios.

The AFS methods, AFS-C and AFS-NC, solve the layout of all floors simultaneously and so have the advantage of allowing for multiple elevator locations. By adjusting the parameters α, K_{MOD_l}, K_{BPL_l}, the initial configuration, and aspect ratio constraints, both AFS-C and AFS-NC yield good results and there is not much difference in running times.

Chapter 6

Conclusions and Future Research

In this report, a three-stage multi-floor layout method using mathematical-programming techniques was presented that provides good solutions to the multi-floor facility layout problem. The first stage is a linear mixed-integer program that assigns departments to floors such that the total of the departmental interaction costs between floors is minimized. Subsequent stages find a locally optimal layout for each floor.

Two versions of this method were presented in detail. The first, FBF, solves the layout of each floor independently of the other floors and thus, allows for only one elevator location. The second version, AFS, solves the layouts of all floors simultaneously, allowing for multiple elevator locations. These versions were implemented, tested, and compared to each other and to existing results in the literature through computational experiments. From the results of the experiments, it is clear that both versions can provide high-quality solutions to the multi-floor layout problem even for large problem instances.

Not only does the proposed method achieve very good results, but it also overcomes some limitations that are present in previous methods. For example, departments are not split across floors, multiple elevator locations are allowed with the AFS version, and the running times are small. The method can find several very good layouts for the same problem in a short time by simply changing the parameters α_l, K_{MOD_l}, and K_{BPL_l} as opposed to other methods that find only one optimal layout. Low cost layouts with realistic department shapes can be found by controlling the aspect ratio of each department through constraints.

It is clear that the initial configuration of departments, and the parameters α_l, K_{MOD_l}, and K_{BPL_l} have a large impact on the quality of the final solution. It is fairly easy to find a combination of these values that yield good solutions, but it is not clear ahead of time which combination it will be. The choice of these values could be the subject of future research. Future work could also involve constructing a single-stage method that assigns departments to floors and optimizes the layout on each floor using the same underlying concepts of the proposed method of this report. Lastly, to find the closest elevator for each department in the Multi-BPL model, the “min” function was used, which is believed to be problematic. No problems were noticed in the computational experiments of this report, but replacing this function could be the subject of future research.

APPENDIX

Appendix A

Test Data for Computational Experiments

A.1 Data for the Armour and Buffa 20-Department Problem

- The height of the facility is 30 and the width is 20.

Table A.1: Initial Configuration for Armour and Buffa 20-Department Problem

Department	**Center**	**Department**	**Center**
1	(155.12000000, 0.00000000)	11	(44.13591924,148.70855736)
2	(153.84646694,19.83630532)	12	(24.75713572,53.13163824)
3	(150.04677915,39.34689906)	13	(4.97184071,155.04030186)
4	(143.78332740,58.21141780)	14	(14.89509178,154.40320800)
5	(135.15895738,76.12010667)	15	(-34.51744728,151.23081774)
6	(124.31528102,92.77890549)	16	(-53.57302724,145.57522163)
7	(111.43035119,107.91427725)	17	(-71.74893958,137.52928440)
8	(96.71573806,121.27769956)	18	(-88.74673632,127.22512013)
9	(80.41305520,132.64974540)	19	(-104.28731410,114.83192291)
10	(62.78999219,141.84368608)	20	(-118.11549706,100.55318868)

Table A.2: Costs for Armour and Buffa 20-Department Problem

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	0	1.8	1.2	0	0	0	0	0	0	1.04	1.12	0	0	1.20	0	0	0	0	0	0
2	1.8	0	0.96	24.45	0.78	0	13.95	0	1.20	1.35	0	0	0	0	0	0	0	0	6.90	0
3	1.2	0.96	0	0	0	2.21	0	0	3.15	3.90	0	0	0	13.05	0	0	0	0	13.65	0
4	0	24.45	0	0	1.08	5.70	7.50	0	2.34	0	0	1.40	0	0	0	0	0	1.50	15.75	0
5	0	0.78	0	1.08	0	0	2.25	1.35	0	1.56	0	0	0	0	1.35	0	0	0	0	0
6	0	0	2.21	5.70	0	0	6.15	0	0	0	0	0.45	0	0	0	0	0	1.05	0	0
7	0	13.95	0	7.50	2.25	6.15	0	24.00	0	1.87	0	0	0	0.96	0	0	0	1.65	0	3.75
8	0	0	0	0	1.35	0	24.0	0	0	0	0	0	0.60	0	0	0	0	0	7.50	33.45
9	0	1.20	3.15	2.34	0	0	0	0	0	0	0	0	0	7.50	0	0	7.50	0	0	0
10	1.04	1.35	3.90	0	1.56	0	1.87	0	0	0	0.36	12.0	0	18.6	1.92	0	0	0	5.25	0
11	1.12	0	0	0	0	0	0	0	0	0.36	0	2.25	0	3.00	0.96	22.50	0	0	0	0
12	0	0	0	1.40	0	0.45	0	0	0	12.0	2.25	0	0	0	1.65	0	15.00	0	8.40	0
13	0	0	0	0	0	0	0	0.60	0	0	0	0	0	8.00	1.04	6.00	0	0	0	0
14	1.20	0	13.05	0	0	0	0.96	0	7.50	18.6	3.00	0	8.00	0	9.75	0	0	0.90	0	0
5	0	0	0	0	1.35	0	0	0	0	1.92	0.96	1.65	1.04	9.75	0	0	5.25	0	0	0
16	0	0	0	0	0	0	0	0	0	0	2.25	0	6.00	0	0	0	12.00	0	0	0
17	0	0	0	0	0	0	0	0	7.50	0	0	15.0	0	0	5.25	12.00	0	0	7.50	0
18	0	0	0	1.50	0	1.05	1.65	0	0	0	0	0	0	0.90	0	0	0	0	4.65	0
19	0	6.90	13.65	15.75	0	0	0	7.50	0	5.25	0	8.40	0	0	0	0	7.50	4.65	0	0
20	0	0	0	0	0	0	3.75	33.45	0	0	0	0	0	0	0	0	0	0	0	0

Table A.3: Fixed Area of Departments for Armour and Buffa 20-Department Problem

Department	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Area	27	18	27	18	18	18	9	9	9	24	60	42	18	24	27	75	64	41	27	45

A.2 Data for 15-Department and 3-Floor Problem

- There are 15 departments and 3 floors.
- There are 6 elevator locations. They are located at (-7.5,0), (-2.5,2.5), (2.5,2.5), (7.5,0), (2.5,-2.5), and (-2.5,-2.5).
- Department 15 is the receiving/shipping department and is thus fixed to the first floor as a 5×5 square in the bottom right hand corner.
- Departments are assumed to be rectangular.
- The initial configuration of departments for floor number l were determined from the formulas $x_i = r \cos \theta_i$, $y_i = r \sin \theta_i$, where $r = w_F^{max} + h_F^{max}$ and $\theta_i = 2\pi(i-1)/M$, and M is the number of departments on floor l.
- The maximum area for each floor is 75 where $w_F^{max} = 15$ and $h_F^{max} = 5$.
- The distance between any two adjacent floors is 10.
- The radii is scaled by .8 (narrow facility).

Table A.4: Vertical Cost for 15-Department and 3-Floor Problem

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	5	5	5	5	5	5	5	5	5	5	5	5	5	5	1.25
2	5	5	5	5	5	5	5	5	5	5	5	5	5	5	1.25
3	5	5	5	5	5	5	5	5	5	5	5	5	5	5	1.25
4	5	5	5	5	5	5	5	5	5	5	5	5	5	5	1.25
5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	1.25
6	5	5	5	5	5	5	5	5	5	5	5	5	5	5	1.25
7	5	5	5	5	5	5	5	5	5	5	5	5	5	5	1.25
8	5	5	5	5	5	5	5	5	5	5	5	5	5	5	1.25
9	5	5	5	5	5	5	5	5	5	5	5	5	5	5	1.25
10	5	5	5	5	5	5	5	5	5	5	5	5	5	5	1.25
11	5	5	5	5	5	5	5	5	5	5	5	5	5	5	1.25
12	5	5	5	5	5	5	5	5	5	5	5	5	5	5	1.25
13	5	5	5	5	5	5	5	5	5	5	5	5	5	5	1.25
14	5	5	5	5	5	5	5	5	5	5	5	5	5	5	1.25
15	1.25	1.25	1.25	1.25	1.25	1.25	1.25	1.25	1.25	1.25	1.25	1.25	1.25	1.25	1.25

Table A.5: Flow Matrix for 15-Department and 3-Floor Problem

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	240
2	240	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	1200	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	1200	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0	0	600	0
6	0	0	0	0	0	0	0	480	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	480	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	120
9	0	0	0	0	0	0	0	0	0	600	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0	600	0	0	0
11	0	0	0	0	0	0	480	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	600
13	0	0	0	0	0	0	480	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	600	0	0	0
15	0	10	25	0	25	40	0	0	25	0	40	0	20	0	0

Table A.6: Horizontal Costs for 15-Department and 3-Floor Problem

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	0.25
2	1	0	1	1	1	1	1	1	1	1	1	1	1	1	0.25
3	1	1	0	1	1	1	1	1	1	1	1	1	1	1	0.25
4	1	1	1	0	1	1	1	1	1	1	1	1	1	1	0.25
5	1	1	1	1	0	1	1	1	1	1	1	1	1	1	0.25
6	1	1	1	1	1	0	1	1	1	1	1	1	1	1	0.25
7	1	1	1	1	1	1	0	1	1	1	1	1	1	1	0.25
8	1	1	1	1	1	1	1	0	1	1	1	1	1	1	0.25
9	1	1	1	1	1	1	1	1	0	1	1	1	1	1	0.25
10	1	1	1	1	1	1	1	1	1	0	1	1	1	1	0.25
11	1	1	1	1	1	1	1	1	1	1	0	1	1	1	0.25
12	1	1	1	1	1	1	1	1	1	1	1	0	1	1	0.25
13	1	1	1	1	1	1	1	1	1	1	1	1	0	1	0.25
14	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0.25
15	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0

Table A.7: Fixed Area of Departments for 15-Department and 3-Floor Problem

Department	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Area	12	7	6	5	8	27	26	16	22	22	10	14	6	18	25

A.3 Data for 40-Department and 4-Floor Problem

- There are 40 departments and 4 floors.
- There are 3 elevators locations given by $(-\sqrt{26}, 0)$, $(0, \sqrt{26})$, and $(\sqrt{26}, 0)$.
- $c_{ij}^H = 1$ and $c_{ij}^V = 5$ for all $i \neq j$.
- The distance between adjacent floors is 25 distance units.
- The area of each floor is 104.
- Department 40 is fixed to the first floor.

Table A.8: Flow Data for 40-Department and 4-Floor Problem

From→To Department	Flow	From→To Department	Flow	From→To Department	Flow	From→To Department	Flow
1→3	66.25	5→15	10	16→17	10	32→33	25
1→5	101.25	5→23	6	17→13	5	33→34	25
1→9	40	5→34	25	18→19	5	34→2	12.5
1→12	4	6→11	20	19→3	5	36→37	10
1→30	25	6→23	12	20→21	15	37→32	25
1→36	12.5	6→29	17.5	21→22	15	38→39	25
1→38	10	7→24	15	22→23	7.5	39→33	25
2→40	23.75	8→2	17.5	23→8	7.5	40→1	149.5
3→4	70	8→40	15	24→25	7.5	40→7	7.5
4→10	40	9→10	40	25→22	7.5	40→16	4
4→14	5	10→11	16	26→27	17.5	40→18	4
4→25	7.5	11→2	16	27→28	8.75	40→20	6
4→28	17.5	12→13	2	28→29	8.75	40→26	7
4→33	25	13→14	10	29→8	8.75		
5→6	78.75	14→15	5	30→31	12.5		
5→11	40	15→2	4	31→32	12.5		

Table A.9: Initial Configuration for 40-Department and 4-Floor Problem

Department	**Center**	**Department**	**Center**
1	(10.000000 ,0.000000)	21	(-10.000000, 0.000000)
2	(9.876883, 1.564345)	22	(-9.876883, -1.564345)
3	(9.510565, 3.090170)	23	(-9.510565, -3.090170)
4	(8.910065, 4.539905)	24	(-8.910065, -4.539905)
5	(8.090170, 5.877853)	25	(-8.090170, -5.877853)
6	(7.071068, 7.071068)	26	(-7.071068, -7.071068)
7	(5.877853, 8.090170)	27	(-5.877853, -8.090170)
8	(4.539905, 8.910065)	28	(-4.539905, -8.910065)
9	(3.090170, 9.510565)	29	(-3.090170, -9.510565)
10	(1.564345, 9.876883)	30	(-1.564345, -9.876883)
11	(0.000000, 10.000000)	31	(-0.000000, -10.000000)
12	(-1.564345, 9.876883)	32	(1.564345, -9.876883)
13	(-3.090170, 9.510565)	33	(3.090170, -9.510565)
14	(-4.539905, 8.910065)	34	(4.539905, -8.910065)
15	(-5.877853, 8.090170)	35	(5.877853, -8.090170)
16	(-7.071068, 7.071068)	36	(7.071068, -7.071068)
17	(-8.090170, 5.877853)	37	(8.090170, -5.877853)
18	(-8.910065, 4.539905)	38	(8.910065, -4.539905)
19	(-9.510565, 3.090170)	39	(9.510565, -3.090170)
20	(-9.876883, 1.564345)	40	(9.876883, -1.564345)

Table A.10: Fixed Area of Departments for 40-Department and 4-Floor Problem

Department	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Area	8	16	12	12	16	8	8	16	4	16	16	4	16	16	8	8	4	4	4	4
Department	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40
Area	16	16	8	8	4	4	8	16	12	4	4	16	16	4	4	4	4	8	4	32

Note that in order to deal with one of the methods being badly scaled for this problem, the initial configuration for the problem was not obtained by using the method described in 5, rather the position of some circles were changed randomly.

Bibliography

[1] Sue Abdinnour-Helm and Scott W. Hadley. Tabu search based heuristics for multi-floor facility layout. *International Journal of Production Research*, 38(2):365–383, 2000. 13, 14, 24

[2] Miguel F. Anjos and Anthony Vannelli. An attractor-repeller approach to floorplanning. *Mathematical Methods of Operations Research*, 56:3–27, 2002. 19, 20

[3] Miguel F. Anjos and Anthony Vannelli. A new mathematical-programming framework for facility-layout design. *INFORMS J. on Computing*, 18(1):111–118, 2006. ix, 1, 2, 15, 16, 17, 18, 19, 20, 21, 22, 23, 28, 38, 52

[4] Yavuz A. Bozer and Russell D. Meller. A construction-type layout algorithm for multi-floor facilities with capacitated lifts. Technical report, University of Michigan and Auburn University, 1992. 11, 12, 13

[5] Yavuz A. Bozer, Russell D. Meller, and Steven J. Erlebacher. An improvement-type layout algorithm for single and multiple-floor facilities. *Manage. Sci.*, 40(7):918–932, 1994. ix, 4, 5, 7, 24, 42, 43, 46

[6] Elwood S. Buffa, Gordon C. Armour, and Thomas E. Vollmann. Allocating facilities with CRAFT. *Harvard Business Review*, 42(2):136 – 158, 1964. 3

[7] Chir-Ho Chang, Jin-Ling Lin, and Huei-Jiun Lin. Multiple-floor facility layout design with aisle construction. *IEMS*, 5:1–10, 2006. 7

[8] Zvi Drezner. DISCON: A new method for the layout problem. *Operations Research*, 28(6):1375–1384, 1980. 15, 16

[9] Thomas A. Feo and Mauricio G.C. Resende. Greedy randomized adaptive search procedures. *Journal of Global Optimization*, 6:109–133, 1995. 13

[10] Marc Goetschalckx and Takashi Irohara. Efficient formulations for the multi-floor facility layout problem with elevators. *School of Industrial and Systems Engineering Research Report*, 2007. ix, 9, 46, 47

[11] Eric S. Grajo and Yavuz A. Bozer. LayOPT: A data driven facilities layout optimization system for engineers. http://www.pmcorp.com/PublishedPapers/Simulation% 20Publications/Sim-Layout,% 20Material% 20Handling% 20and% 20Warehousing/LayOPTADataDrivenFacilities.pdf. 5

[12] Peter Hahn, J. MacGregor Smith, and Yi-Rong Zhu. The multi-story space assignment problem. *Annals of Operations Research*, 2008. 1, 9

[13] Roger V. Johnson. SPACECRAFT for multi-floor layout planning. *Management Science*, 28(4):407–417, 1982. 4

[14] Bharat K. Kaku, Thomas E. Morton, and Gerald L. Thompson. A heuristic method for the facilities layout problem. Working paper, 1987. 11

[15] Bharat K. Kaku, Gerald L. Thompson, and Ilker Baybars. A heuristic method for the multi-story layout problem. *European Journal of Operational Research*, 37(3):384 – 397, 1988. 10, 11

[16] Jasmit Kochhar and Sunderesh S. Heragu. MULTI-HOPE: A tool for multiple floor layout problems. *International Journal of Production Research*, 36:3421–3435(15), 1998. 6, 7, 10

[17] Kyu-Yeul Lee, Myung-Il Roh, and Hyuk-Su Jeong. An improved genetic algorithm for multi-floor facility layout problems having inner structure walls and passages. *Computers & Operations Research*, 32(4):879 – 899, 2005. 7

[18] Robin S. Liggett and William J. Mitchell. Optimal space planning in practice. *Computer-Aided Design*, 13(5):277 – 288, 1981. Special Issue Design Optimization. 8

[19] Zhi-Quan Luo, Jong-Shi Pang, and Daniel Ralph. *Mathematical Programs with Equilibrium Constraints*. Cambridge University Press, Cambridge, UK, 1996. 22

[20] Kenichiro Matsuzaki, Takashi Irohara, and Kazuho Yoshimoto. Heuristic algorithm to solve the multi-floor layout problem with the consideration of elevator utilization. *Computers & Industrial Engineering*, 36(2):487 – 502, 1999. 7

[21] Russell D. Meller and Yavuz A. Bozer. Solving the facility layout problem with simulated annealing. Technical Report 91-20, Department of Industrial & Operations Engineering, University of Michigan, 1991. ix, 49

[22] Russell D. Meller and Yavuz A. Bozer. A new simulated annealing algorithm for the facility layout problem. *International Journal of Production Research*, 34(6):1675–1692, 1996. 5, 6, 24

[23] Russell D. Meller and Yavuz A. Bozer. Alternative approaches to solve the multi-floor facility layout problem. *Journal of Manufacturing Systems*, 16(3):192 – 203, 1997. 2, 9, 10, 12, 13, 24, 25, 26, 27

[24] Russell D. Meller and Kai-Yin Gau. The facility layout problem: Recent and emerging trends and perspectives. *Journal of Manufacturing Systems*, 15(5):351 – 366, 1996. 4

[25] Jerrold M. Seehof, Wayne O. Evans, James W. Friederichs, and James J. Quigley. Automated facilities layout programs. In *Proceedings of the 1966 21st national conference*, pages 191–199, New York, NY, USA, 1966. ACM. 11

[26] Drew J. van Camp, Michael W. Carter, and Anthony Vannelli. A nonlinear optimization approach for solving facility layout problems. *European Journal of Operational Research*, 57(2):174–189, March 1992. 16