

Graduate Theses, Dissertations, and Problem Reports

2008

Heuristics for the dynamic facility layout problem with unequal area departments

Artak Hakobyan West Virginia University

Follow this and additional works at: https://researchrepository.wvu.edu/etd

Recommended Citation

Hakobyan, Artak, "Heuristics for the dynamic facility layout problem with unequal area departments" (2008). *Graduate Theses, Dissertations, and Problem Reports.* 4376. https://researchrepository.wvu.edu/etd/4376

This Dissertation is protected by copyright and/or related rights. It has been brought to you by the The Research Repository @ WVU with permission from the rights-holder(s). You are free to use this Dissertation in any way that is permitted by the copyright and related rights legislation that applies to your use. For other uses you must obtain permission from the rights-holder(s) directly, unless additional rights are indicated by a Creative Commons license in the record and/ or on the work itself. This Dissertation has been accepted for inclusion in WVU Graduate Theses, Dissertations, and Problem Reports collection by an authorized administrator of The Research Repository @ WVU. For more information, please contact researchrepository@mail.wvu.edu.

HEURISTICS FOR THE DYNAMIC FACILITY LAYOUT PROBLEM

WITH UNEQUAL AREA DEPARTMENTS

Artak Hakobyan

Dissertation Submitted to the College of Engineering and Mineral Resources at West Virginia University in partial fulfillment of the requirements for the degree of

> Doctor of Philosophy in Industrial Engineering

Alan R. McKendall, Jr., Ph.D., Chair Wafik H. Iskander, Ph.D. Majid Jaraiedi, Ph.D. Robert C. Creese, Ph.D. Robert Mnatsakanov, Ph.D.

Department of Industrial & Management Systems Engineering

Morgantown, West Virginia 2008

Keywords: Dynamic facility layout problem, Unequal area departments, Tabu search, Genetic search, Memetic algorithm, Dual simplex method, Meta-heuristics

ABSTRACT

HEURISTICS FOR THE DYNAMIC FACILITY LAYOUT PROBLEM WITH UNEQUAL AREA DEPARTMENTS

Artak Hakobyan

The facility layout problem (FLP) is a well researched problem of finding positions of departments on a plant floor such that departments do not overlap and some objective(s) is (are) optimized. In this dissertation, the FLP with unequal area rectangular shaped departments is considered, when material flows between departments change during the planning horizon. This problem is known as the dynamic FLP. The change in material flows between pairs of departments in consecutive periods may require rearrangements of departments during the planning horizon in order to keep material handling costs low. The objective of our problem is to minimize the sum of the material handling and rearrangement costs. Because of the combinatorial structure of the problem, only small sized problems can be solved in reasonable time using exact techniques. As a result, construction and improvement heuristics are developed for the proposed problem. The construction algorithms are boundary search heuristics as well as a dual simplex method, and the improvement heuristics are tabu search and memetic heuristics with boundary search and dual simplex (linear programming model) techniques. The heuristics were tested on a generated data set as well as some instances from the literature. In summary, the memetic heuristic with the boundary search technique out-performed the other techniques with respect to solution quality.

DEDICATION

I dedicate this work to my wife Nelli Harutyunyan, my parents Samvel Hakobyan and Armenuhi Darbinyan; my brother Haik Hakobyan, his wife Lena, and their wonderful kids, Tigran and Anna.

AKNOWLEDGMENTS

First of all, I would like to thank Dr. McKendall, for all the time that he spent on this dissertation, and for all the help and input that he provided. Without Dr. McKendall's experience and knowledge, this dissertation would have never been completed. I learned a lot from him, not only in the area of research. Also I would like to thank the family of Dr. McKendall for all the off work hours that he spent, working on this dissertation.

I would like to thank my committee member Dr. Robert C. Creese, Dr. Wafik H. Iskander, Dr. Majid Jaraiedi, and Dr. Robert Mnatsakanov for the useful comments and suggestions they provided and for the time they spent on this dissertation. In addition I would like to thank Dr. Myers, for helping me at the beginning of my studies.

I would like to thank my wife Nelli Harutyunyan, who was with me all these years. I thank her for being an understanding and wonderful person. In addition, I thank my parents, Samvel Hakobyan and Armenuhi Darbinyan, and my brother Haik Hakobyan for their support.

TABLE OF CONTENTS

ABSTRACTii
DEDICATIONiii
AKNOWLEDGMENTSiv
TABLE OF CONTENTSv
LIST OF FIGURESviii
LIST OF TABLESx
LIST OF ACRONYMSxii
CHAPTER 11
INTRODUCTION1
1.1 Static Facility Layout Problem with Equal Area Departments2
1.2 Static Facility Layout Problem with Unequal Area Departments
1.2.1 Objective Function
1.2.2 Discrete versus Continuous Approach
1.2.3 Fixed/Variable Shape Departments
1.3 The Dynamic Facility Layout Problem9
CHAPTER 215
PROBLEM STATEMENT15
2.1 Statement of the Problem15
2.2 Problem Assumptions15
2.3 Research Objectives
CHAPTER 317

LITERATURE REVIEW	17
3.1. Discrete Representation of the FLP	18
3.1.1 Static Facility Layout Problem	18
3.1.2 Dynamic Facility Layout Problem	24
3.2. Continuous Representation of the FLP	26
3.2.1 Static Facility Layout Problem	26
3.2.2 Dynamic Facility Layout Problem	34
CHAPTER 4	37
METHODOLOGY	37
4.1 Introduction	37
4.2 Exact Method	
4.3 Construction Algorithms	41
4.3.1 Boundary Search Heuristic (BSH)	42
4.3.1.1 Solution Representation	42
4.3.1.2 BSH Parameters	
4.3.1.3 Construct Layout Plan Using π	43
4.3.1.3.1 The Construction of π	43
4.3.1.3.2 Finding Candidate Locations for <i>i_curr</i> in period <i>t_curr</i>	45
4.3.1.4 BSH Heuristic	
4.3.2 Dual Simplex Technique	
4.4 Improvement Algorithms	
4.4.1 TS Heuristics	
4.4.1.1 TS/BSH	56

4.4.1.2 TS/DUAL	62
4.4.2 Memetic Heuristic	69
4.4.2.1 The Chromosomes Used by the Memetic Heuristic	72
4.4.2.1.1 Type 1 Chromosomes	72
4.4.2.1.2 Type 2 Chromosomes	76
4.4.2.2 The Pseudo-code for the Memetic Heuristic	77
CHAPTER 5	80
COMPUTATIONAL RESULTS	80
5.1 Datasets	80
5.2 Parameter Settings	82
5.2.1 Parameter Settings for the TS Heuristics	82
5.2.2 Parameter Settings for the Memetic Heuristics	83
5.3 Test Environment	85
5.4 Experimental Results	85
CHAPTER 6	95
CONCLUSIONS	95
6.1 Summary of Research	
6.2 Future Research	96
REFERENCES	97
Appendix A. Problem Instance Used to Demonstrate the BSHs	104
Appendix B. Problem Instance Used to Demonstrate the Dual Simplex	Based
Heuristics	106
Appendix C. LP Formulation Used by the Dual Simplex Based Heuris	tics107

LIST OF FIGURES

Figure 1.1	Euclidean, rectilinear, and actual path distance metrics4
Figure 1.2	Graphical representation of two solutions using the discrete representation of
	the FLP with unequal-area departments
Figure 1.3	Two solutions to the unequal area FLP corresponding to the data given in
	Tables 1.1 and 1.2
Figure 1.4	Shapes of department 6 corresponding to extreme values of aspect ratios9
Figure 1.5	Block layout of the DFLP instance obtained by considering both material
	handling and rearrangement costs
Figure 1.6	Block layout of the DFLP instance obtained by considering only material
	handling costs14
Figure 3.1	The layouts corresponding to sequences of departments
	(1, 2, 3, 4, 5, 6) and (1, 5, 3, 4, 2, 6)23
Figure 3.2	Slicing tree from Tam (1992-a)
Figure 3.3	Layout constructed from the slicing tree in Figure 3.2
Figure 3.4	The layout corresponding to the permutation (12, 4, 9, 20, 11, 13, 2,
	18, 16, 9, 13, 8, 14, 6, 1, 5, 17, 7, 10, 15) and bay break
	points (4, 7, 9, 14, 16)
Figure 3.5	Path of movement of a department along the cluster boundary
Figure 3.6	Possible department configurations
Figure 4.1	Example of boundary segments in a partially constructed layout plan46

Figure 4.2	Finding the best position for placing department 11 in period 2 along the					
	vertical segment (y, l)	19				
Figure 4.3	Layout generated by the BSH	54				
Figure 4.4	Layout generated by the dual simplex method	54				
Figure 4.5	Layout plans: (a) before move (1) is performed; (b) after move (1) is					
	performed	;9				
Figure 4.6	Layouts plans: (a) before move (1); (b) after performing move (1) move,					
	exchanging departments 2 and 3 in period 16	5				
Figure 4.7	Layouts plans: (a) before performing move (2); (b) after performing move (2),				
	relocating department 1 in period 36	7				
Figure 4.8	Layout plans: (a) generated using chromosome in Table 4.6(a); (b) generated	l				
	using chromosome in Table 4.6(b)7	4				
Figure 4.9	Applying crossover operation to parent chromosomes π^1 and π^2	5				

LIST OF TABLES

Table 1.1	Areas, aspect ratios, and orientations of departments	.7
Table 1.2	Two sets of lengths and widths corresponding to data in Table1.1	8
Table 1.3	Department dimensions for the DFLP instance1	2
Table 1.4	Costs to transport materials per distance unit between departments	
	in periods 1 and 21	3
Table 1.5	Rearrangement costs for the DFLP instance for period 21	3
Table 1.6	Costs of solutions in Figures 1.5 and 1.61	4
Table 3.1	Geometric constraints for departments from Tam (1992-a)2	7
Table 4.1	The department period pairs in the vector π generated by the BSH for the	
	problem instance in Appendix A4	4
Table 4.2	Evaluating the position of department 11 in period 3, and the positions	
	identified along vertical segment (l, y) in Figure 4.2	0
Table 4.3	The values of r_p_{tij} , h_{ti} , r_{ti} variables set using the layout plan in	
	Figure 4.3	4
Table 4.4	Solutions: (a) before move (1) is performed; (b) after move (1) is	
	performed5	9
Table 4.5	The values r_p_{tij} , h_{ti} , and r_{ti} : (a) corresponding to current layout plan; (b) after	
	moving to a new solution by exchanging departments 2 and 3 in	
	Period 16	5
Table 4.6	Two type 1 chromosomes for the problem instance in Appendix A:	
	(a) generated using technique 1; (b) generated using technique (2)7	3

Table 4.7	Type 2 chromosome (μ)	.76
Table 5.1	Datasets: (a) from Dunker et al. (2005); (b) generated from data set in	
	Balakrishnan <i>et al.</i> (2000)	.81
Table 5.2	Parameter settings for TS heuristics	83
Table 5.3	Parameter Settings for memetic heuristics	.85
Table 5.4	Summary of TS/BSH and MEM/BSH heuristic results	.89
Table 5.5	Summary of TS/BSH and MEM/BSH heuristic execution times in	
1	ninutes	.90
Table 5.6	Summary of TS/DUAL and MEM/DUAL heuristic results	.91
Table 5.7	Summary of TS/DUAL and MEM/DUAL heuristic execution times in	
	minutes	.92
Table 5.8	Summary of the results of metaheuristics	.93
Table 5.9	Results of the proposed heuristics, and the dynamic genetic algorithm by	
	Dunker et al. (2005)	.94
Table 5.10	D Execution times in seconds, during which the proposed techniques found	
	better solutions than the best solution obtained by Dunker et al. (2005) on	
	problem instance P12 in dataset 1	.94
Table A.1	Shorter and longer side lengths of departments1	04
Table A.2	Cost to transport materials a unit distance between departments1	05
Table B.1	Shorter and longer side lengths of departments1	06
Table B.2	Cost to transport materials a unit distance between departments1	06

LIST OF ACRONYMS

- ACO Ant Colony Optimization
- BSH Boundary Search Heuristic
- CBA Cluster Boundary Algorithm
- DFLP Dynamic Facility Layout Problem
- I/O Input and Output
- FLP Facility Layout Problem
- GA Genetic Algorithm
- HAS Hybrid Ant System
- LP Linear Program
- MA Memetic Algorithm
- MEM/BSH Memetic Search Heuristic Based on BSH
- MEM/DUAL Memetic Search Heuristic Based on dual simplex technique
- MILP Mixed Integer Linear Program
- OFV Objective Function Value
- PTS Probabilistic Tabu Search
- QAP Quadratic Assignment Problem
- RPP Rectangle Packing Problem
- SA Simulated Annealing
- SFC Space Filling Curve
- TS Tabu Search

TS/BSH Tabu Search Based on BSH

TS/DUAL Tabu Search Based on dual simplex technique

VLSI Very Large Scale Integration

CHAPTER 1

INTRODUCTION

The facility layout problem (FLP) is a well researched problem of finding positions of departments such that departments do not overlap and some objective(s) is (are) optimized. Among the objectives which are considered in the literature are one or combinations of the following: minimizing costs to transport materials between departments (material handling costs), maximizing some adjacency measure (see **Houshyar and White, 1993** as well as **Wascher and Merker, 1997**), minimizing the time materials travel between departments, minimizing the area of the smallest rectangle enclosing all the departments, maximizing worker safety, and minimizing the costs of assigning departments to locations (assignment cost). According to **Tompkins** *et al.* **(1996)** material handling costs account for 15-70% of the overall operating expenses within manufacturing system, thus a good layout contributes to substantial reduction in costs. In this research, minimizing the sum of material handling costs is considered.

The FLP is related to other problems such as the rectangle packing problem (see **Ahmad** *et al.*, **2006** as well as **Liu and Teng**, **1999**) and the problem of generating very large scale integrated (VLSI) macro-cell layouts (see **Lengauer**, **1990**, **Cohoon** *et al.*, **1991**, **Sherwani**, **1993**, as well as **Schnecke and Vornberger**, **1997**). In the rectangle packing problem (RPP), the task is to assign rectangles to positions (without overlapping) in a rectangular packing space such that space utilization is maximized (i.e., minimizing the area of the rectangle enclosing all the rectangles). The major difference between the RPP and FLP is that RPP does not consider flows between pairs of rectangles. However,

flows between pairs of departments are used to determine material handling cost in the FLP.

The designing of the VLSI macro-cell layouts is a process of laying out the macrocells on a circuit board. The cells have terminals (pins) which are connected to wire nets on the circuit board through which the electric signals travel between the cells. The objectives considered in the literature are: minimizing the area occupied by the cells, minimizing the total length of the wire used on the circuit, minimizing the total distance the electronic signal travels between the cells. The problem of the VLSI layout generation is very closely related to the FLP since the electrical signal that travels between the macro-cells may be thought of as material flow between departments as in the FLP.

Although some authors considered the FLP in which some departments are required to have pre-specified non-rectangular shapes (see **McKendall** *et al.*, **1999**), the most common approach is to assume that the departments have rectangular shapes. In this dissertation, departments are assumed to have rectangular shapes. The following sections review different models of the FLP. In Section 1.1, the FLP with equal area departments is presented, and the FLP with unequal area departments is presented in Section 1.2. The problem in which material flows between departments change during a multi-period planning horizon (dynamic FLP) is presented in Section 1.3.

1.1 Static Facility Layout Problem with Equal Area Departments

The simplest case of the FLP is a FLP with equal size departments where the amounts of materials flowing between pairs of departments do not change during the planning horizon. This problem is called the static FLP with equal size departments, and it was modeled by **Koopmans and Beckmann (1957)** as a quadratic assignment problem (QAP). In this model, the plant floor is divided into grids of equal size rectangles (locations). Then the FLP becomes the assignment of departments to locations such that no two departments are assigned to the same locations, and the sum of the material handling and assignment costs is minimized.

1.2 Static Facility Layout Problem with Unequal Area Departments

1.2.1 Objective Function

For the static FLP with unequal area departments, departments should be laid out within the boundaries of the plant floor in such a manner that they do not overlap. The most commonly used objective is minimizing material handling cost which is the sum of the product of the flows, distances, and transportation cost per unit per distance unit for each pair of departments.

The Euclidean, rectilinear, or actual path distance metric is used to determine the distances materials flow from the output (pickup) station (O) of a department to the input (delivery) stations (I) of other departments. If a Euclidean distance metric is used, then the materials are assumed to flow along a straight line connecting the input and output stations of departments. If a rectilinear metric is used, then the materials are assumed to flow along two perpendicular line segments connecting the input and output stations of the departments. Some authors argue that this metric is more practical then the Euclidean distance metric, since it more closely estimates the real distance that materials flow between departments. When the actual path distance metric is used, it is assumed that the materials flow along the perimeters of departments. The rectilinear distance metric is the

easiest of the three distance metrics to model mathematically. However, in practice the flow of materials between any two departments usually does not occur through other departments, which are between them; therefore, the actual path distance metric is obviously the most practical, though hardest to model mathematically. In Figure 1.1, the flow of materials from the output station of department 5 (D5) to the input station of department 6 (D6) uses the Euclidean distance metric. The rectilinear distance metric is used to obtain the distance from department 1 (D1) to department 5 (D5), and the actual path distance metric is used to obtain the distance from department 6 (D6) to department 2 (D2). In this example, we used three different types of distance metrics for illustrative purposes only, and usually only one of the three distance metrics is used.



1.2.2 Discrete versus Continuous Approach

Earlier it was stated that the FLP with equal area departments can be modeled as a QAP, where the plant floor is divided into grids of equal size locations. Recall, the QAP

assigns departments to locations. The layout or solution of the QAP is often given as a block layout which is a graphical representation of the plant floor illustrating the relative locations of the departments. In this case, the block layout uses the discrete representation to specify the solution or layout. When considering unequal area departments and the discrete representation, the layout is divided into equal size grids, and the FLP is to assign departments to sets of grids (or locations) on the plant floor (see **Armour and Buffa, 1963**, **Bazaraa, 1975**, as well as **Bozer** *et al.*, **1994**). In other words, departments are divided into sub-departments that are assigned to grids (or locations) on the plant floor such that some objective(s) is (are) optimized.

The deficiencies of the majority of the methods presented in the literature considering the discrete representation of the FLP with unequal area departments are that they produce solutions with irregular shape departments (**Bozer** *et. al.*, **1994**), and the computational time may increase considerably when using smaller grid sizes. However, with larger grid sizes, the areas of departments in the layout may differ significantly from the specified areas. In Figure 1.2, examples of two block layouts are shown when the discrete representation of the FLP with unequal area departments is used. The departments, to which the sub-departments belong are shown in parentheses (e.g., department 1 is divided into 6 equal size sub-departments denoted 1 through 6). Notice department 2 in the layout given in Figure 1.2(a) has a rectangular shape. However, department 2 in Figure 1.2(b) has a nonrectangular or irregular shape (L-shaped department).

1(1)	2(1)	3(1)	7(2)	2(1)	3(1)	10(3)	11(3)
4(1)	5(1)	6(1)	8(2)	1(1)	4(1)	12(3)	13(3)
10(3)	11(3)	14(4)	9(2)	5(1)	7(2)		14(4)
12(3)	13(3)	15(4)		6(1)	8(2)	9(2)	15(4)
(a)				(b)			

Figure 1.2: Graphical representation of two solutions using the discrete representation of the FLP with unequal-area departments: (a) with regular shape departments; and (b) with irregular shape departments

Bazaraa (1975) presented a discrete formulation of the FLP with unequal area departments as a quadratic set covering problem such that the objective function is of quadratic type and the constraints are of set partitioning and set covering types. In his model, the set of departments should be covered by the set of locations. Also, **Bazaraa** (1975) presented the generalized quadratic assignment problem formulation for the FLP with unequal area departments such that each department is assigned to a specified number of grids, and at most one department should be assigned to each location.

Some techniques which solve the FLP with unequal area departments consider the continuous representation of the block layout. Using this representation, departments may be placed anywhere on the continuous plane (see Montreuil and Ratliff, 1989, Tam and Li, 1991, as well as Imam and Mir, 1993). In this dissertation, the continuous representation of the block layout is used and is illustrated below.

1.2.3 Fixed/Variable Shape Departments

In the FLP with unequal area departments, the departments may have either fixed or variable shapes. The dimensions of a fixed shape department are defined by specifying the values for the length and width, or longer and shorter side lengths for the department. The dimensions of a variable shape department are usually defined by specifying the area of the department and the lower and upper bounds on allowed ratios of the department's length to its width, or the ratio of the department's longer side length to its shorter side length. This ratio is called the aspect ratio. Furthermore, the departments may be restricted to vertical or horizontal orientation, or may have any orientation. For example, consider the case when the dimensions of the departments are given by specifying the areas and lower and upper bounds on aspect ratios where the aspect ratio of a department is defined as the ratio of the department's longer side length to its shorter side length. The areas, minimum (min) and maximum (max) aspect ratios, and orientations of the departments are specified in Table 1.1.

Dent		Min	Max	
Dept	Area	Aspect	Aspect	Orientation
(U)		Ratio	Ratio	
1	100	1.0	2.5	Any
2	125	1.3	1.8	Any
3	45	2.0	2.0	Any
4	60	1.7	2.1	Horizontal
5	110	2.0	3.0	Any
6	205	1.5	3.0	Any

Table 1.1: Areas, aspect ratios, and orientations of departments

In Tables 1.2(a) and 1.2(b), actual lengths and widths of departments are given such that the areas of the departments are as given in Table 1.1, and the aspect ratios are within the ranges defined in the table. In Figures 1.3(a) and 1.3(b) the continuous representations of two solutions are presented such that the departments have lengths and widths specified in Tables 1.2(a) and 1.2(b), respectively. Notice the orientations of the departments correspond to the orientations given in Table 1.1. Recall, the continuous representations of the two solutions given in Figure 1.3 are called block layouts.

Dept (i)	Length	Width	Aspect Ratio		Dept (i)	Length	Width	Aspect Ratio
1	14.541	6.877	2.114		1	10.955	9.129	1.200
2	9.281	13.469	1.451		2	8.575	14.577	1.700
3	4.744	9.487	2.000		3	9.487	4.744	2.000
4	10.628	5.646	1.882]	4	11.168	5.373	2.079
5	17.736	6.202	2.860]	5	14.832	7.416	2.000
6	21.172	9.683	2.186		6	17.536	11.691	1.500
(a)				-	(b)			

Table 1.2: Two sets of lengths and widths corresponding to data in Table 1.1



Figure 1.3: Two solutions to the unequal area FLP corresponding to the data given in Tables 1.1 and 1.2

The main purpose of using lower and upper bounds on the aspect ratio of a department is to ensure that the lengths of longer and shorter sides of a department are greater than or are equal to specific values, and the area of a department is equal to the area specified. For example, the shapes department 6 in Table 1.1 can have, according to vertical and horizontal orientations, are shown in Figures 1.4(a) and 1.4(b), respectively. As it can be seen in Figures 1.4(a) and 1.4(b), the dimensions of department 6 could have been alternatively defined by restricting department 6 to have an area of 205 and by specifying the minimum lengths of the departments shorter and longer sides to be 8.27

and 17.54, respectively. The minimum values of shorter and longer sides may be derived from the bounds on aspect ratios as follows:

Shorter side minimum length $= \frac{area}{\sqrt{area * maximum aspect ratio}}$

Longer side minimum length = $\sqrt{\text{area} * \text{minimum aspect ratio}}$



Figure 1.4: Shapes of department 6 corresponding to extreme values of aspect ratios: (a) for vertical orientation; (b) for horizontal orientation

1.3 The Dynamic Facility Layout Problem

The FLPs discussed in Sections 1.1 and 1.2 consider the static case in which the amounts of materials that flow between pairs of departments are fixed during the planning horizon. This type of FLP is called the static FLP. In contrast, in the dynamic environment, the material flows between pairs of departments change during the planning horizon. Also, the sizes of departments may change to accommodate these changes. In **McKendall and Shang (2006)**, the authors list some of the causes for changes in material flow as follows.

• Termination of production of some products

- Introduction of new products
- Change in demands of products
- Change in designs of products

In addition, companies which produce products with shorter life cycles (e.g., computer manufacturing companies), are more likely to have higher frequency of changes in material flows. Also, **Nicol and Hollier (1983)** point out that if the effective lifetime of a layout is defined as the elapsed time from installation until at least one-third of all key manufacturing operations are replaced, then it was found that nearly half of the companies surveyed had an average layout stability of two years or less.

If the material flows change during the planning horizon, the planning horizon can be divided into time periods (e.g., months, years, etc.), during which the material flows between departments do not change. Data for material flows between pairs of departments for each period can be forecasted. In this dissertation, the dynamic FLP (DFLP) is considered, which is the problem of finding positions of departments in each period, such that departments do not overlap, and the sum of material handling costs and costs of rearranging departments between consecutive periods is minimized.

The department rearrangement costs may be divided into two categories: fixed and variable costs (**Balakrishnan and Cheng**, **1998**). Fixed costs are defined as costs which do not depend on how much departments have been rearranged. In contrast, variable costs are based on the distance the departments are transported and the increase or decrease in department sizes. In most of the papers available in the DFLP literature, they use fixed rearrangement costs. Nevertheless, examples of rearrangement costs are as follows.

- 1) Setup costs associated with preparing the department(s) for rearrangement
- 2) Cost of leasing the equipment for rearranging departments
- Costs associated with the loss of production during rearrangement of departments
- 4) Distance based costs associated with transporting centers of departments (D'Souza and Mohanty, 1986, as well as Montreuil and Laforge, 1992)
- 5) Distance based costs associated with a unit-distance displacement of the west, east, south, and north sides of departments (Montreuil and Laforge, 1992)
- Labor costs associated with hourly wages paid to personnel responsible for rearranging departments

A DFLP instance with 2 periods and 6 departments is shown below. Department data for periods 1 and 2 are defined in Table 1.3. The material handling costs per distance unit and department rearrangement costs are given in Table 1.4 and Table 1.5, respectively. Two solutions for the DFLP instance were obtained and are shown in Figures 1.5 and 1.6. The block layouts shown in Figure 1.5 were obtained by solving the DFLP, when the objective is to minimize the sum of material handling and rearrangement costs. In contrast, the layouts shown in Figure 1.6 were obtained by solving two static FLPs separately for each of the two periods, without considering rearrangement costs as shown in Table 1.5. In Figures 1.5 and 1.6, the input and output stations of departments are denoted by triangles pointing down and up, respectively. Notice that only three departments are rearranged (i.e., departments 1, 4, and 6) in solution 1 (Figure 1.5). Both shapes and positions of centroids of departments 1 and 4 changed in period 2, whereas only the shape of department 6 changed. However, all departments except department 3

are rearranged in period 2 in solution 2 (Figure 1.6), since rearrangement costs are not considered. In Table 1.6, the evaluations of solutions are given with respect to material handling and rearrangement costs. As it can be seen, the material handling cost for solution 2 (layouts in Figure 1.6) is lower, but the rearrangement cost is much higher than for solution 1 (layouts in Figure 1.5), since rearrangement cost was ignored when solving the two static FLPs for each period. Thus, the sum of material handling and rearrangement costs corresponding to solution 1 (i.e., total cost = 1035.44) is lower than the total cost corresponding to solution 2 (i.e., total cost = 1482.27) by over 43%. The reason for such a large difference in total cost is due to relatively high rearrangement cost (see Table 1.5). Hence, the DFLP should not be solved by solving a series of static FLPs for each period such that rearrangement costs are ignored. In this dissertation, the DFLP is solved such that the sum of the material handling and rearrangement costs is minimized.

Period (t)	Dept. (i)	Area	Min Aspect Ratio	Max Aspect Ratio	Orientation
	1	40.00	2.50	2.50	Any
	2	60.00	1.50	3.00	Any
1	3	300.00	2.00	3.00	Any
	4	50.00	1.50	3.00	Vertical
	5	50.00	1.00	3.00	Any
	6	250.00	2.50	2.50	Horizontal
	1	50.00	2.00	2.00	Any
	2	60.00	1.50	3.00	Any
2	3	300.00	2.00	3.00	Horizontal
	4	50.00	1.50	3.00	Vertical
	5	55.00	1.50	3.00	Any
	6	200.00	2.00	2.00	Any

Table 1.3: Department dimensions for the DFLP instance

The second secon			
Pe	170	r l	

Period 2

i/j	1	2	3	4	5	6	i∕j	1	2	3	4	5
1	0	10	3	0	0	0	1	0	0	8	8	0
2	13	0	14	10	0	0	2	0	0	14	10	13
3	2	3	0	10	0	0	3	0	8	0	10	2
4	0	10	4	0	7	10	4	9	10	4	0	0
5	0	0	8	8	0	0	5	0	10	8	0	0
6	2	0	0	4	10	0	6	10	0	0	14	12

Table 1.4: Costs to transport materials per distance unit between departments in periods 1 and 2

Dept. (i)	Fixed Cost to Rearrange the Department	Cost to Transport the Department Per Distance Unit		
1	15	5		
2	10	7		
3	60	20		
4	10	7		
5	12	6		
6	40	20		

Table 1.5: Rearrangement costs for the DFLP instance for period 2



Figure 1.5: Block layout of the DFLP instance obtained by considering both material handling and rearrangement costs



Figure 1.6: Block layout of the DFLP instance obtained by considering only material handling costs

	Material Handling Cost	Rearrangement Cost	Cumulative Cost
Solution 1	954.85	80.59	1035.44
Solution 2	892.63	589.64	1482.27

Table 1.6: Costs of solutions in Figures 1.5 and 1.6

CHAPTER 2

PROBLEM STATEMENT

2.1 Statement of the Problem

In this dissertation, the DFLP with unequal area departments is considered. The problem is to find the layouts of N rectangular shape departments on the plant floor for each period in the planning horizon such that the sum of the material handling and rearrangement costs is minimized. Most, if not all of the models in the literature consider this objective.

2.2 Problem Assumptions

In this dissertation, the following assumptions are made for the DFLP.

- Plant floor and departments have rectangular shapes. The shape of each department is defined by specifying department orientation (i.e., vertical, horizontal, or any), and lengths of shorter and longer sides of the department. More specifically, departments have fixed shapes.
- 2. The continuous representation of the DFLP is considered.
- 3. In each period, the departments should be laid out on the plant floor such that they are within the boundaries and no two departments overlap.
- 4. Each department in each period has one input (I) station to which materials flow in from other departments and one output (O) station from which the materials flow out to other departments. Without lost of generality, I/O stations are at

centroids of departments. The amount of flow from output stations to input stations of all departments are known for each period.

- 5. Departments are rearranged at the beginning of a period if necessary, and rearrangement costs are considered.
- 6. The rectilinear distance metric is used to measure the distances between input and output (I/O) stations of departments.

In assumption 5, a department is rearranged in consecutive periods, if either its shape, centroid or locations of I/O stations change. This assumption was made in **Dunker** *et al.* (2005).

2.3 Research Objectives

The main objectives of this research are:

- 1. To develop construction algorithms, based on a boundary search technique and the dual simplex method with an LP formulation, for the DFLP.
- 2. To develop a tabu search (TS) and a memetic heuristic, improvement heuristics, based on the boundary search and dual simplex method for the DFLP.
- 3. To evaluate the effectiveness of the proposed heuristics by comparing their results to the results obtained from the "best" techniques in the literature for the DFLP.
- 4. To generate a new set of test problems to thoroughly evaluate the effectiveness of the proposed heuristics.

CHAPTER 3

LITERATURE REVIEW

The QAP is shown to be NP-hard (see **Sahni and Gonzalez, 1976**), and there exists no algorithm which solves the problem in polynomial time. **Frances and White (1974)** noted, that except for relatively small-sized problems, an exact solution to the QAP cannot be obtained at a reasonable computational cost. Therefore, heuristic solution procedures are generally used to obtain "good" solutions.

Exact algorithms, which guarantee optimal solution, are able to solve unequal area static FLPs with up to 13 departments in reasonable time (see Meller *et al.*, 1999, Castillo and Westerlund, 2005, as well as Castillo *et al.*, 2005). A number of suboptimal heuristics that solve the FLP, have been developed in the last several decades. These heuristics do not guarantee optimal solutions to problems, but usually find good solutions in reasonable time. Most of the heuristics in the literature are either construction or improvement type heuristics. In construction type heuristics, a layout or solution is generated from scratch. Improvement type heuristics require an initial solution(s) as input, generated by some construction type heuristic. The initial solution is improved iteratively. Wilhelm and Ward (1987) state that construction type heuristics have been found to yield superior solutions. In the sections that follow, the review of the FLP literature is presented. Review of the literature can also be found in Singh and Sharma (2006), Liggett (2000), Meller and Gau (1996), Welgama and Gibson (1995), and

Kusiak and Heragu (1987). For a review of the DFLP, specifically, see Balakrishnan and Cheng (1998).

3.1 Discrete Representation of the FLP

3.1.1 Static Facility Layout Problem

Koopmans and Beckmann (1957) were the first to model the FLP with equal size departments, as a QAP. They consider profit associated with each department to location assignment. Also, there is a flow of commodity between pairs of departments. The mathematical formulation maximizes the profit from assignment of departments to locations minus the material handling costs between facilities, subject to the condition that each department is assigned to exactly one location, and exactly one department is assigned to each location. The commonly used QAP formulation, modified from **Koopmans and Beckmann (1957)** is the following:

Minimize
$$\sum_{i} \sum_{k} A_{ik} x_{ik} + \sum_{i} \sum_{k} \sum_{j} \sum_{l} C_{ikjl} x_{ik} x_{jl}$$
 (3.1a)

s.t.
$$\sum_{k} x_{ik} = 1$$
 for $i = 1, ..., N$ (3.1b)

$$\sum_{i} x_{ik} = 1 \text{ for } k = 1, \dots, N$$
(3.1c)

$$x_{ik} = 0 \text{ or } 1 \text{ for } i, k = 1, ..., N$$
 (3.1d)

Where the parameters used in this model are:

- *N* is the number of departments.
- f_{ij} is the flow of materials from department *i* to department *j* where i, j = 1, ..., N.
- d_{kl} = distance from location k to location l where k, l = 1, ..., N.

- c_{ikjl} = cost per unit flow of materials from department *i* to department *j* per distance unit from location *k* to location *l*.
- C_{ikjl} is material handling cost from department *i* located at location *k* to *j* located at location *l* such that $C_{ikjl} = c_{ikjl} f_{ij} d_{kl}$.
- A_{ik} is the cost of assigning department *i* to location *k* where *i*, *k* = 1,..., *N*.
 The decision variables are:
- *x_{ik}* is a binary variable which is 1 if department *i* is assigned to location *k* and zero otherwise.

Gilmore (1962) and **Lawler (1963)** were the first to develop optimal procedures for the QAP problem defined by **Koopmans and Beckmann (1957)** based on branch and bound techniques. **Gilmore (1962)** also developed two heuristics which are modified versions of the branch and bound algorithm. The heuristics consider only certain promising branches. However the solution is not guaranteed to be optimal. Although the techniques by **Gilmore (1962)** and **Lawler (1963)** are computationally more effective than complete enumeration of all possible assignments, according to **Gilmore (1962)**, his algorithm is probably not computationally feasible for *N* much larger than 15.

Hillier (1963) developed a pairwise exchange heuristic for solving the QAP. At each iteration, the heuristic considers the pairs of neighbor work centers (departments), as candidates for interchanging their locations. The heuristic chooses the pair, which will result in greatest positive improvement in the objective function value. The heuristic stops, when there is no exchange, which results in improvement in objective function value. Also **Hillier (1963)** describes a procedure which considers non neighbor work centers for exchanges. In this article, Hillier points out how the heuristic can be modified

to solve FLP with unequal area work centers. One of the proposed methods is to divide the work centers into several work centers (sub-work centers) where the dimensions of each is equal to the dimensions of smallest work center, and to divide the flow of the work center to and from other work centers between those sub-work centers. To avoid splitting of the divided work-centers, he suggests assigning large artificial flows between sub-work centers of the same work centers.

Armour and Buffa (1963) developed a pairwise exchange improvement type heuristic for solving the FLP with unequal area departments. At each iteration the algorithm exchanges the locations of two departments. Two departments are eligible for exchange if they are either adjacent in current layout, or have equal areas. Two departments are picked at current iteration for exchange, if the improvement in the objective function value associated with the exchange of centers of the departments is positive and is greatest among all the eligible pairs of departments. The heuristic moves to a new solution, by exchanging the locations of subdepartments of exchange. However, the algorithm may produce irregular (nonrectangular) shaped departments. Buffa *et al.* (1964) developed an improvement type heuristic called CRAFT (Computerized Relative Allocation of Facilities Technique), for solving the FLP with unequal area departments. The algorithm is similar to that described by Armour and Buffa (1963), except that it also considers exchanges of locations of three departments.

Bazaraa (1975) formulated the unequal area FLP using the discrete representation as a quadratic set covering problem and developed a branch and bound algorithm for solving the problem. The plant area is divided into blocks. The shapes and areas of

20

objects (departments) in terms of basic blocks are determined by the analyst. For each department the analyst must specify the sets of candidate locations to which the subdepartments of departments may be assigned. He also formulated the problem as a generalized quadratic assignment problem such that each department is assigned to a specified number of grids, and at most one department should be assigned to each location.

Burkard and Rendl (1984) as well as **Wilhelm and Ward (1987)** applied simulated annealing (SA) heuristic to solve the QAP. The simulated annealing heuristic was developed by **Kirkpatrick** *et al.* **(1983)** for solving combinatorial optimization problems, and it is based on natural phenomena of bringing the melt metal to its lowest energy state by slowly lowering the temperature of the metal. If the metal is cooled too quickly, imperfections can occur. The simulated annealing heuristic is a random pairwise exchange heuristic, which avoids getting trapped at the local optimum by considering non-improving exchanges.

Hassan *et al.* (1986) developed a construction heuristic, called SHAPE, for solving the discrete representation of the unequal area FLP. The areas of departments are given. The heuristic orders the departments in such a way, that the departments with higher interactions with all other departments are earlier in the list, and the pairs of departments with relatively high interactions between them are as close as possible in the list. The heuristic picks one department at a time from the list of departments and assigns to it neighboring squares on the layout in such a way, that the department have maximally regular (rectangular) shape. The first department is placed at the center of the plant floor, and the rest of the departments are placed around it. In this dissertation, this type of heuristic is referred to as a boundary search heuristic. The procedure for determining the department placement order is an improvement of methods by **Apple and Deisenroth** (1972) and **Lee and Moore (1967)**. The drawback of the heuristic is that some irregularity in shapes of departments is still possible. In addition, smaller size square grids increase computational time.

Li and Mashford (1990) applied a genetic search algorithm to solve the QAP. The genetic algorithm (GA) is an improvement type heuristic developed by Holland (1975), for solving combinatorial optimization problems, and it resembles the natural phenomena of the survival of the fittest (i.e., when most fit in the population survive and reproduce).

Skorin-Kapov (1990) were the first to apply a tabu search (TS) improvement type heuristic to the QAP. Tabu search heuristic was introduced by **Glover (1986)** and was improved by **Glover (1989)** and **Glover (1990)**. The heuristic by **Skorin-Kapov (1990)** is a pairwise exchange heuristic which uses memory (tabu list) to store the list of a number of recent exchanges. The number of iterations a move is declared tabu is called the tabu duration (tenure length). Therefore, non-improving moves (or solutions) are selected using the tabu list such that the heuristic may escape from "poor" local optimum.

Bozer *et al.* (1994) developed a pairwise exchange heuristic, called MULTIPLE, for solving the discrete representation of a single and multi-floor FLP with unequal area departments. The heuristic uses the space filling curve (SFC) to construct the layouts, in which the departments are not split. The SFC is a continuous line passing through all the grids of each floor. The solution is represented as a sequence of department numbers. The layout corresponding to specific sequence of department numbers is uniquely constructed by consecutively assigning the required number of grids across SFC to departments. The
use of SFCs allows the heuristic not to restrict the exchanges to equal size or adjacent departments, as CRAFT by **Armour and Buffa (1963)** does. Figure 3.1 is an example from **Bozer** *et al.* **(1994)** of two different layouts corresponding to solutions (1, 2, 3, 4, 5, 6) and (1, 5, 3, 4, 2, 6) respectively. The layout in Figure 3.1(b) is obtained from layout in Figure 3.1(a) by exchanging departments 2 and 5. To force the shapes of departments to be maximally regular, they use a shape measure. The exchange of departments is rejected, if it results in a layout with department(s) which violate the shape measure.



Figure 3.1: The layouts corresponding to sequences of departments (1, 2, 3, 4, 5, 6) and (1, 5, 3, 4, 2, 6)

Chiang and Chiang (1998) applied simulated annealing, tabu search, probabilistic tabu search (PTS) and a hybrid of tabu search and simulated annealing heuristics to solve the QAP. Also, **Drezner (2008)** used a memetic algorithm (MA) to solve the QAP. The MA was developed by **Norman and Moscato (1989)** and is a hybrid of GA and TS.

3.1.2 Dynamic Facility Layout Problem

Rosenblatt (1986) was the first to introduce the DFLP with equal area departments, which minimizes the sum of material handling and rearrangement costs. **Rosenblatt** (1986) considers rearrangement costs C_{km} , associated with rearranging from some layout A_k to some other layout A_m . The author developed dynamic programming method for solving the problem.

Lacksonen and Enscore (1993) modified and analyzed five methods for solving the DFLP with equal area departments. Four of them originally were heuristics for solving the static FLP with equal area departments modified by the authors to solve the DFLP with equal areas. The original algorithms are: pairwise exchange routine; cutting planes algorithm for solving the QAP by **Burkard and Bonniger (1983)**; branch and bound algorithm by **Pardalos and Crouse (1989)**, modified to store only the 25 most promising nodes and to stop the algorithm after 50000 nodes are analyzed; cut tree algorithm by **Gomory and Hu (1961)**. The fifth heuristic is **Rosenblatt's (1986)** dynamic programming method for solving the DFLP with equal areas. Authors mention that the modified cutting planes algorithm by **Burkard and Bonniger (1983)** in a series of tests outperformed the four other heuristics.

Conway and Venkataramanan (1994) used a GA to solve the DFLP with equal area departments. On the other hand, Kaku and Mazzola (1997) developed a TS heuristic for the DFLP.

Corry and Kozan (2004) developed an ant colony optimization (ACO) algorithm to solve the DFLP with unequal area fixed shape departments. ACO heuristics first were presented by **Dorigo** *et al.* (1996) for solving the traveling salesman problem. These

types of heuristics simulate how the ants search for food by leaving a chemical trail called pheromone trail. The amount of pheromone trail left by an ant depends on the amount of food found. If the food source is far from the ant colony nest, fewer ants will be able to follow the trail in a given amount of time, and the pheromone will eventually evaporate. **Corry and Kozan (2004)** use a graph representation that has a node for each department and each grid on the plant floor. Unlike most of the techniques which use discrete representation of the problem, the heuristic constructs rectangular shape departments. However, as the authors mention, the level of grid resolution dictates the size of the graph, which should be kept as small as possible. In addition, according to the authors, it is desirable to keep grid resolution low for faster computation times. This makes the technique impractical for solving problems with large degree of variability in department sizes.

McKendall and Shang (2006) developed hybrid ant systems (HASs) for solving the DFLP with equal area departments. The first technique (HAS I) is a modification of the hybrid ant system (HAS) by Gambardella *et al.* (1999) for solving the QAP. The second technique (HAS II) is a modification of HAS I, which uses SA instead of a pairwise exchange heuristic. Finally, the third technique (HAS III) is a modification of HAS I with a look-ahead/look-back strategy added to the pairwise exchange heuristic. Similarly, McKendall *et al.* (2006) developed two simulated annealing heuristics for the DFLP. The first heuristic is a direct adaptation of SA to solve the DFLP. The second SA heuristic is a modification of the first SA, with added look-ahead/look-back strategy.

3.2. Continuous Representation of the FLP

3.2.1 Static Facility Layout Problem

Heragu (1989) developed a mixed integer linear programming (MILP) formulation for the continuous representation of the FLP with unequal area fixed shape departments. The author used two binary variables per pair of departments in the department nonoverlap constraints.

Montreuil (1990) presented MILP formulations for the static FLP with unequal area variable shape departments. Four binary variables are used for each pair of departments, which represent relative positions of the departments. Also the author linearized the nonlinear area constraints of the form $a_i = w_i h_i$, and $h_i / w_i \le \alpha_i$, where a_i is the given area of department *i*, w_i and h_i are the length and width of the department, and α_i is the upper bound on the aspect ratio of the department. According to Meller *et al.* (1999), the departments in the solutions to the MILP by Montreuil (1990) tend to have smaller areas than required. Montreuil *et al.* (1993) used design skeletons from Montreuil and Ratliff (1989) to preset the values of binary variables representing the relative positions of departments in the MILP formulation in Montreuil (1990).

Tam (1992-a) developed a GA to solve the FLP with unequal area variable shape departments. The binary slicing tree structure, such as the one shown in Figure 3.2, is used to represent the layout. The leaves of the slicing tree are department numbers, and the internal nodes are the branching operators. Four types of branching operators (left cut, right cut, upper cut, and bottom cut) define the relative positions of the departments under the internal nodes. The slicing tree is generated once at the beginning of the heuristic, and stays unchanged throughout the execution of the heuristic. New solutions are generated

by changing only the operators in internal nodes. A numeric value is assigned to each node, which is equal to the cumulative area of the departments, the corresponding leaves of which branch from the node. These values are used to determine the point where the block should be cut by horizontal or vertical line. The algorithm also considers the dead spaces, which are areas that cannot be occupied by departments.

		Aspect ratio								
Facility id	Area	lower bound	upper bound							
1	100	0.5	1.2							
2	80	0.4	1.1							
3	50	0.3	3							
4	60	0.1	5.5							
5	120	0.9	1.9							
6	40	0.4	1.2							

 Table 3.1: Geometric constraints for departments from Tam (1992-a)



Figure 3.2: Slicing tree from Tam (1992-a)



Figure 3.3: Layout constructed from the slicing tree in Figure 3.2

The objective function contains terms for penalizing the solutions, which have departments violating the aspect ratio constraints, or have departments which intersect dead spaces. Department data for a FLP problem instance is shown in table 3.1. In Figures 3.2 and 3.3 the slicing tree and the layout constructed from the slicing tree are shown. This example is taken from **Tam (1992-a)**. The first cut is a vertical cut which partitions the plant floor into two rectangles with dimensions 12.78 by 18 and 12.22 by 18. Since the values, assigned to nodes, branching from the root node are 230 and 220, the cut is done in such a way, that left and right rectangles resulted from the cut have areas 230 and 220, respectively (i.e., cumulative areas of departments below the left and right nodes, branching from the root node). The process of constructing the layout from the slicing tree is continued in this way, until space is allocated to all departments.

Tam (1992-b) developed a SA heuristic to solve the FLP with unequal area variable shape departments. The considered model and the solution representation are similar to that in Tam (1992-a). Garces-Perez *et al.* (1996) developed a GA which utilizes slicing tree structure to construct the layouts. They ensure that the aspect ratio constraints are

satisfied for all departments by expanding the blocks, after the layout is constructed. The algorithm also has operators for obtaining new tree structures. Among other authors that use slicing tree structures to solve the unequal area FLP are Schnecke and Vornberger (1997), Tam and Chan (1998), Shayan and Al-Hakim (1999), Al-Hakim (2000), Valenzuela and Wang (2001), and Shayan and Chittilappilly (2004).

Tate and Smith (1995) developed a GA for solving the FLP with unequal area variable shape departments. The authors use the flexbay structure, developed by Tong (1991), to construct the layout. The plant floor area is divided in one direction into bays of varying widths. Next, rectangular areas are allocated to departments within bays. The solution is represented as two chromosomes. The first chromosome is a permutation of department numbers, which represents the order in which areas are allocated to departments within bays. The second chromosome contains information about the number of bays, and information on where in the sequence in chromosome 1 the breaks between bays occur. The heuristic uses mutation operators to merge and split adjacent bays. The objective function includes a penalty term for penalizing the layouts with departments, violating the aspect ratio constraints. In Figure 3.4, a layout taken from **Tate and Smith (1995)** is shown, which corresponds to the permutation of departments (12, 4, 9, 20, 11, 13, 2, 18, 16, 19, 13, 8, 14, 6, 1, 5, 17, 7, 10, 15) and bay break points (4, 7, 9, 14, 16) (i.e., chromosomes 1 and 2, respectively). To construct this layout, the heuristic allocates required areas to departments 12, 4, 9, and 20 in bay 1. Next, the required areas are allocated to departments 11, 13, and 2 in bay 2, and the heuristic proceeds in this manner, until areas are allocated to all the departments.

12	11		19		17		
	11	18	13		7		
4				1	<i>'</i>		
	3		8				
9					10		
		16	14		10		
20	2			5			
			6		15		

Figure 3.4: The layout corresponding to the permutation (12, 4, 9, 20, 11, 13, 2, 18, 16, 19, 13, 8, 14, 6, 1, 5, 17, 7, 10, 15) and bay break points (4, 7, 9, 14, 16)

Imam and Mir (1998) developed a construction type algorithm for solving the FLP with unequal area fixed shape departments. The order of placing the departments is the same as in Welgama and Gibson (1993), except that the first department to be placed is the department which has the greatest flow with all other departments. Each time a department is placed, the linked list of boundary segments is created, and the department is moved along the segments in a stepwise manners, as demonstrated in Figure 3.5 taken from Imam and Mir (1998). At each step, the material handling cost of the department with already placed departments is calculated, and it is checked if the department overlaps with other departments. The best position found along the boundary of the placed departments is chosen for placing the department. The deficiency of the technique is that the execution time of the heuristic increases when using smaller step size. In contrast, large step size may result in a poor solution.



Figure 3.5: Path of movement of a department along the cluster boundary

Meller et al. (1999) modified the mathematical formulation by Montreuil (1990) for the FLP with unequal area variable shape departments. To decrease the number of nodes considered by branch and bound algorithm, the authors introduced a number of valid constraints (cutting planes), which are satisfied in any valid solution. These constraints are used to obtain better lower bounds, when solving the relaxed problem, obtained by a allowing some of the department separation binary variables to be continuous. Also the authors reduce the problem symmetry by forcing the center of some department with high interaction with other departments to be in one of the four quarters of the plant floor. This reduces the solution space without affecting the value of the objective function the optimal solution. In addition, Meller et al. (1999) used more accurate area constraints, than the constraints used in Montreuil (1990). The largest problem solved to optimality is a 7 department problem. Sherali et al. (2003) improved the model by Meller et al. (1999) by further improving the linearized area constraints, decreasing the problem symmetry and modifying the valid constraints. Sherali et al. (2003) linearized the area constraints by using a number of tangential supports per

department (cutting planes). The accuracy of the linearized area constraints in the solution depends on the number of tangential supports used. **Castillo and Westerlund** (2005) used a technique for linearizing area constraints, similar to the one presented in **Sherali** *et al.* (2003). The main difference in their technique is that the technique ensures that the actual area of each department in the solution is within ε % error of the required area, for any $\varepsilon \in (0,1)$.

Gau and Meller (1999) developed an iterative approach to solve the FLP with unequal area variable shape departments. A GA solves the problem using a slicing tree structure by **Tam (1992-a, b)**. The relative locations of departments in the solution generated by the GA are used to set the subset of the binary variables (in the range of 50%-100%) in the MILP formulation of **Meller** *et al.* (1999). The solution obtained by solving the MILP is used to generate an initial population of solutions for the GA, and the iterative loop is closed. In contrast to **Tam (1992-a, b)**, which uses a fixed tree structure, the technique by **Gau and Meller (1999)** uses dummy departments, to allow changes to the structure of the slicing tree.

Kim and Kim (2000) presented a MILP formulation and developed a two phase heuristic for solving the FLP with unequal area fixed shape departments. Four different configurations are considered for each department, obtained by rotating the department three times clockwise 90° from its basic orientation. Figure 3.6 is taken from **Kim and Kim (2000)** and is an example of the four possible configurations of the department. In addition to binary variables used in non-overlap constraints, four binary variables per department are used, for different department configurations.



The first phase of the heuristic is a construction type heuristic, which places the departments on the plant floor, one at a time. To place the current department, a MILP is solved, in which the binary variables for configurations and relative positions of previously selected departments are fixed. Therefore, at each iteration only binary variables corresponding to the department, being placed, are unknown. The department ordering procedure is stochastic, and it favors the departments with higher flows with previously selected departments. Therefore, different solutions are obtained by running the algorithm several times. The best solution obtained in the first phase is an input to the second phase. The improvement heuristic of the second phase considers four improvement types: exchange of positions of two departments, department configuration and position exchange, department configuration adjustment, and sub-area optimization. When performing each of the improvement types, most of the binary variables are fixed, and the MILP is solved with only a small number of binary variables unknown. This allows solving relatively large problems. The heuristic stops when the solution cannot be improved any farther.

Dunker *et al.* (2003) used a GA to solve the FLP with unequal area fixed shape departments. The shapes of the departments are fixed and the heuristic considers pick-up/drop-off points. They decompose the problem, by forming groups of departments with

relatively high flows between them. The layout for each group of departments is obtained by a GA. After layouts for groups are obtained, rectangles are drawn around departments in each group, and the arrangement of these rectangles is found. The chromosomes store information on the relative locations of departments in each group, which is used to fix corresponding binary variables in the MILP formulation and to solve the resulting relaxed MILP problems.

3.2.2 Dynamic Facility Layout Problem

Montreuil and Venkatadri (1991) developed a linear programming (LP) formulation for the DFLP with unequal area variable shape departments. In their model, positions of departments in the final layout are known. Also the areas of departments increase in consecutive periods (departments grow), and the boundaries of each department in each period should be within the boundaries of the same department in the next period. The mathematical formulation does not require binary variables, since the relative positions of pairs of departments are known. Therefore, large problems can be solved to optimality. Montreuil and Laforge (1992) improved the model by Montreuil and Venkatadri (1991) by relaxing the assumptions that the department areas increase in consecutive periods and that the boundaries of each department in each period should be within the next period. Similar to Montreuil and Venkatadri (1991), the mathematical formulation is linear, since the relative positions of departments in each probable future are specified by the designer and fixed rearrangement costs are not considered.

Lacksonen (1994) developed a two stage heuristic for solving the DFLP with unequal area variable shape departments. In Stage 1, all departments are assumed to have equal sizes, and the DFLP with equal area departments is solved by the cutting plane heuristic presented in Lacksonen and Enscore (1993). The rearrangement costs are determined at the end of Stage 1. In Stage 2, for every time period, a static unequal area FLP is solved as a modification of the MILP by Montreuil (1990). Stage 2 includes constraints, which ensure that the departments and time periods, which are not rearranged in Stage 1, are not rearranged in Stage 2 as well. In addition, the information about relative positions of departments in Stage 1 solution is used to preset some of the binary variables, used in department separation (non-overlap) constraints. Also, Lacksonen (1994) used piecewise linearization of area constraints. The linearization constraints ensure that the areas of departments are within 0% and +3% of required areas, for maximum aspect ratio of 2. Lacksonen (1997) developed a heuristic which fixes 80% of the binary variables in the model by Lacksonen (1994), and solved the MILP with the remaining 20% of binary variables using a revised branch and bound method.

Yang and Peters (1998) considered time windows when solving the DFLP with unequal area fixed shape departments. Each time window consists of a number of time periods, such that the material flows between departments are aggregated over these periods. The authors solve a series of static FLPs, one for each time window, using the MILP formulation. The structured hexagonal adjacency graph from **Goetschalckx (1992)** is used to fix the binary variables corresponding to relative positions of departments in each time window. **Dunker** *et al.* (2005) extended the GA presented in **Dunker** *et al.* (2003) to solve the DFLP with unequal area fixed shape departments. The authors store generation of solutions for each period. Each gene stores information about the relative positions of departments in a layout for a period. The solution (layout plan) corresponding to a gene is obtained by solving the relaxed MILP formulation for the static FLP in which the only unknown binary variables are variables representing the orientations of departments and configurations of I/O stations. Dynamic programming is used to evaluate the fitness of each gene γ in period t^* , which takes into account the rearrangement costs. The dynamic programming technique finds the best sequence of genes in periods preceding and succeeding period t^* . Thus $\prod_{t=1,t\neq t^*}^T N(t)$ layouts are evaluated, where N(t) is the number of genes in the population corresponding to period t^* .

CHAPTER 4

METHODOLOGY

4.1 Introduction

The DFLP is a combinatorial optimization problem, which is a generalization of the QAP, and the QAP is an NP-hard problem (**Sahni and Gonzalez, 1976**). There exists no exact technique, which solves the problem in polynomial time. In this chapter, a MILP formulation of the problem is presented, followed by construction and improvement heuristics. The MILP formulation can be used to solve only small instances of the problem, but the heuristics presented in this dissertation can be used to find good solutions in reasonable time for larger problem instances.

4.2 Exact Method

In this section, a MILP formulation is presented for the DFLP with unequal area departments. Similar formulations can be found in **Montreuil (1990)**, **Lacksonen (1994)**, and **Dunker** *et al.* (2005); however, the formulation presented in **Dunker** *et al.* (2005) is the closest to this formulation. The other authors considered variable shape departments, which is not considered here.

First, we give the notation used in the mathematical formulation. Note, the variables and indexes always start with small letters, and parameters start with capital letters. Indexes:

i, j = 1, ..., N: N is the number of departments;

t = 1,..., T: *T* is the number of periods;

Parameters:

 F_{tij} = Cost to transport materials a unit distance from department *i* to department *j* in period *t*;

 $F'_{tij} = F_{tij} + F_{tji}$ = total flow between departments *i* and *j* in period *t* (upper triangular matrix);

 R_{ti} = Rearrangement cost of shifting department *i* at the beginning of period *t*;

 Sh_{ti} = Shorter side length of department *i* in period *t*;

 Lng_{ti} = Longer side length of department *i* in period *t*;

 $DeptOrient_{ii} = \begin{cases} 0, \text{ if department } i \text{ in period } t \text{ can have any orientation;} \\ 1, \text{ if department } i \text{ in period } t \text{ is restricted to horizontal orientation;} \\ 2, \text{ if department } i \text{ in period } t \text{ is restricted to vertical orientation;} \end{cases}$

L = Length of the plant floor;

W = Width of the plant floor;

M = A large number;

Variables:

 (x_{ti}, y_{ti}) = The location of department *i* in period *t*;

 l_{ti} , w_{ti} = The length and width of department *i* in period *t*;

 x_p_{tij} , y_p_{tij} = Horizontal and vertical distances between the centers of departments *i* and *j* in period *t*;

 $h_{ti} = \begin{cases} 1, & \text{If department } i \text{ has horizontal orientation in period } t; \\ 0, & \text{Otherwise;} \end{cases}$

 $left_{tij} = \begin{cases} 1, \text{ If department } i \text{ is to the left of department } j \text{ in period } t \text{ (i.e., } x_{ti} + 0.5 * l_{ti} \le x_{ij} - 0.5 * l_{ij} \text{)}; \\ 0, \text{ Otherwise;} \end{cases}$ $below_{tij} = \begin{cases} 1, \text{ If department } i \text{ is below department } j \text{ in period } t \text{ (i.e., } y_{ii} + 0.5 * w_{ii} \le y_{ij} - 0.5 * w_{ij} \text{)}; \\ 0, \text{ Otherwise;} \end{cases}$ $r_{ti} = \begin{cases} 1, \text{ If department } i \text{ is rearranged at the beginning of period } t; \\ 0, \text{ Otherwise;} \end{cases}$

The MILP formulation for the DFLP with unequal area departments is as follows.

Minimize total cost =

$$\sum_{t=1}^{T} \sum_{i=1}^{N} \sum_{j>i}^{N} F'_{tij} \left(x_{p_{tij}} + y_{p_{tij}} \right) + \sum_{t=2}^{T} \sum_{i=1}^{N} R_{ti} r_{ti}$$
(4.1)

Subject to:

$$(x_{ti} + 0.5l_{ti}) - (x_{tij} - 0.5l_{tj}) \le M(1 - left_{tij}) \quad \forall t, i, j$$
(4.2)

$$(y_{ti} + 0.5w_{ti}) - (y_{ij} - 0.5w_{ij}) \le M(1 - below_{tij}) \quad \forall t, i, j$$

(4.3)

$$left_{ij} + left_{iji} + below_{ijj} + below_{iji} = 1 \quad \forall t, i, j$$

$$(4.4)$$

$$x_{ti} + 0.5l_{ti} \le L \quad \forall t, i \tag{4.5}$$

$$x_{ti} - 0.5l_{ti} \ge 0 \quad \forall t, i \tag{4.6}$$

$$y_{ti} + 0.5w_{ti} \le W \quad \forall t, i \tag{4.7}$$

$$y_{ti} - 0.5w_{ti} \ge 0 \quad \forall t, i \tag{4.8}$$

$$x_{p_{ij}} \ge x_{ii} - x_{ij} \quad \forall t, i, j > i$$

$$\tag{4.9}$$

$$x_{p_{ij}} \ge x_{ij} - x_{ii} \quad \forall t, i, j > i$$

$$(4.10)$$

$$y_p_{tij} \ge y_{ti} - y_{tj} \quad \forall t, i, j > i$$

$$(4.11)$$

$$y_{p_{ij}} \ge y_{ij} - y_{ii} \quad \forall t, i, j > i$$

$$(4.12)$$

$$l_{ti} = Lng_{ti}h_{ti} + Sh_{ti}(1-h_{ti}) \quad \forall t, i$$

$$(4.13)$$

$$w_{ti} = Lng_{ti}(1-h)_{ti} + Sh_{ti}h_{ti} \quad \forall t, i$$

$$(4.14)$$

$$h_{ti} = 1 \quad \forall t, i, DeptOrient_{ti} = 1 \tag{4.15}$$

$$h_{ti} = 0 \quad \forall t, i, DeptOrient_{ti} = 2 \tag{4.16}$$

$$x_{ti} - x_{t-1,i} \le Mr_{ti} \quad \forall i, t > 1$$
 (4.17)

$$-x_{ti} + x_{t-1,i} \le Mr_{ti} \quad \forall i, t > 1$$
(4.18)

$$y_{ti} - y_{t-1,i} \le Mr_{ti} \quad \forall i, t > 1$$
 (4.19)

$$-y_{ti} + y_{t-1,i} \le Mr_{ti} \quad \forall i, t > 1$$
(4.20)

$$w_{ti} - w_{t-1,i} \le Mr_{ti} \quad \forall i, t > 1 \tag{4.21}$$

$$-w_{ti} + w_{t-1,i} \le Mr_{ti} \quad \forall i, t > 1$$
(4.22)

$$x_{ti}, y_{ti}, l_{ti}, w_{ti}, x_p_{tij}, y_p_{tij} \ge 0 \ \forall t, i, j$$

$$(4.23)$$

 $h_{ti} = 0 \text{ or } 1, r_{ti} = 0 \text{ or } 1, left_{tij} = 0 \text{ or } 1, below_{tij} = 0 \text{ or } 1 \forall t, i, j$

The first term in the objective function (4.1) is used to obtain material handling costs, and the second term is for rearrangement costs. Constraints (4.2)-(4.8) are very similar to those presented in **Sherali** *et al.* (2003). These constraints ensure that the departments do not overlap and are within the boundaries of the plant floor. Constraints (4.9)-(4.12) are used to obtain the rectilinear distances between departments. Similar constraints are used by **Sherali** *et al.* (2003). Constraints (4.13)-(4.16) are used to control the orientations of the departments. Similar constraints are used in **Dunker** *et al.* (2005). Constraints (4.17)-(4.22) are slightly modified constraints from **Dunker** *et al.* (2005). These constraints ensure that the department has the same values of length, width, and center coordinates in any two consecutive periods in which the department is not rearranged. Last, the restrictions on the variables are given in constraints (4.23).

4.3 Construction Algorithms

As mentioned earlier, the mathematical formulation can solve only small instances of the DFLP. In this section, two construction heuristics are presented, which find solutions in reasonable computation time. The first heuristic, boundary search heuristic (BSH), constructs the layout by consecutively placing the departments along the boundary of already placed departments. The second heuristic uses an LP formulation and a dual simplex algorithm to construct the layout plans (i.e., solutions).

The heuristics developed in this dissertation obtain layouts, which fit within the plant floor boundaries. However, if the plant dimensions are too small, some of the layouts obtained by the heuristics may span outside the plant floor boundaries. Such solutions (layouts plans) are called *plant floor infeasible* solutions (layouts plans), and the solutions (layouts plans) in which all the departments in all periods fit within the plant floor boundaries are called *plant floor feasible* solutions (layout plans). By not discarding the *plant floor infeasible* solutions, the heuristics have a chance of exploring larger solution spaces and eventually may arrive at better solutions, which fit within plant floor boundaries. Nevertheless, the *plant floor feasible* solutions are always better than the *plant floor infeasible* solutions. If there are layout plans, which are either all *plant floor feasible* or all *plant floor infeasible*, then the layout plan with the lower OFV is better.

4.3.1 Boundary Search Heuristic (BSH)

4.3.1.1 Solution Representation

The BSH described in this dissertation constructs the layout by consecutively selecting some department i in some period t and placing the department at the most favorable position found along the boundary of already placed departments. The solution is represented as a vector of department period pairs as follows.

$$\pi = \{(i_1, t_1), (i_2, t_2), \dots, (i_{N*T}, t_{N*T})\}$$

If department pair (i_k, t_k) precedes (i_r, t_r) in this vector, then the BSH will place the department i_k in period t_k before it places the department i_r in period t_r , where $k, r \in \{1, 2, ..., NT\}$.

4.3.1.2 BSH Parameters

In addition to the notation defined above (including section 4.2), the following notation is used for the BSH heuristic.

$$f(\pi) = OFV$$
 of solution π ,

- $\pi_{per_{t}}$ = Ordered list of department numbers already placed in period *t* such that department *i* precedes *j*, if and only if (*i*, *t*) precedes (*j*, *t*) in π ,
- $|\pi_per_t|$ = Number of departments which have already been placed on the plant floor in period *t* (cardinality of the vector π *per_t*);

(*i_curr*, *t_curr*) = Department *i_curr* being placed in period *t_curr* by the BSH;

 $\pi_{partial}$ = Partial solution constructed by the BSH with respect to π_{pert} for all *t*;

 $TC_{i \ curr, t \ curr} =$ Cost of locating department i_curr in period t_curr ;

feas_st = 1 if the solution found by the heuristic is *plant floor feasible*, and 0 otherwise;

- $flow(x_cand, y_cand) =$ The flow cost between department i_curr and all placed departments in period t_curr (i.e., departments in $\pi_per_{t_curr}$) if the center of department *i* curr in period *t* curr is at candidate location (*x* cand, *y* cand);
- (cg_x, cg_y) = The most favorable location on the plant floor for department i_curr in period t curr;
- $hor_segms_d_t$, $hor_segms_u_t$ = Vectors, the elements of which are vectors themselves, storing the coordinates of horizontal boundary segments in period *t* facing downward and upward, respectively (details will be explained later);
- *vert_segms_l_t*, *vert_segms_r_t* = Vectors, the elements of which are vectors themselves, storing the coordinates of vertical boundary segments in period *t* facing leftward and rightward, respectively (details will be explained later);

4.3.1.3 Construct Layout Plan Using π

4.3.1.3.1 The Construction of π

The solution vector π is initialized in such a way, that the following two conditions are satisfied:

1) If
$$\sum_{t=1}^{T} \sum_{\substack{d=1\\d\neq i}}^{N} F'_{tid} \ge \sum_{t=1}^{T} \sum_{\substack{d=1\\d\neq j}}^{N} F'_{tjd}$$
, then (i, t_1) precedes (j, t_2) in solution π for any t_1 and t_2 .

2) If
$$\sum_{\substack{d=1\\d\neq i}}^{N} F'_{t_1 i d} \ge \sum_{\substack{d=1\\d\neq i}}^{N} F'_{t_2 i d}$$
, then (i, t_1) precedes (i, t_2) in solution π for any t_1 and t_2

In other words, the departments, which have higher cumulative flow with other departments over all periods, are placed in vector π first, in condition (1). In other word, in condition (1), each department *i* is placed in all periods in vector π and only then some

other department *j* is placed in all periods. In condition 2, the order of periods in which department *i* is placed in vector π is determined by the flow of department *i* with all other departments in each of these periods. For example, the solution π constructed for the problem instance given in Appendix A is shown in Table 4.1 below.

(<i>i</i> , <i>t</i>) Pair	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36
Department(i)	8	8	8	9	9	9	12	12	12	6	6	6	1	1	1	2	2	2	10	10	10	3	3	3	7	7	7	5	5	5	11	11	11	4	4	4
Period(<i>t</i>)	1	3	2	1	2	3	1	2	3	1	2	3	3	1	2	1	3	2	2	3	1	3	1	2	1	3	2	3	1	2	1	3	2	1	3	2

Table 4.1: The department period pairs in the vector π generated by the BSH for the problem instance in Appendix A

In this example, first department 8 will be placed in period 1, since the total flow between

department 8 and all other departments is the highest (i.e., $\sum_{t=1}^{3} \sum_{d=1}^{12} F'_{t8d} = 366$) and the $d \neq 8$

highest flow between department 8 and all other departments is in period 1 (i.e., $\sum_{d=1}^{12} F'_{1,8,d} = 147$). The next highest flow between department 8 and all other departments is $\int_{d=1}^{d} F'_{1,8,d} = 147$

in period 3 (i.e., $\sum_{\substack{d=1\\d\neq8}}^{12} F'_{3,8,d} = 132$). Hence, department 8 will be placed next in period 3,

and then department 8 in period 2. Department 9 is not placed, until department 8 is placed in all periods, since the total flow between department 9 and all other departments is the next highest (i.e., $\sum_{t=1}^{3} \sum_{d=1}^{12} F_{t9d}^{*} = 326$). After department 9 is placed in periods 1, 2,

and 3, then department 12 will be placed in each period, and so on.

4.3.1.3.2 Finding Candidate Locations for *i_curr* in period *t_curr*

Given a solution vector π , the BSH uses four vectors to determine the location of the current department being placed. These vectors are ordered special structures of four types of boundary segments: downward (hor segms d_t), upward (hor segms u_t), leftward (vert segms l_t), and rightward (vert segms r_t). The binary search algorithm is used to efficiently search for segments to find feasible regions in which the department may be placed. Each horizontal or vertical segment used in these vectors has the form segment = $\langle c, c_1, c_2 \rangle$. If the segment is horizontal then c is the Y coordinate of the segment endpoints, and c_1 and c_2 are the X coordinates of the segment endpoints. In a similar manner, if the segment is vertical then c is the X coordinate of the segment endpoints, and c_1 and c_2 are the Y coordinates of the segment endpoints. Every time some department i is placed in the layout in period t, the BSH ensures that the vectors hor segms d_t , hor segms u_t , vert segms l_t , and vert segms r_t are modified in such a way that two conditions are satisfied. These conditions are demonstrated below considering only vert segms r_t . The conditions for vectors hor segms d_t , hor segms u_t , *vert segms l_t* are exactly the same.

All the segments <*c*, *c*₁, *c*₂> in *vert_segms_r*_{ts} have the same value of *c* components, and are ordered in ascending order based on the value of the *c*₁ component, where *s* = 1,..., |*vert_segms_r*_t|. For example, if the partially constructed layout in period 2 is as defined in Figure 4.1, then *s* = 1, ..., 6 and

 $vert_segms_r_2 = \{vert_segms_r_{21}, vert_segms_r_{22}, ..., vert_segms_r_{26}\}.$

 $vert_segms_r_2 = \{[(i,h), (e,d)], [(c,b)], [(g,f)], [(v,u)], [(r,q), (n,m)], [(t,s), (p,o)]\}.$ $vert_segms_r_2 = \{[<2.5, 3.5, 6>, <2.5, 7, 9.5>], [<3, 9.5, 11.5>], ..., [<8.5, 1.5, 3.5>, <8.5, 6.5, 8.5>]\}$

• The segments $\langle c, c_1, c_2 \rangle$ in *vert_segms_r_{ts}* have smaller value of *c* component than segments in *vert_segms_r_{ts+1}*, for $s = 1, ..., |vert_segms_r_t| - 1$.

For example, the value of the *c* component of segments in *vert_segms_r*₂₁ is 2.5, and the value of the *c* component of segments in *vert_segms_r*₂₂ is 3. Note, the partial layout plan in Figure 4.1 was obtained using a different solution π and was constructed for illustrative purposes only. Also, when placing departments, the coordinates of departments can be negative.



At each iteration, the BSH selects a department-period pair (*i_curr*, *t_curr*) from π and places the department *i_curr* in period *t_curr*. Before placing the department, the

coordinates of the center of gravity (i.e., the most favorable location) for the department are calculated as follows.

$$cg_{x} = \sum_{\forall j \in \pi_{per_{t_{curr}}}} \left(F'_{t_{curr,i_{curr,j}}} x_{t_{curr,j}} \right) / \sum_{\forall j \in \pi_{per_{t_{curr}}}} F'_{t_{curr,i_{curr,j}}} dr_{j} dr_{j}$$

The center of gravity will be closer to the departments already placed in period t_curr , which have higher flow with department i_curr . To find the best position for placing the department i_curr in period t_curr , the BSH for each of the orientations of the department (i.e., horizontal and vertical) tries to find rectangular region(s) (i.e., feasible regions) along each boundary segment, within which the department being placed may be moved along the segment without overlapping with other departments. The binary search algorithm searches the vectors $hor_segms_d_t$, $hor_segms_u_t$, $vert_segms_l_t$, or $vert_segms_r_t$, as well as the vectors contained in these vectors to quickly identify the feasible regions. Within each identified feasible region, the heuristic considers candidate location (x_cand , y_cand) for placing the department, closest to the center of gravity (cg_x , cg_y). The value flow(x_cand , y_cand) is calculated using the formula below:

$$flow(x_cand, y_cand) = \sum_{\forall j \in \pi_per_{t_curr, i_curr, j}} F'_{t_curr, i_curr, j} \left(x_cand - x_{t_curr, j} \right) + \left| y_cand - y_{t_curr, j} \right| \right)$$

If all candidate locations result in *plant floor infeasible* layout plans, then the candidate location with the lowest value of $flow(x_cand, y_cand)$ is selected. Otherwise, the location with the lowest value of $flow(x_cand, y_cand)$ among the locations, resulting in

a *plant floor feasible* layout plan, is selected. In addition, the best position, (x cand, *y cand*), is compared to the location of the department *i curr* in period *t curr* - 1 (if the position is not occupied, and the department *i curr* has already been placed in period t curr - 1) and the position of the department i curr in period t curr + 1 (if the position is not occupied, and the department *i* curr has already been placed in period t curr + 1). When comparing any two locations from the resulting candidate locations, the preference is given to the location, which will result in a *plant floor feasible* layout plan. If both locations result in a *plant floor feasible* layout plan or both locations result in a *plant* floor infeasible layout plan, then the combined material flow cost between the department *i curr* and all placed departments in period *t curr*, and the rearrangement cost of department *i* curr in periods t curr (if t curr > 1) and t curr + 1 (if t curr < T) is used as a comparison criteria. If the department has not been placed in previous (next) period, then the rearrangement cost of department *i* curr in period t curr (t curr + 1) is zero. If the vector $\pi per_{t curr}$ is empty (i.e., no departments have been placed in period t curr), and the department *i* curr has not been placed neither in previous nor in next period, then the department is centered on the rectangle, enclosing all the placed departments in all periods.

In Figure 4.2, the example of finding the best location of department 11 in period 2 along the segment $(y, l) = \langle 5.5, 3, 10.5 \rangle$ (see Figure 4.1) is presented. Recall, the problem data for this example are given in Appendix A. The current layout corresponding to period 3 is shown in Figure 4.1, and no departments have been placed in period 1. As it was mentioned above, the partial layout plan in Figure 4.1 was obtained using a different solution π and was constructed for illustrative purposes only. In this example, the BSH

searches for feasible rectangular regions within which department 11 can move parallel to segment (y, l), without overlapping with other departments (i.e., dark gray regions). The BSH tries both horizontal and vertical orientations of the department (demonstrated in Figures 4.2(a) and in Figure 4.2(b), respectively) and searches for feasible rectangular regions within the rectangle with corner points A, B, C, and D. Within each feasible region, the BSH considers candidate locations for department 11 (e.g., rectangles outlined by dashed lines in Figure 4.2), which is closest to the center of gravity for department 11 (e.g., point (3.61, 6.14)). For each candidate location (*cand_x, cand_y*), the BSH calculates the value *flow(cand_x, cand_y*). The BSH evaluates all the candidate locations along all the boundary segments and compares the best location found with the location of department 11 in period 3. Note, the rearrangement cost in period 2 is 0, since the department has not been placed in period 1.



(b) when the department has vertical orientation

The candidate locations for placing department 11 in period 2 along segment (*y*, *l*) as well as their flow cost, rearrangement costs, and total cost are shown in Table 4.2. The first two candidate positions are illustrated in Figure 4.2(a), and candidate positions 3 and 4 are illustrated in Figure 4.2(b). See Figure 4.1 for the location of department 11 in period 3. From the total cost of the candidate locations as well as the total cost of the assignment as in period 3, it is best to locate department 11 at the same location of the department in period 3 (i.e., $TC_{11,2} = 183$).

	X	Y	flow(X,Y)	Rearr. Cost In Period 2	Rearr. Cost In Period 3	TC_{i_curr,t_curr}
Candidate Position 1	4	5	157	0	50	207
Candidate Position 2	4	8	158	0	50	208
Candidate Position 3	4.5	4.5	175	0	50	225
Candidate Position 4	4.5	8.5	180	0	50	230
Position in Period 3	4	4	183	0	0	183

Table 4.2: Evaluating the position of department 11 in period 3, and the positions identified along vertical segment (l, y) in Figure 4.2

4.3.1.4 BSH Heuristic

The BSH is a modification of the Cluster Boundary Algorithm (CBA) presented by **Imam and Mir (1998)** for solving the static FLP. The main differences between the CBA and the BSH is that the CBA does not use an efficient technique for searching for feasible rectangular regions for placing the departments and is used for solving only static FLPs. The CBA heuristic moves the department being placed along the linked list of boundary segments in a stepwise manner to determine the feasible positions for placing the department. At each step, the OFV should be evaluated. If the step size is too small, then the computational time will be extremely high. On the other hand, using a larger step size will decrease the solution space and may result in poor layouts.

The steps of the BSH are as follows:

Step 1: Initialize π as described in Section 4.3.1.3.1;

 $\operatorname{Set} f(\pi_{partial}) = 0$

Set k = 1 where k = position of i_curr in vector π ,

Step 2: Set $feas_st = 0$;

Initialize the values of *i_curr* and *t_curr* as the *k*-th department period pair in π , If $|\pi per_t| = 0$

Place the department *i_curr* at the center of the plant floor in period *t_curr* and initialize the values of l_{t_curr,i_curr} , w_{t_curr,i_curr} , x_{t_curr,i_curr} , and y_{t_curr,i_curr} from the position of the department;

Else

- Calculate the coordinates of the center of gravity (cg_x, cg_y) for department *i curr* in period *t curr*;
- Find the best candidate location along all segments in *vert_segms_l_{t_curr}*, *vert_segms_r_{t_curr}*, *hor_segms_d_{t_curr}*, *hor_segms_u_{t_curr}*, considering both horizontal and vertical orientation of department *i_curr* as described in section 4.3.1.3.2.
- Compare the best candidate position with the position of department i_curr in period $t_curr 1$ (if department i_curr has been placed in previous period) and the position of department i_curr in period $t_curr + 1$ (if department i_curr has been placed in next period) as described in section 4.3.1.3.2.

Select the best position among these positions and initialize the values of $x_{t_curr,i_curr}, y_{t_curr,i_curr}, l_{t_curr,i_curr}$, and w_{t_curr,i_curr} from the best position found. That is, $\pi_{partial}$ is updated.

- Set $TC_{i_curr, t_curr} = \text{cost}$, if the department is placed at the found position. In other words, TC_{i_curr, t_curr} is a cumulative flow cost of department i_curr with placed departments in period t_curr and rearrangement cost of the department in periods t_curr (if $t_curr > 1$ and department i_curr has been placed in period $t_curr - 1$) and $t_curr + 1$ (if $t_curr < T$ and department i_curr has been placed in period $t_curr + 1$).
- Step 3: Modify the vectors *hor_segms_d*, *hor_segms_u*, *vert_segms_l*, and *vert_segms_r* according to the position of the department found;

Set $f(\pi_{partial}) = f(\pi_{partial}) + TC_i \text{ curr, t curr};$

Set *feas_st* = 1, if placing the department *i_curr* at the location found will result in *plant floor feasible* layout plan, and 0 otherwise. Add *i curr* to vector π *pert curr*

If k < NT, then set k = k + 1 and go to Step 2;

Otherwise

Set $f(\pi) = f(\pi_{partial})$ and output the values of x_{ti} , y_{ti} , l_{ti} , w_{ti} , for all t = 1, ..., Tand i = 1, ..., N, and exit the heuristic;

4.3.2 Dual Simplex Technique

The construction heuristic presented in this section sets the values of the variables below using the layout plan (solution) generated by the BSH given earlier, generates an LP formulation of the problem, and solves the LP problem using a dual simplex algorithm. The data from the BSH used to construct the LP formulation are the locations of the departments as well as their lengths and widths for each period (x_{ti} , y_{ti} , l_{ti} , w_{ti}), the orientation of each department in each period (h_{ti}), and department rearrangements (r_{ti}). The data need for the LP formulation are defined as follows.

$$r_{-}p_{iij} = \begin{cases} 1, \text{ If } left_{ijj} = 1 (i.e., x_{ii} + 0.5l_{ii} \le x_{ij} - 0.5l_{ij}); \\ 2, \text{ If } left_{iji} = 1 (i.e., x_{ii} - 0.5l_{ii} \ge x_{ij} + 0.5l_{ij}); \\ 3, \text{ If below}_{iij} = 1 (i.e., y_{ii} + 0.5w_{ii} \le y_{ij} - 0.5w_{ij}); \\ 4, \text{ If below}_{iji} = 1 (i.e., y_{ii} - 0.5w_{ii} \ge y_{ij} + 0.5w_{ij}); \\ h_{ti} = \begin{cases} 1, \text{ if department } i \text{ should be horizontally oriented in period } t; \\ 0, \text{ Otherwise}; \end{cases}$$

$$r_{ti} = \begin{cases} 1, \text{ if department } i \text{ may be rearranged in period } t; \\ 0, \text{ Otherwise}; \end{cases}$$

The LP formulation generated based on these values is shown in Appendix C. Setting the values of h_{ti} and r_{ti} using the layout plan generated by the BSH is straightforward. However, the values of r_p_{tij} , where j > i, are set using the following rules:

If
$$\max((x_{ti} - 0.5l_{ti}) - (x_{tj} + 0.5l_{tj}), (x_{tj} - 0.5l_{tj}) - (x_{ti} + 0.5l_{ti})) \ge \max((y_{ti} - 0.5w_{ti}) - (y_{tj} + 0.5w_{tj}), (y_{tj} - 0.5w_{tj}) - (y_{ti} + 0.5w_{ti}))$$

 $r_{-}p_{tij} = \begin{cases} 1, \text{ If department } i \text{ is to the left of department } j \text{ in current layout plan;} \\ 2, \text{ Otherwise;} \end{cases}$

Else

 $r_{-}p_{ij} = \begin{cases} 3, \text{ If department } i \text{ is below department } j \text{ in current layout plan;} \\ 4, \text{ Otherwise;} \end{cases}$

These rules are used for setting the values of r_p_{tij} , when there are both vertical and horizontal separation between departments *i* and *j* in period *t*. For example, in Figure 4.3 below, in period 1 departments 1 and 3 are separated both vertically and horizontally. In this case, the rules above will ensure that vertical separation is used, since the departments are further apart vertically.

Once the above values are obtained and the LP formulation is constructed, the dual simplex method is used to generate the solution. It is important to note, that the dual simplex method is used so that after an initial layout plan is generated optimally for the above values, the optimal layout plan for different values can be obtained more quickly. More specifically, the optimal tableau is updated (i.e., right hand sides are updated) for the new values, and the dual simplex method quickly determines the optimal solution for the new values (i.e., different DFLP formulation). Consider the small example given in Appendix B. The layout plan generated by the BSH for the example is shown in Figure 4.3. The values of h_{ti} , r_{ti} , and r_p_{tij} are generated as discussed above and given in Table 4.3 and the LP formulation in Appendix C, is shown in Figure 4.4. The OFV of the solution generated by the dual simplex algorithm (i.e., $f(\pi) = 745$) is always the same

or better than the OFV of the solution generated by the BSH (i.e., $f(\pi) = 754.75$), since the BSH is an approximation technique and the dual simplex algorithm is an exact method.



		Peri	od 1	1		Peri	od 2	2	Period 3						
h _{ti}	1	1	0	1	1	1	0	1	1	1	0	1			
r _{ti}	1	1	1	1	0	0	1	0	0	0	0	0			
	r_p _{tij}														
Dept(i/j)	1	2	3	4	1	2	3	4	1	2	3	4			
1	-	4	4	4	-	4	2	4	-	4	2	4			
2	-	-	2	3	-	-	3	3	-	-	3	3			
3	-	-	-	1	-	-	-	1	-	-	-	1			

Table 4.3: The values of r_p_{tij} , h_{ti} , r_{ti} variables set using the layout plan in Figure 4.3



4.4 Improvement Algorithms

The BSH presented in Section 4.3.1 is a construction heuristic. However, improvement heuristics are commonly used to improve solutions generated from construction heuristics. In this section, two improvement heuristics (TS and memetic heuristic) are presented for the DFLP with unequal area departments. Memetic heuristics were first presented by **Norman and Moscato (1989).** These heuristics use the strengths of both genetic algorithm (GA) and TS heuristics (this will be explained later). Tabu search was introduced by **Glover (1986)**, and it is a steepest descent-type heuristic, which uses memory to avoid getting trapped at poor local optima. GA was developed by **Holland (1975)**, which resembles the natural phenomena of survival of the fittest (i.e., most fit in the population survive and reproduce offsprings which are fit).

4.4.1 TS Heuristics

In this section, two TS heuristics are presented. One of the heuristics (TS/BSH) uses the BSH to construct layout plans, and the other TS heuristic (TS/DUAL) uses the LP formulation and the dual simplex algorithm, as discussed earlier, to construct layout plans. Since both heuristics have similarities, first the basic idea of TS and some of its components common to both TS/BSH and TS/DUAL are presented and discussed, and later the notation, components, and pseudo-code specific to each of the heuristics are presented and/or discussed.

Both TS/BSH and TS/DUAL heuristics start with an initial solution obtained from the BSH. This solution is defined as the current solution. Both TS heuristics explore the entire neighborhood of the current solution. In other words, an operation or move is

55

performed on the current solution such that a new layout plan is generated. There are two possible moves, which are defined as follows.

- 1) Exchange the positions of some departments *i* and *j* in some period *t*.
- Move some department *i* in some period *t* to a better location (i.e., non-occupied position on the boundary of department(s)) in period *t*.

TS/BSH uses move (1), and TS/DUAL uses moves (1) and (2). However, move (2) is considered first. Move (1) is considered only if an improved move (2) does not exist. The details of how both heuristics perform these moves are explained later. Nevertheless, all possible moves are considered, and the best admissible move (i.e., either a tabu move that gives the best layout plan ever found or the best move that is not classified as tabu). A move recently performed is defined as tabu, but the tabu restriction may be overridden if the move gives the best solution found thus far (this is called the aspiration criterion). As a result, the best admissible move gives the new current solution. Then the tabu list and best found solution (if necessary) is updated, and the neighborhood of the current solution is explored. This process is repeated until a stopping criterion is satisfied.

4.4.1.1 TS/BSH

As stated previously, the TS/BSH generates new layout plans, by performing move (1). More specifically, move (1) exchanges the positions of department period pairs (i, t) and (j, t) in the solution π to produce a new solution $\overline{\pi}$, and a modification of the BSH is run to obtain a new layout plan and its cost (OFV). More generally, the TS/BSH explores the entire neighborhood of the current solution, and estimates the improvement in the OFV corresponding to each move (1). The total number of moves is the combination of N

pick 2, which is N(N-1)/2. To estimate an improvement in the OFV, which will result from move (1), it is assumed that the center points of exchanged departments will be swapped in the resulting layout plan (ignoring the lengths and widths of the exchanged departments). The improvement in the OFV calculated this way may not be equivalent to the actual improvement in the OFV, since the actual positions of the exchanged departments as well as some other departments may be different when the move is performed; thus, giving a different improvement in the OFV. After estimating the improvement in the OFV for each move, the best N Moves moves are ranked in descending order with respect to the estimated improvement in the OFV. Next, the first move is performed to obtain the candidate solution $\overline{\pi}$ such that the layout plan (x_{ti}, y_{ti}, l_{ti} , w_{ti} for all i and t) is generated using a modification of the BSH, and $f(\bar{\pi})$ is obtained. Note: the BSH is modified such that the displacement of not exchanged departments is minimized. This will be explained in detail below. If the candidate solution $\overline{\pi}$ is better than the current solution π (e.g., $f(\overline{\pi}) < f(\pi)$ and feas st = 1 for $\overline{\pi}$), then the candidate solution $\overline{\pi}$ becomes the current solution (i.e., set $\pi = \overline{\pi}$ and update $x_{ti}, y_{ti}, l_{ti}, w_{ti}$). If the candidate solution $\overline{\pi}$ is worse than the current solution π , the second move is performed to obtain the candidate solution $\overline{\pi}$, and the process is repeated until either $\overline{\pi}$ is better than the current solution π or the best N Moves moves are tried and the best among them is selected and becomes the current solution π (update $x_{ti}, y_{ti}, l_{ti}, w_{ti}$).

As discussed previously, the BSH is modified such that the displacement of not exchanged departments is minimized. More specifically, not exchanged departments before the first exchange in the solution $\overline{\pi}$ have the same locations as well as lengths and widths in the layout plan obtained for solution π . However, for the exchanged

departments, the heuristic tries to force the center of the first exchanged department to be as close as possible to the center of the second exchanged department before generating the new layout plan and vice versa. That is, the center points (x_{ii}, y_{il}) of the exchanged departments as well as all other departments for solution $\overline{\pi}$, after the first exchange in solution π , are used to determine their corresponding centers of gravity (cg_x, cg_y) and $flow(x_cand, y_cand)$. As mentioned previously, the heuristic tries to force the new center point (x_{ii}, y_{il}) of the first exchanged department to be as close as possible to the center of the other exchanged department and vice versa by creating high flow between the exchanged departments.

For example, if the current solution π is as shown in Table 4.4(a), and the layout plan obtained for π is as in Figure 4.5(a), then move (1) corresponding to exchanging departments 1 and 6 in period 2, results in $\overline{\pi}$ and the layout plan, which are shown in Table 4.4(b) and Figure 4.5(b), respectively. When placing any department period pair (*i*, *t*), which precedes both department period pairs (1, 2) and (6, 2), the modified BSH places department *i* in period *t* at the same position as in the layout plan constructed for π (i.e., same x_{ti} , y_{ti} , l_{ti} , w_{ti}). This improves computational time, since the positions of such departments are known, and are not calculated. When placing department period pairs 21 through 36 that are not exchanged, they are in the vicinity of their center points in the layout plan for solution π (i.e., the layout plan in Figure 4.5(a)). In a similar manner, the modification of the BSH ensures that the exchanged department 1 (6) is in the vicinity of the center of department 6 (1) in period 2 in layout plan for solution π .
Dept. Per. Pair	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36
Department(i)	8	8	8	2	2	2	11	11	11	9	5	5	9	4	7	3	1	3	3	7	1	б	б	б	12	12	12	10	10	10	1	5	4	7	4	9
Period(t)	1	3	2	1	3	2	1	3	2	3	1	3	2	3	3	3	1	1	2	1	2	1	2	3	1	2	3	2	3	1	3	2	2	2	1	1
(a)																																				
Dept. Per. Pair	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36
Denartment(i)	8	8	8	2	2	2	11	11	11	0	£	£	0	A	~			~	2	7	6	6	-	6	10	10	12	10	10	10	1	5	A	7	А	0
Deparamente,	١٩.	191	0	4	4	4	11	111	111	2	2	2	2	4	1	۲	1	3	٢		D	0	1	D	12	12	12	10	10	110		2	14	14	4	2
Period(t)	1	3	2	1	2	2	1	3	2	3	1	3	9	4	7	د 3	1	3	3	1	р 2	в 1	1	о З	12	2	3	2	3	1	1 3	2	4	2	4	1

Table 4.4: Solutions: (a) before move (1) is performed; (b) after move (1) is performed



Figure 4.5: Layout plans: (a) before move (1) is performed; (b) after move (1) is performed

Once the best move is selected and performed, the TS/BSH heuristic uses a fixedsize array, called $tabu_list_{iij}$, to keep track of the most recent moves (i.e., the tabu moves). If move (1) is performed, which exchanges the locations of some departments *i* and *j* in period *t*, then the entry $tabu_list_{iij}$, is set to the current iteration (*curr_ts_iter*). When this exchange is considered at a latter iteration, whether the move is tabu or not depends on one of the following conditions.

- If the move is estimated to produce a layout that is not better than the best layout plan found thus far (i.e., f(π) ≥ ofv*), then the duration the move is defined as tabu is *Ten_Len*, and *curr_ts_iter tabu_list_{tij} > Ten_Len* is used to determine if the move is tabu restricted.
- If the move is estimated to produce a layout that is better than the best layout plan found thus far (i.e., $f(\bar{\pi}) < ofv^*$), then the duration the move is defined as tabu is 0.5Ten_Len, and curr_ts_iter tabu_list_{tij} > 0.5Ten_Len is used to determine if the move is tabu restricted.

The heuristic parameters common to both TS/BSH and TS/DUAL are as follows.

curr_ts_iter = Current TS iteration;

 $tabu_list_{tij}$ = Tabu list, which keeps track of the last iteration when each pair of departments *i* and *j* in each period *t* was exchanged;

Ten Len = The number of TS iterations a move is declared tabu;

 $N_Moves =$ Maximum number of best predicted moves to store during each TS iteration where the moves are ranked based on descending order of their estimated improvements;

Max_Duration = Maximum amount of time to run the TS heuristic;

 $x_{ti}^{*}, y_{ti}^{*}, l_{ti}^{*}, w_{ti}^{*}$ = The location, length, and width of each department *i* in each period *t* in best solution found;

 ofv^* = The OFV of the best solution found;

 $feas_st^* = 1$ if the best solution found by the heuristic is *plant floor feasible*, and 0 otherwise;

 $i_{z}, j_{z}, and t_{z} = Vectors$, storing the exchanged department pairs *i* and *j* and periods *t* of best *N_Moves* moves of type 1, ranked in descending order with respect to the estimated improvement in the OFV;

est_impr_z = Vector, storing the estimated improvements of best *N_Moves* moves of type 1 stored in vectors i_{1z} , j_{1z} , and t_{1z} ;

The steps of the TS/BSH heuristic are as follows.

Step 1: Find the initial solution using the BSH and initialize parameters.

Set *curr_ts_iter* = 0; Set π , *ofv*^{*}, x_{ti}^* , y_{ti}^* , l_{ti}^* , w_{ti}^* , and *feas_st*^{*}, from solution constructed by the BSH; For t = 1, ..., T; For i = 1, ..., N - 1; j = i + 1, ..., N;

Initialize tabu list: Set *tabu_list_{tij} = -Ten_Len*;

Step 2: Update current solution OFV and feasibility status as well as best solution found for curr_ts_iter > 0.

Set of v curr = The OFV of the layout plan for solution π (i.e., x_{ti} , y_{ti} , l_{ti} , w_{ti}).

Set *feas_st_curr* = 1, if the departments in the current layout plan fit within plant floor borders, and 0 otherwise;

- If feas_st_curr > feas_st* or feas_st_curr = feas_st* and $ofv_curr < ofv*$ then initialize ofv^* , feas_st*, x^*_{ti} , y^*_{ti} , l^*_{ti} , and w^*_{ti} from the values ofv_curr , feas st curr, x_{ti} , y_{ti} , l_{ti} , and w_{ti} respectively;
- Step 3: Check stopping criterion and update current iteration.If the TS has been running for more than Max_Duration minutes, then

terminate the TS heuristic;

Else set *curr_ts_iter* = *curr_ts_iter* + 1

Step 4: Determine best N Moves moves based on estimated OFV.

Set Z = 0; (current number of best moves of type (1) stored)

- For each period *t*, and each department pair *i* and *j* (j > i)
 - Estimate the improvement *impr_l* in the OFV, which will result if departments *i* and *j* are exchanged in period *t*;

If $(Z < N_Moves \text{ or } impr_1 > est_impr_2)$ and $(curr_ts_iter - tabu_list_{tij} > Ten_Len \text{ or}$ $(ofv_curr - impr_1 < ofv* \text{ and } curr_ts_iter - tabu_list_{tij} > 0.5Ten_Len))$ then store the values of $impr_1$, t, i, j in vectors est_impr_z , t_1_z , i_1_z , j_1_z , respectively; If $Z \ge N_Moves$ Then remove the worst move from these vectors; Else set Z = Z + 1; Step 5: Determine best admissible move from constructing layout plan.

For z = 1, ..., Z

Perform the move (1) corresponding to exchanging departments i_l_z and

 j_l_z in period t_l_z ;

Set *ofv_new* = OFV of the solution, resulting from the move;

Set *feas_st_new* = 1, if the solution, resulting from the move is *plant floor feasible*, and 0 otherwise.

If feas_st_new > feas_st_curr or (feas_st_new = feas_st_curr and ofv_new < ofv curr)

Then go to step 6;

Else continue;

Step 6: Update new solution to current solution.

Set π , of v, x_{ti} , y_{ti} , l_{ti} , w_{ti} , and feas_st from the best solution found in step 5;

Step 7: Set *tabu_list_{tij} = curr_ts_iter*;

Go to Step 2;

4.4.1.2 TS/DUAL

The TS/DUAL is similar to the TS/BSH in Section 4.4.1.1, except that TS/DUAL uses both move (1) and (2), and the moves are performed using the dual simplex technique in Section 4.3.2, instead of the BSH. Besides the most recent moves defined as tabu, a move that rearranges all the departments in some period is also defined as tabu,

since this avoids high rearrangement cost between periods. As discussed earlier, the TS/DUAL starts with an initial layout plan, obtained from the BSH. However, the TS/DUAL initializes the values r_p_{tij} , h_{ti} , and r_{ti} , using the initial layout plan, constructed by the BSH, and runs the dual simplex method using these values, to obtain the optimal tableau (optimal layout plan for the values r_p_{tij} , h_{ti} , and r_{ti}). Next, the TS/DUAL evaluates all moves of type (2) (recall, that move (2) moves some department *i* in some period *t* to an available (not occupied) location on the boundary). If there is at least one move (2) and at least one move (1), which is estimated to improve the current layout plan, the TS/DUAL performs the best estimated move (2), and the next iteration of TS/DUAL is performed. Otherwise, the best N_Moves moves of type 1 are performed, as in TS/BSH. However, the dual simplex method is used, as opposed to the BSH, to perform and evaluate the moves.

The TS/DUAL performs move (1) and (2) by modifying the right hand sides in the current optimal simplex tableau, and quickly re-optimizes the simplex tableau using the dual simplex method. More specifically, the right hand sides are modified in such a way that the new optimal simplex tableau corresponds to new values of r_p_{tij} , h_{ti} , r_{ti} (i.e., different DFLP formulation).

When performing move (1), involving departments *i*, *j* and period *t*, where j > i, the right hand sides in the optimal simplex tableau are modified in such a way that relative positions of departments *i* and *j* in period *t* are swapped. In other words, if $r_p_{tij} = 1$ (or 2) in current optimal simplex tableau, then $r_p_{tij} = 2$ (or 1) after the move is performed. In a similar manner, if $r_p_{tij} = 3$ (or 4) in current optimal simplex tableau, then $r_p_{tij} = 4$ (or 3) after the move is performed. In addition, the orientation of department *i* becomes the

same as the orientation of department *j* before the move, and vise versa. The values of r_{ti} , $r_{t+1,i}$, $r_{t,j}$, and $r_{t+1,j}$ (i.e., rearrangement statuses), corresponding to new optimal simplex tableau may change as well. More specifically, if department *i* (or *j*) is not rearranged in period *t* (and/or *t* + 1), before the move, then department *i* (or *j*) becomes rearranged in period *t* (and/or *t* + 1) after performing the move. In addition to rearranging not rearranged exchanged department(s), move (1) may also force the exchanged department(s) to become not rearranged in period(s) *t* and/or *t* + 1. That is, if the area of department *i* (or *j*) in period *t* - 1 (and/or *t* + 1), then department *i* (or *j*) becomes not rearranged in period *t* (and/or *t* + 1) after the move *t* - 1 (and/or *t* + 1).

In addition to modifying the values of r_p_{tij} , h_{ti} , r_{ti} , for the exchanged departments *i* and *j* in period *t*, move (1) may change the values of r_p_{tij} for some of the not exchanged departments. More specifically, if the relative position of a not exchanged department *d* and an exchanged department *i* (r_p_{tid}) is different from the relative position of department *d* and exchanged department *j* (r_p_{tidj}) in period *t* in the current layout (i.e., current optimal simplex tableau), then the new relative position of department *d* with department *j*, before the move is performed (i.e., in the current layout), and vise versa. This is explained in the example below.

For example, if current values of r_p_{tij} , h_{ti} , and r_{ti} are as in Table 4.5(a), and the layout corresponding to the current optimal layout plan is as in Figure 4.6(a), then performing move (1), exchanging departments 2 and 3 in period 1, results in a layout shown in Figure 4.6(b). The new values of r_p_{tij} , h_{ti} , and r_{ti} are shown in Table 4.5(b).

Note that $r_p p_{123} = 2$ before the move, and $r_p p_{123} = 1$ after the move, for the exchanged departments (see tables 4.5(a) and 4.5(b)). Also, the relative position of a not exchanged department 4 and exchanged department 2 after the move, is the same as the relative position of department 4 with exchanged department 3 before the move is performed (e.g., $r_p p_{124} = 4$ and $r_p p_{134} = 1$ before the move, and $r_p p_{124} = 1$ and $r_p p_{134} = 4$ after the move). Note, the orientations of exchanged departments are changed after the move is performed. Only the values in grayed cells in Table 4.5(a) may change values, and only the bold values are actually changed by performing the move. Hence, only a few right hand side values change in the LP formulation, when a move is performed.

		Peri	od 1	1		Peri	od 2	2		Peri	od 3	3	1			Peri	od 1		I	Peri	od 2	2	I	Peri	od 3	}
h _{ti}	0	1	0	1	0	0	0	1	1	0	0	1	1	h _{ti}	0	0	1	1	0	0	0	1	1	0	0	1
r _{ti}	1	1	1	1	1	1	0	1	1	0	0	1		r _{ti}	1	1	1	1	1	1	1	1	1	0	0	1
	<u>r_p_tij</u>															r_p	e tij									
Dept(i/j)	1	2	3	4	1	2	3	4	1	2	3	4	1	Dept(i/j)	1	2	3	4	1	2	3	4	1	2	3	4
1	-	1	4	1	-	1	4	1	-	4	2	2		1	-	4	1	1	1	1	4	1	-	4	2	2
2	-	-	2	4	-	-	2	З	-	-	2	2		2	•	I	-	1	1	-	2	З	-	-	2	2
3	-	-	-	1	-	-	-	1	-	-	-	3		3	-	-	-	4	-	-	-	1	-	-	-	3
(a)														(b)												

Table 4.5: The values r_p_{tij} , h_{ti} , and r_{ti} : (a) corresponding to current layout plan; (b) after moving to a new solution by exchanging departments 2 and 3 in period 1



Figure 4.6: Layouts plans: (a) before move (1); (b) after performing move (1), exchanging departments 2 and 3 in period 1

The move (2) is equivalent to moving some department i in some period t to a better, non-occupied position on the boundary of department(s), as stated above. For each orientation (i.e., horizontal or vertical) of department i, along each side of each department d, d = 1, ..., N and $d \neq i$, the heuristic identifies feasible rectangular regions within which the department may move parallel to the side of department d, without overlapping with other departments (e.g., the graved region in Figure 4.7 below) and within boundary of plant, if current layout is feasible. Similar to the BSH, the TS/DUAL uses the center gravity for department i to find the estimated best position of the department within the rectangular feasible region. Move (2) is considered for every department *i* in each period *t*, and the department *i* 2^* in period *t* 2^* corresponding to the best move (as defined previously, which is based on estimated improvement (*impr* 2^*) and feasibility status) is selected. As mentioned earlier, if there are no improving moves of move (2), move (1) is considered. Also, it should be noted that the tabu list is not used when evaluating type (2) moves. In other words, an improving move (2) is never considered to be a tabu move. The new orientation of department i 2^* in period t 2^* as well as whether the moved department is rearranged or not in periods t 2^* and t $2^* + 1$ in the new solution are stored in orient 2* as well as rearr 2* and rearr next 2*, respectively. The leftmost x, right most x, lower most y, and upper most y coordinates of the feasible region identified for department i_2^* are stored in $l_x_2^*$, $r_x_2^*$, $l_y_2^*$, and $u \neq 2^*$, respectively. The TS/DUAL performs move (2) by modifying the right hand sides in the current optimal simplex tableau in such a way, that after re-optimizing the tableau, the department $i 2^*$ is at the best position within the rectangle given by values l x 2*, r x 2*, l y 2*, and u y 2*. To achieve this, the changes to the right hand sides

ensure that the relative position of any not changed department *d* with an exchanged department i_2^* in period t_2^* is the same, as the relative position of department *d* with rectangle, given by the values of $l_x_2^*$, $r_x_2^*$, $l_y_2^*$, and $u_y_2^*$. Figure 4.7 is an example of a move (2), exchanging department 1 in period 3 (i.e., $i_2^* = 1$ and $t_2^* = 3$). The feasible rectangular region, within which the estimated best position for department 1 in period 3 was found using the center of gravity, is the grayed region (i.e., $l_x_2^* = 5$, $r_x_2^* = 15$, $l_y_2^* = 7$, and $u_y_2^* = 11$), and the layout obtained by performing move (2) is shown in Figure 4.7(b). Note, departments 2 and 3 are below the grayed region, and they are below department 1, after the move is performed (see Figure 4.7(b)). In a similar manner, department 4 is to the left of the grayed region, and department 4 is to the left of the grayed region, and department 4 is to the left of the grayed region, and department 4 is to the left of department 1, after the move is performed.



In addition to modifying the relative positions of departments, the changes to right hand sides ensure that the orientation of department i_2^* in period t_2^* as well as the rearranged statuses of the exchanged department i_2^* in periods t_2^* and $t_2^* + 1$

correspond to the values of *orient_2** as well as *rearr_2** and *rearr_next_2**, respectively. Also, it should be noted, that the real improvement, resulted from move (2) is always greater or equal to the estimated improvement. The reason for this is that the center of gravity point is used to find the estimated best position within the feasible region, which is not guaranteed to give the optimal position. Dual simplex, on the other hand, finds the best position for the department within this region.

The steps for the TS/DUAL are the same as the steps for the TS/BSH in Section

4.4.1.1, except that Step 3 should be modified as follows.

Step 3: Check stopping criterion, update current iteration, and perform move (2), if there is an improving move (2).

If the TS has been running for more than *Max_Duration* minutes, then terminate the TS heuristic;

Else set *curr ts iter* = *curr ts iter* + 1;

Set *impr_2** = -1;

Set feas_st_2* = 0; (plant floor feasibility status corresponding to best move (2))

For t = 1, ..., T

For i = 1, ..., N

Calculate estimated improvement impr_2 corresponding to move

(2) involving department i and period t.

Set *feas_st_2* to 1, if move (2) involving department *i* and period *t* is estimated to result in *plant floor feasible* layout plan, and 0 otherwise.

```
If feas st 2 > feas st 2^* or
```

 $feas_st_2 = feas_st_2*$ and $impr_2 > impr_2*$

Store the coordinates of rectangle, within which the dept *i*

should be placed in l_x_{2*} , r_x_{2*} , l_y_{2*} , and u_y_{2*} ; Set $i_{2*} = i$; $t_{2*} = t$;

Set
$$impr_2^* = impr_2$$
; $feas_st_2^* = feas_st_2$;

Set the following:

 $orient _ 2^* = \begin{cases} 1, \text{ if department } i \text{ should be horizontally oriented in period } t; \\ 0, \text{ otherwise;} \end{cases}$ $rearr _ 2^* = \begin{cases} 1, \text{ if department } i \text{ may rearrange in period } t; \\ 0, \text{ otherwise;} \end{cases}$

*rearr_next_2** = $\begin{cases} 1, \text{ if department } i \text{ may rearrange in period } t+1; \\ 0, \text{ otherwise}; \end{cases}$

If *feas_st_2**>*feas_st_curr* or (*feas_st_2**=*feas_st_curr* and *impr_2**>0) then Perform the best move (2) found, and go to Step 2;

4.4.2 Memetic Heuristic

In this section, a memetic heuristic is presented for the DFLP with unequal areas, which is comprised of GA and TS. The GA generates a number of solutions (chromosomes) and adds them to the new generation of solutions P_g , where g is the current iteration of the GA. The GA uses one of two types of solutions. The first solution type is a vector of department period pairs π , as used in the BSH, and the second solution type is a special structure from which the solution, similar to solution π , used by the BSH may be obtained. The memetic heuristic starts by randomly generating the initial population of solutions in P_1 . More specifically, each solution in P_1 is generated randomly, and the OFV and *plant floor feasibility* status is evaluated by constructing the layout plan using the BSH. On the other hand, each solution in P_g , g > 1, is either randomly generated (i.e., mutation operation is used), or it is generated from two solutions, randomly selected from population P_{g-1} (i.e., crossover operation is used). The chromosomes, obtained by applying crossover operation, inherit features from both

parent chromosomes. At each generation g of the GA, Max_Num_Cross chromosomes are generated using crossover operation, but only Gen_Size (Gen_Size < Max_Num_Cross) best chromosomes are kept in the new generation P_g . The generated chromosome is added to P_g , only if it is better then the worst chromosome in P_g and a chromosome similar to π has not already been added to P_g . After the new population P_g , g > 1, is generated, Num_Rand_Chrom ($Num_Rand_Chrom < Gen_Size$) chromosomes are randomly generated (mutation operation), and replace the worst chromosomes in P_g . The chromosomes in each generation P_g are stored in such a way, that higher quality chromosomes precede lower quality chromosomes.

The technique used here was used by **Drezner (2003)**. That is, the number of generated chromosomes is greater than the population size, and only the best *Gen_Size* unique chromosomes are kept in the generation. However, **Drezner (2003)** used this technique to solve the QAP. Since only the best solutions are kept in the population, and the crossover operation is used, good features of parent solutions are passed to next generations. In addition, the mutation operation diversifies the search space. However, unlike heuristics which use steepest descent, the GA may obtain solutions in the vicinity of the local optima, without ever converging to local optima. Therefore, either the TS/BSH or TS/DUAL is run on some solutions, generated by the GA (i.e. solutions, which are considered to be good, and satisfy some criteria), only to obtain an improved best solutions, stored in x^*_{ti} , y^*_{ti} , l^*_{ti} , w^*_{ti} . Therefore, the GA is combined with either TS/BSH or TS/DUAL, and the resulting heuristic is called a memetic heuristic. The memetic heuristic in this dissertation is called MEM/BSH if it uses TS/BSH, and it is called MEM/DUAL, otherwise.

One of two sets of criteria is used, to determine if the TS (i.e., TS/BSH or TS/DUAL) is run on the solution, generated by the GA. The TS is run on the solution generated by the GA based on the following criteria.

- *g* ≥ *TS_Start_Generation*, where *TS_Start_Generation* is the generation number starting from which the TS heuristic can begin;
- $f(\pi) \le ofv^*$ or $(f(\pi) ofv^*)/f(\pi) \le \beta_{BSH}$, where π is the solution used by the

BSH to obtain the corresponding layout, and $\beta_{BSH} \in (0, 1)$;

Note: the criteria above are used for the MEM/BSH heuristic.

The TS is run on the solution generated by the GA based on the following criteria.

- $g \ge TS_Start_Generation$
- $f(\pi) \leq ofv^{GA}$ or $(f(\pi) ofv^{GA})/f(\pi) \leq \beta_{DUAL}$, where ofv^{GA} is the OFV of the best solution generated either by mutation or by crossover operations, and $\beta_{DUAL} \in (0, 1)$;

Note: these criteria are used for the MEM/DUAL heuristic. The greater values of β_{BSH} or β_{DUAL} will result in TS being run on poor solutions, generated by the mutation or crossover operations; this may result in a waste of computational time. The reason that different criteria was applied for the MEM/DUAL is that it is computationally more expensive to perform moves in the TS/DUAL than in the TS/BSH. The second criteria assures that the TS is run only on relatively good initial solutions when β_{DUAL} is small enough. Alternatively, the first criteria may be used in MEM/DUAL with a small value of β_{BSH} . The disadvantage of this approach, however, is that most of the solutions will not satisfy the criteria, if at some iteration the TS/DUAL generates new best solution, and the

value of ofv^* becomes much lower than the OFVs of solutions ($f(\pi)$ s) generated randomly or by crossover.

The stopping criterion for both the TS/BSH and TS/DUAL heuristics, when used in memetic heuristic, is the maximum number of *Max_Num_TS_Iter* consecutive iterations without improvement over the best solution found by the TS heuristic. The stopping criterion for both the MEM/BSH and MEM/TSH heuristics is the maximum amount of time *Max_Duration*, to run the heuristics.

4.4.2.1 The Chromosomes Used by the Memetic Heuristic

The memetic heuristic uses one of two types of chromosomes. The type 1 chromosome is used for problems with relatively low rearrangement costs. On the other hand, the type 2 chromosomes are used for problems, with relatively high rearrangement costs. It is important to note that the memetic heuristic in this dissertation uses either type 1 chromosomes or type 2 chromosomes, but not both.

4.4.2.1.1 Type 1 Chromosomes

The type 1 chromosome π , used by the GA is similar to the solution defined for the BSH. The type 1 chromosomes are generated either randomly (by mutation) or by applying the crossover operation on two parent chromosomes. Two different techniques are used, for mutation. Technique 1 randomly generates departments, and then randomly generates periods for each department. An example of a chromosome generated this way is demonstrated in Table 4.6(a). When the layout plan is constructed for solution π in Table 4.6(a), the BSH will place department 6 in periods 1, 2, and 3, before department 7

is placed in any period. Next department 7 will be placed in periods 1, 3, and 2, and so

on.

(<i>i</i> , <i>t</i>) Pair	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36
Department(i)	б	б	б	7	7	7	12	12	12	3	3	3	4	4	4	5	5	5	11	11	11	2	2	2	1	1	1	8	8	8	10	10	10	9	9	9
Period(t)	1	2	3	1	3	2	1	2	3	3	1	2	1	3	2	3	1	2	1	3	2	1	3	2	3	1	2	1	3	2	2	3	1	1	2	3
(a)																																				
(i t) Dain					-	~	-	0		10																								24		
(<i>t</i> , <i>t</i>) rair	1	2	3	4	5	0	1	Ø	י און	101	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36
(<i>i</i> , <i>i</i>) rair Department(<i>i</i>)	8	2	3	4 11	5 7	0 8	7 9	8 12	9 10	10 10	11	12 1	13 9	14 5	15 6	16 б	17 6	18 10	19 3	20 12	21 1	22 4	23 7	24 8	25 4	26 11	27 2	28 4	29 2	30 11	31 3	32 1	33 5	34 9	35 2	36 5
Department(i) Period(t)	8	2 3	3 7 2	4 11 2	5 7 1	0 8 3	7 9 1	8 12 3	9 10 3	10 10 2	11 12 1	12 1 1	13 9 3	14 5 2	15 6 2	16 6 3	17 6 1	18 10 1	19 3 3	20 12 2	21 1 3	22 4 3	23 7 3	24 8 2	25 4 2	26 11 3	27 2 2	28 4 1	29 2 3	30 11 1	31 3 2	32 1 2	33 5 3	34 9 2	35 2 1	36 5 1

Table 4.6: Two type 1 chromosomes for the problem instance in Appendix A:(a) generated using technique 1; (b) generated using technique (2)

Technique 2 randomly generates department period pairs. An example of a chromosome generated this way is shown in Table 4.6(b). When constructing the layout plan for the solution π in Table 4.6(b), department 8 will be placed in period 1, then department 3 will be placed in period 1 and so on. Chromosomes generated using technique 1, tend to produce better layouts for problem instances, in which a large number of departments have relatively high rearrangement costs, since layout plans with less rearrangements are generated. On the other hand, the chromosomes generated by technique 2, add variety to the population. As a result, layout plans with more rearrangements are generated from the chromosomes in Table 4.6(a) and 4.6(b) are shown in Figure 4.8(a) and 4.8(b), respectively. As it can be seen, there are less rearranged departments in the layout plan shown in Figure 4.8(a). The probability of generating random solutions using technique 1 is γ , and the probability of generating random solutions using technique 2 is $(1-\gamma)$, where $\gamma \in (0,1)$. Before the mutation operation is performed, the crossover operation is performed, which is discussed next.



Figure 4.8: Layout plans: (a) generated using chromosome in Table 4.6(a); (b) generated using chromosome in Table 4.6(b)

As in most permutation problems such as the proposed problem, the crossover

operation may produce infeasible chromosomes, if a technique is not used to generate

feasible chromosomes. The following technique is used to generate feasible chromosome

 π , when performing the crossover operation to parents π^1 and π^2 :

Step 0: Set *k1* = 0.2*NT*; *k2* = 0.5*NT*; *cross_point* = 1; *num_cross_points* = 0;

Step 1: Set *num_cross_points* = *num_cross_points* + 1;

Add crossover point *cross_point* to vector *cross_points*;

Set *cross_point* = *cross_point* + Random number between *k1* and *k2*;

If *cross_point* $\geq NT$ then go to Step 2;

Else go to Step 1;

Step 2: Set cross_point = NT; num_cross_points = num_cross_points + 1; Add crossover point cross_point to vector cross_points; Set r = 1;

- Step 3: Copy the genes (department period pairs) $cross_points_r$ through $cross_points_{r+1}$ from chromosome π^1 to the same positions in chromosome π , Set r = r + 2; If $r < num_cross_points$ go to Step 3 Else go to Step 4
- Step 4: Copy all the department period pairs in chromosome π^2 which have not been copied from π^1 into positions in π which have not been filled, while preserving the precedence order of department period pairs in π^2 ;

An example of generating child 1 chromosome π , from parent chromosomes π^1 and π^2 is shown in Figure 4.9. In this example, the vector of crossover points, *cross_points* is {1, 11, 28, 36}. The heuristic copies the department period pairs 1 through 11 and 28 through 36 from chromosome π^1 into chromosome π . Then the department period pairs in chromosome π^2 , which have not been already copied to chromosome π from π^1 are copied at positions 12 through 27 in π . Note that the precedence relationship of department period pair (1, 1) and (7, 3) is the same in both chromosomes π and π^2 . Note, to generate child 2, we change the order of parents. Hence, each parent pair produces two offsprings.

Parent 1 (π^1)																																				
(i, t) Pair	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36
Department(<i>i</i>)	4	4	8	12	10	5	3	1	7	4	8	1	11	б	5	11	1	11	12	2	9	7	9	3	10	б	б	5	9	8	2	2	7	3	12	10
Period(t)	2	3	3	2	2	3	2	2	1	1	2	3	2	3	1	1	1	3	3	1	3	3	1	3	1	1	2	2	2	1	2	3	2	1	1	3
Parent 2 (π^2)																																				
(i, t) Pair	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36
Department(i)	1	2	1	7	4	11	7	12	3	б	9	1	12	9	11	б	5	9	4	5	11	8	2	б	10	4	10	7	3	8	3	10	12	5	8	2
Period(t)	2	2	1	3	2	1	1	1	1	3	1	3	2	2	3	2	2	3	1	3	2	2	3	1	2	3	1	2	2	1	3	3	3	1	3	1
Child Chromos	om	.e (π)												-						-															
(i, t) Pair	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36
Department(<i>i</i>)	4	4	8	12	10	5	3	1	7	4	8	1	7	11	б	9	1	11	б	9	11	б	10	3	12	5	2	5	9	8	2	2	7	3	12	10
Period(t)	2	3	3	2	2	3	2	2	1	1	2	1	3	1	3	1	3	3	2	3	2	1	1	3	3	1	1	2	2	1	2	3	2	1	1	3

Figure 4.9: Applying crossover operation to parent chromosomes π^1 and π^2

4.4.2.1.2 Type 2 Chromosomes

The Type 2 chromosome (solution) μ has the following representation:

 $\mu = \{depts, periods\}$

where $depts = (depts_1, depts_2, ..., depts_N)$, periods = $(periods_1, periods_2, ..., period_N)$, $depts_k$ = department in position k, and $periods_{depts_k}$ = vector of periods for department $depts_k$. To construct the layout plan for solution μ , the memetic heuristic first generates solution π from μ , and then the BSH constructs the layout plan for the solution π . For example, the department period pairs, generated from type 2 chromosome μ in Table 4.7 are similar to the department period pairs π in Table 4.6(a). For instance, first department 6 ($depts_1 = 6$) is placed in period 1, 2, and 3 ($periods_{depts_1} = periods_6 = \{1, 2, 3\}$) in that order, in the layout plan. Next, department 7 ($depts_2 = 7$) is placed and so on. As it can be seen, type 2 chromosomes are equivalent to type 1 chromosomes, generated using technique 1 in Section 4.4.2.1.1. It should be noted, however, that the crossover operation described above is not guaranteed to produce solutions, which are similar to type 2 chromosomes.

depts	6, 7, 12, 3	, 4, 5, 11, 2,	1, 8, 10, 9
periods 1	3	1	2
periods 2	1	3	2
periods 3	3	1	2
periods 4	1	3	2
periods 5	3	1	2
periods 6	1	2	3
periods 7	1	3	2
periods ₈	1	3	2
periods 9	1	2	3
periods 10	2	3	1
periods 11	1	3	2
periods 12	1	2	3

Table 4.7: Type 2 chromosome (μ)

Similar to type 1 chromosomes, type 2 chromosomes are generated either randomly (mutation operation) or by applying the crossover operation to a pair of type 2 chromosomes. To generate a random solution, the memetic heuristic randomly generates the vector of department numbers *depts*. Next, for each department *depts*, the vector *periods*_{*depts*_{*k*}} is generated randomly. The crossover operation is applied to ordered list of departments *depts* in parent chromosomes μ^1 and μ^2 to generate ordered list of departments vector *depts* in child chromosome μ . The array *periods* in new chromosome μ is the same as the array *periods* in chromosome μ^1 . Although the crossover operation applies to only the array *depts*, not *periods*, in type 2 chromosomes, it is similar to the crossover operation in Section 4.4.2.1.1. However, the values used for *k1* and *k2* are *0.2N* and *0.5N*, respectively.

4.4.2.2 The Pseudo-code for the Memetic Heuristic

The steps of the memetic heuristics are given below, but first some additional notation is defined.

 π_worst = worse solution in current population $g(P_g)$. Recall, in P_g , solutions are ordered in ascending order based on OFV.

feas $st(\pi)$ = feasibility status of layout plan obtained for solution π ,

feas_st(π *worst*) = feasibility status of layout plan obtained for solution π *worst*;

Step 1: Initialize parameters.

Initialize parameters Gen_Size , $Max_Num_Cross > Gen_Size$, Num_Rand_Chrom , $Max_Num_TS_Iter$, $Max_Duration$, γ , β , TS Start Generation; Determine if type 1 or type 2 chromosomes should be used (i.e., π or μ). Is discussed in Chapter 5;.

Set g = 1;

Set *curr_parents_offspr_count* = 0; (number of chromosomes generated from current pair of chromosomes using the crossover operator)

Set $ofv^* = Big$ Number;

Set *feas* $st^* = 0$;

Step 2: Start new population.

Set *chromosome_count* = 0; (number of chromosomes generated at iteration g)

Step 3: Generate chromosome.

If *chromosome_count* \geq *Max_Num_Cross* then go to step 5;

Else

If g = 1 then

Randomly generate chromosome π (or μ);

Else

If *curr* parents offspr count = 0 then

Randomly pick two chromosomes $\pi'(\mu')$ and $\pi''(\mu'')$ from the generation P_{g-1} and set $\pi^1 = \pi'(\mu^1 = \mu')$ and $\pi^2 = \pi''(\mu^2 = \mu'')$; Set *curr parents offspr count = curr parents offspr count* + 1;

Else

Set
$$\pi^1 = \pi'' (\mu^1 = \mu'')$$
, and $\pi^2 = \pi' (\mu^2 = \mu')$;
Set *curr parents offspr count* = 0;

Generate chromosome $\pi(\mu)$ from $\pi^{1}(\mu^{1})$ and $\pi^{2}(\mu^{2})$ by applying crossover operation;

Generate the layout plan corresponding to chromosome $\pi(\mu)$ using the BSH; Step 4: Add chromosome $\pi(or \mu)$ to new population, and possibly run TS/BSH or TS/DUAL with $\pi(or \mu)$ as a starting solution.

Set $\pi_{worst} = P_{g,Gen Size}$;

If chromosome_count < Gen_Size or $(feas_st(\pi) > feas_st(\pi_worst)$ or $(feas_st(\pi) = feas_st(\pi_worst)$ and $f(\pi) < f(\pi_worst)$) If $feas_st(\pi) > feas_st^*$ or $(feas_st(\pi) = feas_st^*$ and $f(\pi) < ofv^*$) Initialize ofv^* , $feas_st^*$, x^*_{ti} , y^*_{ti} , l^*_{ti} , and w^*_{ti} from the values $f(\pi)$, $feas_st(\pi)$, x_{ti} , y_{ti} , l_{ti} , and w_{ti} respectively;

Add chromosome π to new generation P_g . When adding the chromosome to the generation, make sure that the higher quality solutions precede lower quality solutions. Also, if *chromosome_count* \geq *Gen_Size*, then drop the worst (*Gen_Size*-th) chromosome from P_g ;

If the solution, generated satisfies the criteria, described in Section 4.4.2, then Run the TS heuristic (TS/BSH or TS/DUAL), with π (or μ) as a starting solution. The TS heuristic will modify the values of ofv^* , $feas_st^*$, x^*_{ti} , y^*_{ti} , l^*_{ti} , and w^*_{ti} , if it finds a better solution, than the best solution found thus far.

Set *chromosome_count* = *chromosome_count* + 1, and go to Step 3;

Step 5: Check stopping criterion and add random solutions to the new population.If the heuristic has been running for more than Max_Duration minutes, then go to Step 6;

Else

Remove the last Num_Rand_Chrom (worst) chromosomes from P_g , and add Num_Rand_Genes randomly generated chromosomes to the generation; While adding new chromosomes to the generation, make sure that higher quality solutions precede lower quality solutions;

Set g = g + 1, and go to Step 2;

Step 6: Output the best solution (i.e., x^*_{ti} , y^*_{ti} , l^*_{ti} , w^*_{ti} for all i = 1, ..., N and t = 1, ..., T), and terminate the heuristic;

CHAPTER 5

COMPUTATIONAL RESULTS

5.1 Datasets

The only data set found in the literature for the DFLP with unequal area departments with fixed shapes is the dataset presented in Yang and Peters (1998). The first problem, P6, is a 6-department problem with 6 periods, and the second problem, P12, is a 12-department problem with 4 periods. Yang and Peters (1998) consider low and high rearrangement cost of 50 and 200, respectively, for each department. Dunker et al. (2005) solved the problems in Yang and Peters (1998), but used the rearrangement cost of 19 and 50 for problems P6 and P12, respectively, to allow for more department rearrangements in the solution. In addition, the problems in this dataset use an initial layout in period 0 (i.e., the relative positions as well as lengths and widths of departments in initial layout prior to period 1 are specified); therefore, the rearrangement costs in period 1 should be considered. As a result, two datasets are used in this dissertation. Dataset 1 consists of problems P6 and P12 from **Dunker** et al. (2005), and dataset 2 was generated from the dataset for the DFLP with equal area departments in **Balakrishnan** et al. (2000). See the characteristics (i.e., number of departments and periods) of dataset 1 and 2 in Tables 5.1(a) and 5.1(b), respectively.

The dataset in **Balakrishnan** *et al.* (2000) contains 24 5-period problems and 24 10-period problems. Only 5-period problems from **Balakrishnan** *et al.* (2000) dataset were used to generate problem instances for dataset 2. The problem instances in dataset 2 were generated by randomly selecting 4 problems from the 5-period problems with 6

departments, 15 departments, and 30 departments in **Balakrishnan** *et al.* (2000). Thus, 12 problem instances were generated where the department dimensions were generated randomly as follows. Since departments in **Balakrishnan** *et al.* (2000) have unit sizes, the lengths and width of departments were randomly generated in the range between [0.5, 1.5], to minimize the change in the relationship between the flow and rearrangement costs in the original problems. In addition, the rearrangement costs of six problems (i.e., two 6 department problems, two 15 department problems, and two 30 department problems) were multiplied by 1.5. The problems selected to be modified are the

problems, with the largest values of the $\sum \sum R_{ti}$

$$\sum_{t=1}^{T} \sum_{i=1}^{N} R_{ti} / \sum_{t=1}^{T} \sum_{i=1}^{N} \sum_{j=i+1}^{N} F'_{tij}$$

Problem#	Num. of Periods (T)	Num. of Depts. (N)	Plant Floor Length	Plant Floor Width
P6	6	6	30	30
P12	4	12	50	50
(9)				

(a)				
Problem#	Num. of Periods (T)	Num. of Depts. (N)	Plant Floor Length	Plant Floor Width
P01	5	6	5	5
P02	5	6	5	5
P03	5	6	5	5
P04	5	6	5	5
P05	5	15	9	9
P06	5	15	9	9
P07	5	15	9	9
P08	5	15	9	9
P09	5	30	12	12
P10	5	30	12	12
P11	5	30	12	12
P12	5	30	12	12

(b)

Table 5.1: Datasets: (a) from Dunker *et al.* (2005); (b) generated from data set in Balakrishnan *et al.* (2000)

5.2 Parameter Settings

5.2.1 Parameter Settings for the TS Heuristics

The parameters that need to be set for TS/BSH and TS/DUAL are the *Max_Duration*, *Ten_Len* and *N_Moves*. The values used to set these parameters are shown in Table 5.2. The value *Max_Duration* is the same for all heuristics (TS/BSH, TS/DUAL, MEM/BSH, and MEM/DUAL) for each problem to make sure that comparisons between the heuristics are done fairly.

Good values for the values of the *N_Moves* parameter were found by experimentation. The larger values of this parameter result in better solutions at each iteration, since the heuristics perform a number of type 1 moves, and pick the move, resulting in the best improvement (remember that the TS heuristics use only estimated improvement in OFV, and the real improvement is found only after performing the move). However, using too large values for this parameter will result in smaller number of iterations.

Finally the value of *Ten_Len* parameter is determined by multiplying the number of department pairs in all periods (i.e, the size of neighborhood) by 0.15 or 0.30. Smaller values of *Ten_Len* tend to result in poor solutions, since the TS heuristics spend to much time repeating the same moves, or the heuristic may get trapped in local optima (i.e., cycling). On the other hand, using too large values for the *Ten_Len* parameter results in restricted solution space, and too many good moves may be overlooked.

Buchlow#	Max_Duration	M. Mouros	Ten_	Len
rroblem#	(hours)	IV_MOVES	TS/BSH	TS/DUAL
P6	1	10	30	30
P12	5	20	15	30
(a)				
Problem#	Max_Duration	N Monas	Ten_	Len
1100lein#	(hours)	IV_INOVES	TS/BSH	TS/DUAL
P01	2	10	30	30
P02	2	10	30	30
P03	2	10	30	30
P04	2	10	30	30
P05	4	20	30	30
P06	4	20	30	30
P07	4	20	30	30
P08	4	20	15	30
P09	8	30	15	30
P10	8	30	15	30
P11	8	30	30	30
P12	8	30	15	30

(b)

Table 5.2: Parameter settings for TS heuristics; (a) for problems in dataset 1; (b) for problems in dataset 2

5.2.2 Parameter Settings for the Memetic Heuristics

In addition TS heuristics, to parameters used in the parameters TS Start Generation, β_{BSH} (or β_{DUAL}), Gen Size, Max Num Cross, Num Rand Chrom, Max Num TS Iter should be set for the memetic heuristics. The values used for the parameters Max Duration, Ten Len, and N Moves, are similar to the ones used by TS heuristics. The value used for the parameter TS Start Generation was 60. This means that the TS heuristic (TS/BSH or TS/DUAL) is not applied to the solutions generated during the first 60 generations. The memetic heuristic generates 60 generations in a very short time, even for larger problems, if the TS is not used. Starting from the *TS_Start_Generation* generation, the TS heuristic is applied on some promising solutions. Since the TS heuristic depends on initial solutions, the memetic heuristics obtain better solutions using this technique, than by applying the TS heuristic starting from the first generation, using large amounts of computation time on poor solutions.

As it was discussed in Section 4.4.2.1, two types of chromosomes can be used in the memetic heuristics (i.e., type 1 and type 2 chromosomes). Type 1 chromosomes were used for solving problems P6 and P12 from dataset 1 and problems P01, P02, P03, and P04 from dataset 2. Type 2 chromosomes were used, when solving 15- and 30-department problems in dataset 2. An easy way to find out which type of chromosome to use, is to run the memetic heuristic two times, for some number of generations each time, without applying the TS. First run can be performed using type 1 chromosomes, and the second run can be performed using type 2 chromosomes. The type 1 or type 2 chromosomes can be selected, based on which run resulted in a better solution. This technique was used to determine which type of chromosome to use.

If type 1 chromosome is used, than the parameter γ , discussed in Section 4.4.2.1.1 should be set. The value of 0.1 was used for this parameter in all cases, whenever applicable. The number of random solutions, *Num_Rand_Chrom*, generated at each generation was set to 0.1*Gen_Size*. The values used for parameters *Gen_Size*, *Max_Num_Cross*, *Max_Num_TS_Iter*, and β_{BSH} and β_{DUAL} are shown in Table 5.3.

		MEM/BSH Heuri	istic Parameters			MEM/DUAL Heu	ristic Parameters	
Problem#	Gen_Size	Max_Num_Cross	Max_Num_TS_Iter	$\boldsymbol{\beta}_{BSH}$	Gen_Size	Max_Num_Cross	Max_Num_TS_Iter	$\boldsymbol{\beta}_{DUAL}$
P6	30	120	500	0.1	10	40	400	0.01
P12	40	160	1000	0.1	15	60	600	0.01
(a)								
		MEM/BSH Heuri	stic Parameters			MEM/DUAL Heu	ristic Parameters	
Problem#	Gen_Size	Max_Num_Cross	Max_Num_TS_Iter	β_{BSH}	Gen_Size	Max_Num_Cross	Max_Num_TS_Iter	$\boldsymbol{\beta}_{DUAL}$
P01								
P02	20	120	500	0.1	10	40	200	0.01
P03	50	120	500	0.1	10	40	500	0.01
P04								
P05								
P06	40	160	1000	0.05	15	60	600	0.005
P07	40	100	1000	0.05	15	00	000	0.005
P08								
P09								
P10	50	200	2000	0.05	15	60	600	0.005
P11	50	200	2000	0.03	15	00	000	0.005
P12								

Table 5.3: Parameter settings for memetic heuristics: (a) for problems in dataset 1; (b) for problems in dataset 2

5.3 Test Environment

(b)

All metaheuristics were coded using C++Builder 6, and the problems were solved on a set of Dell Optiplex GX620 computers. The computers had Pentium IV, 3.6GHz processors, 2GB of memory, and Windows XP operating system.

5.4 Experimental Results

Each problem in data sets 1 and 2 were solved by all four metaheuristics. Since the memetic heuristics (i.e., MEM/BSH and MEM/DUAL) are stochastic, and the outcome can be different for different runs, every problem was solved 5 times by each of the two memetic heuristics. The OFVs and the runtimes (i.e., the times in which the heuristics found the best solution) of the TS/BSH and MEM/BSH heuristics are shown in Tables 5.4 and 5.5, respectively, and the OFVs and the runtimes of the TS/DUAL and MEM/DUAL heuristics are shown in Tables 5.6 and 5.7, respectively. The summary of the results of all metaheuristics is shown in Table 5.8. As it can be seen, the MEM/BSH

obtained better results than TS/BSH on all problems (average percent improvement of 2.307) except on problem P06 from dataset 2. The MEM/DUAL obtained better results than TS/DUAL on all problem instances (average percent improvement of 2.439). Therefore for the current values of parameters, the memetic heuristics are superior to the TS heuristics.

The MEM/DUAL heuristic obtained best results on 5 problem instances out of 14 problem instances (i.e., problem instance P6 in dataset 1, and problem instances P01 -P04 in dataset 2). Therefore, the MEM/DUAL performed better on smaller problems (6 department problems), and the MEM/BSH performed better on larger problems (i.e., 12department problem instance in dataset 1, and 15- and 30-department problem instances in dataset 2). In addition, MEM/DUAL performed better than TS/BSH on 12 problem instances. The reason, that the MEM/DUAL does not perform as well as the MEM/BSH, is that it is computationally more expensive to perform moves using DUAL simplex technique (remember, TS/DUAL uses DUAL technique to perform the moves), than to perform moves using the BSH. Therefore, the TS/BSH is able to perform more iterations during the execution of the heuristic. On the other hand, the dual based heuristics (i.e., TS/DUAL and MEM/DUAL) has a better chance to obtain a global optimal solution, given that it generates sufficient number of diverse solutions, since each generated solution corresponds to a solution to a MILP formulation of the problem, with the values of integer variables preset. The MEM/BSH, on the other hand, may never obtain the global optimal solution (i.e., best layout plan), since the solutions are generated using construction type heuristic (i.e., the modified BSH). In addition, the position of each department being placed depends on the positions of already placed departments, and not so much by the departments placed later. Hence, this is may be a drawback of the BSH heuristics.

The summary of the results of the proposed techniques and the results of the dynamic genetic algorithm from Dunker et al. (2005), on the problem instances in dataset 1, is shown in Table 5.9. Since **Dunker** et al. (2005), perform the analysis of their technique, considering the case, when the position of the initial layout is centered inside the plant floor area, similar approach was used in this dissertation. The BSH handles initial layout by assuming that the initial layout is an additional period in which all of the department positions are fixed. To solve the problems in dataset 1 by the proposed techniques, based on the DUAL simplex technique (i.e., TS/DUAL and MEM/DUAL), the initial layout is ignored, and the cost of rearranging all of the departments in period 1 is added to the final OFV. The initial layout was not considered in DUAL simplex technique, since it would result in too many infeasible layouts during the execution of the heuristics. Dunker et al. (2005) obtained better results on the 6-department problem instance with 6 periods than any of the proposed techniques (percent improvement of 1.46). However, all four proposed techniques outperformed the technique by **Dunker** et al. (2005) on a larger problem instance, 12-department problem instance with 4 periods (percent improvement of 1.72). The worst solutions obtained by any of the four proposed techniques were better than the best solution obtained by Dunker et al. (2005). The execution times, during which the proposed heuristics obtained better solutions, than the technique by **Dunker** et al. (2005), are shown in Table 5.10. As it can be seen, the longest time it took to outperform the technique by Dunker et al. (2005) is 260 seconds. However, it should be noted, that Dunker et al. (2005) used Pentium IV, 1.5 GHz computer. The reason that **Dunker** *et al.* (2005) outperformed the proposed techniques on a smaller problem instance may be contributed to the fact that they use a relaxed MILP formulation, in which the only binary variables are the variables used for orientations and rearrangement statuses of departments (i.e., h_{bi} and r_{bi}). The proposed heuristics on the other hand do not use binary variables, and the orientations and rearrangement statuses of departments are determined by the heuristics (i.e., BSH or TS/BSH). As **Dunker** *et al.* (2005) mention, the number of binary variables in the reduced mixed integer problems increases linearly, which could theoretically result in an exponential increase in computational time.

				MEM/I	BSH Solutio	n OFVs					
Problem#	Initial Sol. (BSH)	TS/BSH	Run1	Run2	Run3	Run4	Run5	Aver. OFV (MEM/ BSH)	Worst. OFV (MEM/ BSH)	Best OFV (MEM/ BSH)	Improvement of Best MEM/BSH Sol. Over TS/BSH Solution (%)
P6	6,967.9	6,648.3	6,615.6	6,628.9	6,637.3	6,619.5	6,619.5	6,624.2	6,637.3	6,615.6	0.49
P12	29,779.6	26,845.5	26,826.3	26,774.9	26,938.1	26,789.1	26,640.4	26,793.7	26,938.1	26,640.4	0.77

				MEM/I	BSH Solutio	n OFVs					
Problem#	Initial Sol. (BSH)	TS/BSH	Run1	Run2	Run3	Run4	Run5	Aver. OFV (MEM/ BSH)	Worst. OFV (MEM/ BSH)	Best OFV (MEM/ BSH)	Improvement of Best MEM/BSH Sol. Over TS/BSH Solution (%)
P01	97,795.0	96,013.0	95,239.3	94,813.5	95,421.4	94,849.9	95,070.2	95,078.9	95,421.4	94,813.5	1.27
P02	104,277.0	99,371.9	97,333.7	96,905.7	96,747.4	96,897.5	97,367.6	97,050.4	97,367.6	96,747.4	2.71
P03	87,282.1	85,185.2	83,893.4	83,821.3	83,821.3	83,893.4	83,821.3	83,850.1	83,893.4	83,821.3	1.63
P04	111,295.6	106,611.9	104,031.6	104,776.7	104,838.7	103,837.1	104,614.1	104,419.6	104,838.7	103,837.1	2.67
P05	508,806.6	468,186.9	464,249.9	465,468.1	465,121.7	465,707.5	461,718.8	464,453.2	465,707.5	461,718.8	1.40
P06	465,412.5	437,095.4	445,516.9	440,195.7	443,542.4	445,797.6	442,666.7	443,543.9	445,797.6	440,195.7	-0.70
P07	502,905.7	481,511.8	475,397.5	473,665.0	470,239.3	476,377.3	475,676.7	474,271.2	476,377.3	470,239.3	2.40
P08	575,306.4	540,766.0	529,286.8	524,136.6	528,043.6	525,868.5	533,259.4	528,119.0	533,259.4	524,136.6	3.17
P09	617,073.2	576,867.3	567,922.5	580,857.5	558,896.3	564,291.3	570,478.9	568,489.3	580,857.5	558,896.3	3.22
P10	652,395.2	595,500.0	560,992.2	564,863.4	560,934.4	563,137.9	557,420.1	561,469.6	564,863.4	557,420.1	6.83
P11	607,983.2	557,695.5	555,914.0	555,036.6	553,652.6	550,638.6	555,455.8	554,139.5	555,914.0	550,638.6	1.28
P12	582,723.8	543,113.4	543,064.3	545,130.8	540,057.3	542,301.0	531,703.7	540,451.4	545,130.8	531,703.7	2.15

(b)

Table 5.4: Summary of TS/BSH and MEM/BSH heuristic results: (a) for problems in dataset 1; (b) for problems in dataset 2

					MEM/	BSH Solu	tions			
Problem#	TS/BSH	Run1	Run2	Run3	Run4	Run5	Average Runtime	Shortest Runtime	Longest Runtime	Total Runtime
P6	27.77	11.17	0.03	49.38	0.14	0.05	12.15	0.03	49.38	60
P12	82.05	100.06	146.24	107.34	174.61	23.75	110.40	23.75	174.61	300

			MEM/BSH Solutions									
Problem#	TS/BSH	Run1	Run2	Run3	Run4	Run5	Average Runtime	Shortest Runtime	Longest Runtime	Total Runtime		
P01	99.87	58.48	26.06	65.56	18.90	49.04	43.61	18.90	65.56	120		
P02	83.35	3.97	15.99	11.59	1.63	25.14	11.67	1.63	25.14	120		
P03	117.09	45.56	5.35	0.65	15.23	3.63	14.09	0.65	45.56	120		
P04	0.00	3.32	15.80	4.25	1.90	26.48	10.35	1.90	26.48	120		
P05	61.46	157.78	13.72	31.67	51.02	153.84	81.61	13.72	157.78	240		
P06	71.31	14.76	106.97	238.94	102.28	100.49	112.69	14.76	238.94	240		
P07	40.32	29.26	20.50	200.21	27.65	209.86	97.50	20.50	209.86	240		
P08	120.48	101.45	19.22	199.73	23.05	13.80	71.45	13.80	199.73	240		
P09	216.06	442.34	319.84	116.43	139.47	110.51	225.72	110.51	442.34	480		
P10	33.75	309.69	22.83	220.25	99.47	156.79	161.81	22.83	309.69	480		
P11	234.03	212.52	241.98	410.28	469.96	141.95	295.34	141.95	469.96	480		
P12	403.19	539.06	240.13	11.45	576.87	445.48	362.60	11.45	576.87	480		

(b)

Table 5.5: Summary of TS/BSH and MEM/BSH heuristic execution times in minutes: (a) for problems in dataset 1; (b) for problems in dataset 2

				MEM/DUAL Solution OFVs							
Problem#	Initial Sol. (BSH)	TS/DUAL	Run1	Run2	Run3	Run4	Run5	Aver. OFV (MEM/ DUAL)	Worst. OFV (MEM/ DUAL)	Best OFV (MEM/ DUAL)	Improvement of Best MEM/DUAL Sol. Over TS/DUAL Solution (%)
P6	6,986.9	6,680.0	6,602.5	6,608.5	6,620.0	6,605.5	6,606.0	6,608.5	6,620.0	6,602.5	1.17
P12	29,582.8	27,059.5	26,817.0	26,888.0	26,959.5	26,978.5	26,907.0	26,910.0	26,978.5	26,817.0	0.90

				MEM/D	UAL Soluti	on OFVs					
Problem#	Initial Sol. (BSH)	TS/DUAL	Run1	Run2	Run3	Run4	Run5	Aver. OFV (MEM/ DUAL)	Worst. OFV (MEM/ DUAL)	Best OFV (MEM/ DUAL)	Improvement of Best MEM/DUAL Sol. Over TS/DUAL Solution (%)
P01	97,795.0	97,059.3	94,776.5	94,776.5	94,776.5	94,776.5	94,776.5	94,776.5	94,776.5	94,776.5	2.41
P02	104,277.0	99,383.3	96,182.4	96,182.4	96,182.4	96,182.4	96,182.4	96,182.4	96,182.4	96,182.4	3.33
P03	87,282.1	86,469.6	83,785.2	84,457.2	83,785.2	84,457.2	84,011.3	84,099.2	84,457.2	83,785.2	3.20
P04	111,295.6	106,241.2	103,797.5	103,406.4	103,797.5	103,406.4	103,406.4	103,562.8	103,797.5	103,406.4	2.74
P05	508,806.6	472,801.3	461,929.7	461,895.0	467,155.6	464,541.3	466,545.3	464,413.4	467,155.6	461,895.0	2.36
P06	465,412.5	445,292.3	447,364.3	449,973.4	449,869.0	445,346.8	444,394.8	447,389.6	449,973.4	444,394.8	0.20
P07	502,905.7	481,750.3	476,414.8	481,163.5	476,258.9	479,385.3	476,984.0	478,041.3	481,163.5	476,258.9	1.15
P08	575,306.4	542,660.3	535,274.9	536,324.0	530,531.6	537,490.0	530,865.0	534,097.1	537,490.0	530,531.6	2.29
P09	617,073.2	583,568.2	579,177.7	580,601.2	583,380.7	583,631.7	578,406.2	581,039.5	583,631.7	578,406.2	0.89
P10	652,395.2	615,757.9	573,907.8	590,776.7	592,421.7	596,668.7	587,038.2	588,162.6	596,668.7	573,907.8	7.29
P11	607,983.2	574,625.2	567,177.6	574,649.0	567,533.5	561,262.3	557,497.7	565,624.0	574,649.0	557,497.7	3.07
P12	582,723.8	558,066.0	546,362.6	553,271.8	541,129.2	549,650.9	553,965.4	548,876.0	553,965.4	541,129.2	3.13

(b)

Table 5.6: Summary of TS/DUAL and MEM/DUAL heuristic results: (a) for problems in dataset 1; (b) for problems in dataset 2

		MEM/DUAL Solutions								
Problem#	TS/DUAL	Run1	Run2	Run3	Run4	Run5	Average Runtime	Shortest Runtime	Longest Runtime	Total Runtime
P6	0.21	58.41	9.89	1.90	34.85	4.93	21.99	1.90	58.41	60
P12	1.40	28.12	20.62	65.07	1.65	26.53	28.40	1.65	65.07	300

			MEM/DUAL Solutions										
Problem#	TS/DUAL	Run1	Run2	Run3	Run4	Run5	Average Runtime	Shortest Runtime	Longest Runtime	Total Runtime			
P01	1.36	0.05	0.05	0.73	0.06	0.05	0.19	0.05	0.73	120			
P02	1.12	26.03	52.72	46.96	72.30	32.18	46.04	26.03	72.30	120			
P03	0.00	105.42	2.03	28.27	0.32	58.97	39.00	0.32	105.42	120			
P04	0.11	8.88	45.32	37.05	27.22	1.03	23.90	1.03	45.32	120			
P05	19.84	212.68	79.41	65.41	136.76	229.04	144.66	65.41	229.04	240			
P06	23.90	49.91	238.46	175.63	18.55	191.39	134.79	18.55	238.46	240			
P07	30.53	150.16	201.00	88.06	10.66	93.68	108.71	10.66	201.00	240			
P08	39.16	59.52	50.92	90.26	1.08	185.49	77.45	1.08	185.49	240			
P09	417.57	46.19	453.85	54.73	135.43	238.53	185.74	46.19	453.85	480			
P10	129.99	46.60	69.92	7.73	108.21	102.32	66.95	7.73	108.21	480			
P11	226.58	434.85	118.52	39.43	151.70	135.61	176.02	39.43	434.85	480			
P12	271.20	380.04	86.84	82.20	122.13	357.94	205.83	82.20	380.04	480			

(b)

Table 5.7: Summary of TS/DUAL and MEM/DUAL heuristic execution times in minutes: (a) for problems in dataset 1; (b) for problems in dataset 2

Problem#	TS/BSH	MEM/BSH	TS/DUAL	MEM/DUAL
P6	6,648.3	6,615.6	6,680.0	6,602.5
P12	26,845.5	26,640.4	27,059.5	26,817.0
(a)				
Problem#	TS/BSH	MEM/BSH	TS/DUAL	MEM/DUAL
P01	96,013.0	94,813.5	97,059.3	94,776.5
P02	99,371.9	96,747.4	99,383.3	96,182.4
P03	85,185.2	83,821.3	86,469.6	83,785.2
P04	106,611.9	103,837.1	106,241.2	103,406.4
P05	468,186.9	461,718.8	472,801.3	461,895.0
P06	437,095.4	440,195.7	445,292.3	444,394.8
P07	481,511.8	470,239.3	481,750.3	476,258.9
P08	540,766.0	524,136.6	542,660.3	530,531.6
P09	576,867.3	558,896.3	583,568.2	578,406.2
P10	595,500.0	557,420.1	615,757.9	573,907.8
P11	557,695.5	550,638.6	574,625.2	557,497.7
P12	543,113.4	531,703.7	558,066.0	541,129.2

(b)

Table 5.8: Summary of the results of metaheuristics: (a) for problems in dataset 1; (b) for problems in dataset 2

	Dunker et.al, 2005	TS/BSH	MEM/BSH	TS/DUAL	MEM/DUAL	
Average OFV	6,569.0		6,624.2		6,608.5	
Worst OFV	6,613.0	6,648.3	,648.3 6,637.3	6,680.0	6,620.0	
Best OFV	6,507.5		6,615.6		6,602.5	
Average Runtime	29.4		12.15		21.99	
Fastest Runtime	8.4	27.77	0.03	0.21	1.90	
Longest Runtime	57.86		49.38		58.41	

	Dunker et.al, 2005	TS/BSH	MEM/BSH	TS/DUAL	MEM/DUAL
Average OFV	27,748.0		26,793.7		26,910.0
Worst OFV	28,344.5	26,845.5	26,938.1	27,059.5	26,978.5
Best OFV	27,098.5		26,640.4		26,817.0
Average Runtime	160		110.40		28.40
Fastest Runtime	54	82.05	23.75	1.40	1.65
Longest Runtime	314		174.61		65.07

(b)

Table 5.9: Results of the proposed heuristics, and the dynamic genetic algorithm by Dunker *et al.* (2005): (a) on problem instance P6 in dataset 1; (b) on problem instance P12 in dataset 1

TS/BSH		\mathbf{N}	ŒM/BS	н		TS/DUAT	MEM/DUAL				
	Run 1	Run 2	Run 3	Run 4	Run 5	15/DUAL	Run 1	Run 2	Run 3	Run 4	Run 5
154	223	410	243	197	123	84	31	64	260	99	213

Table 5.10: Execution times in seconds, during which the proposed techniques found better solutionsthan the best solution obtained by Dunker *et al.* (2005) on problem instance P12 in dataset 1
CHAPTER 6

CONCLUSIONS

6.1 Summary of Research

The DFLP with unequal area and fixed department shapes is a combinatorial optimization problem, and there exists no exact technique, which optimally solves the problem in polynomial time. Therefore, two construction type heuristics and four improvement type heuristics were developed to solve the problem in reasonable computational time. The heuristics are BSH, dual simplex method, TS/BSH, TS/DUAL, MEM/BSH, MEM/DUAL. The BSH is a construction type heuristic, which constructs the layout by placing departments on the boundary of placed departments. An LP formulation with a dual simplex method constructs layout plans for the proposed problem. The TS/BSH and TS/DUAL are tabu search heuristics, which use the BSH and dual simplex method, respectively, to generate layout plans. Finally, MEM/BSH and MEM/DUAL are memetic heuristics, which use the TS/BSH and TS/DUAL, respectively. The memetic heuristics (i.e., MEM/BSH and MEM/TS) were found to obtain better solutions than the tabu search heuristics (i.e., TS/BSH and TS/TS). In addition, MEM/DUAL generated better solutions, than BSH based improvement heuristics on small problem instances. On the other hand, BSH based improvement heuristics were found to be superior on larger problem instances. All improvement type heuristics found better solutions for the larger problem instance than the technique by Dunker et al. (2005).

6.2 Future Research

The following issues may be considered in future research:

- Modify the heuristics, to consider variable shape departments, and I/O stations not at the center points of departments.
- Improve the BSH based heuristics, to perform better on smaller problem instances.

REFERENCES

Ahmad, A. R., O. A. Basir, M. H. Imam, and K. Hassanein. 2006. An efficient, effective, and robust decoding heuristic for metaheuristics-based layout optimization. *International Journal of Production Research* 44: 1545-1567.

Al-Hakim, L. 2000. On solving facility layout problems using genetic algorithms. *International Journal of Production Research* 38: 2573-2582.

Apple, J. M. and M. P. Deisenroth. 1972. A computerized plant layout analysis and evaluation technique (PLANET), *A.I.I.E Technical Papers, Twenty-third Conference. Anaheim, California, USA*.

Armour, G. C. and E. S. Buffa. 1963. A heuristic algorithm and simulation approach to relative location of facilities. *Management Science* 9: 294-309.

Balakrishnan, J. and C. H. Cheng. 1998. Dynamic layout algorithms: a state-of-the-art survey. *Omega* 26: 507-521.

Balakrishnan, J., C. H. Cheng, and D.G. Conway. 2000. An improved pair-wise exchange heuristic for the dynamic plant layout problem. *International Journal of Production Research* 38: 3067-3077.

Bazaraa, M. S. 1975. Computerized layout design: a branch and bound approach. *AIIE Transactions* 7: 432-438.

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

Buffa, E. S., G. C. Armour, and T. E. Vollman. 1964. Allocating facilities with CRAFT. *Harvard Business Review* 42: 136-158.

Burkard, R. E. and T. A. Bonniger. 1983. A heuristic for quadratic boolean programs with applications to quadratic assignment problems. *European Journal of Operational Research* 13: 374-386.

Burkard, R. E. and F. Rendl. 1984. A thermodynamically motivated simulation procedure for combinatorial optimization problems. *European Journal of Operational Research* 17: 169-174.

Castillo, I. and T. Westerlund. 2005. An ε-accurate model for optimal unequal-area block layout design. *Computers and Operations Research* 32: 429-447.

Castillo, I., J. Westerlund, S. Emet, and T. Westerlund. 2005. Optimization of block layout design problems with unequal areas: A comparison of MILP and MINLP optimization methods. *Computers and Chemical Engineering* 30: 54-69.

Chiang, W-C. and C. Chiang. 1998. Intelligent local search strategies for solving facility layout problems with the quadratic assignment problem formulation. *European Journal of Operational Research* 106: 457-488.

Cohoon, J. P., S. U. Hegde, W. N. Martin, and D. S. Richards. 1991. Distributed genetic algorithms for the floorplan design problem. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 10: 483-492.

Conway, D. G. and M. A. Venkataramanan. 1994. Genetic search and the dynamic facility layout problem. *Computers & Operations Research* 21: 955-960.

Corry, P. and E. Kozan. 2004. Ant colony optimisation for machine layout problems. *Computational Optimization and Applications* 28: 287-310.

Dorigo, M., V. Maniezzo, and A. Colorni. 1996. Ant system: optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics, Part B* 26: 29-41.

D'Souza, G. and B. B. Mohanty. 1986. Interactive multilevel, multicriteria dynamic approach facility layout analysis. *Fall Industrial Engineering Conference*, p. 152-161.

Dunker, T., G. Radons, and E. Westkamper. 2003. A coevolutionary algorithm for a facility layout problem. *International Journal of Production Research* 41: 3479–3500.

Dunker, T., G. Radons, and E. Westkamper. 2005. Combining evolutionary computation and dynamic programming for solving a dynamic facility layout problem. *European Journal of Operational Research* 165: 55-69.

Drezner, Z. 2003. A new genetic algorithm for the quadratic assignment problem. *INFORMS Journal on Computing* 15: 320-330.

Drezner, Z. 2008. Extensive experiments with hybrid genetic algorithms for the solution of the quadratic assignment problem. *Computers and Operations Research* 35: 717-736.

Frances, R. L. and J. A. White. 1974. Facility layout and location: an analytical approach. Englewood Cliff, N.J.: Prentice-Hall.

Gambardella, L. M., E. D. Taillard, and M. Dorigo. 1999. Ant colonies for the quadratic assignment problem. *Journal of Operational Research Society* 50: 167–176.

Garces-Perez, J., D. A. Schoenefeld, and R. L. Wainwright. 1996. Solving facility layout problems using genetic programming. *Genetic Programming. Proceedings of the First Annual Conference*, p 182-190.

Gau, K. Y. and R. D. Meller. 1999. An iterative facility layout algorithm. *International Journal of Production Research* 37: 3739-3758.

Gilmore, P. C. 1962. Optimal and suboptimal algorithms for the quadratic assignment problem. *Journal of the Society for Industrial and Applied Mathematics* 10: 305-313.

Glover, F. 1986. Future paths for integer programming and links to artificial intelligence. *Computers & Operations Research* 13: 533-549.

Glover, F. 1989. Tabu Search-Part I. ORSA Journal on Computing 1: 190-206.

Glover, F. 1990. Tabu Search-Part II. ORSA Journal on Computing 2: 4-32.

Goetschalckx, M. 1992. An interactive layout heuristic based on hexagonal adjacency graphs. *European Journal of Operational Research* 63: 304-321.

Gomory, R. E. and T. C. Hu. 1961. Multi-terminal network flows. *Journal of the Society for Industrial and Applied Mathematics* 9: 551-570.

Hassan, M. M. D., G. L. Hogg, and D. R. Smith. 1986. SHAPE: a construction algorithm for area placement evaluation. *International Journal of Production Research* 24: 1283-1295.

Heragu, S. S. 1989. Knowledge based approach to machine cell layout. *Computers & Industrial Engineering* 17: 37-42.

Hillier, F. S. 1963. Quantitative tools for plant layout analysis. *The Journal of Industrial Engineering* 14: 33-40.

Holland, J. H. 1975. Adaptation in natural and artificial systems. Ann Arbor: University of Michigan Press.

Houshyar, A. and B. White. 1993. Exact optimal solution for facility layout: deciding which pairs of locations should be adjacent. *Computers & Industrial Engineering* 24: 177-187.

Imam, M. H. and M. Mir. 1993. Automated layout of facilities of unequal areas. *Computers & Industrial Engineering* 24: 355-366.

Imam, M. H. and M. Mir. 1998. Cluster boundary search algorithm for building-block layout optimization. *Advances in Engineering Software* 29: 165-173.

Kaku, B. K. and J. B. Mazzola. 1997. A tabu-search heuristic for the dynamic plant layout problem. *INFORMS Journal on Computing* 9: 374-384.

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

Kirkpatrick, S., C. D. Gelatt, and M. P. Vecchi. 1983. Optimization by simulated annealing. *Science* 220: 671-680.

Koopmans, T. C. and M. J. Beckmann. 1957. Assignment problems and the location of economic activities. *Econometrica* 25: 53-76.

Kusiak, A. and S. S. Heragu. 1987. The facility layout problem. *European Journal of Operational Research* 29: 229-251.

Lacksonen, T. A. and E. E. Enscore. 1993. Quadratic assignment algorithms for the dynamic layout problems. *International Journal of Production Research* 31: 503-517.

Lacksonen, T. A. 1994. Static and dynamic layout problems with varying areas. *Journal of the Operational Research Society* 45: 59-69.

Lacksonen, T.A. 1997. Preprocessing for static and dynamic facility layout problems. *International Journal of Production Research* 35: 1095-1106.

Lawler, E. L. 1963. The quadratic assignment problem. *Management Science* 9: 586-599.

Lee, R. C. and J. M. Moore. 1967. CORELAP-computerized relationship layout planning, *Industrial Engineering* 18: 195-200.

Lengauer, T. 1990. Combinatorial algorithms for integrated circuit layout. *John Wiley & Sons, New York*.

Li, T. and J. Mashford. 1990. A parallel genetic algorithm for quadratic assignment, *Proceedings of the ISMM International Conference. Parallel and Distributed Computing, and Systems*, p 391-394.

Liggett, R. S. 2000. Automated facilities layout: past, present and future. *Automation in Construction* 9: 197-215.

Liu, D. and H. Teng. 1999. An improved BL-algorithm for genetic algorithm of the orthogonal packing of rectangles. *European Journal of Operational Research* 112: 413–420.

McKendall, A. R., J. S. Noble, and C. M. Klein. 1999. Facility layout of irregular-shaped departments using a nested approach. *International Journal of Production Research* 37: 2895-2914.

McKendall, A. R. and J. Shang. 2006. Hybrid ant systems for the dynamic facility layout problem. *Computers & Operations Research* 33: 790-803.

McKendall, A. R., J. Shang, and S. Kuppusamy. 2006. Simulated annealing heuristics for the dynamic facility layout problem. *Computers & Operations Research* 33: 2431-2444.

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

Meller, R. D., V. Narayanan, and P. H. Vance. 1999. Optimal facility layout design. *Operations Research Letters* 23: 117-127.

Montreuil, B. and H. D. Ratliff. 1989. Utilizing cut trees as design skeletons for facility layout. *IIE Transactions* 21: 136-143.

Montreuil, B. 1990. A modelling framework for integrating layout design and flow network design. *Progress in Material Handling and Logystics* 2: 95-115

Montreuil, B. and U. Venkatadri. 1991. Strategic interpolative design of dynamic manufacturing systems layouts. *Management Science* 37: 682-694.

Montreuil, B. and A. Laforge. 1992. Dynamic layout design given a scenario tree of probable futures. *European Journal of Operational Research* 63: 271-286.

Montreuil, B., U. Venkatadri, and H. D. Ratliff. 1993. Generating a layout from a design skeleton. *IIE Transactions* 25: 3-15.

Nicol, L. M. and R. H. Hollier. 1983. Plant Layout in Practice. *Material Flow* 3: 177-188.

Norman, M.G. and P. Moscato. 1989, A competitive and cooperative approach to complex combinatorial search. *Caltech Concurrent Computation Program, Report 826*.

Pardalos, P. M. and J. V. Crouse. 1989. A parallel algorithm for the quadratic assignment problem. *Proceedings of Supercomputing*, p 351-360.

Rosenblatt, M. J. 1986. The dynamics of plant layout. *Management Science* 32: 76-86.

Sahni, S. and T. Gonzalez. 1976. P-complete approximation problems. *Journal of the Association for Computing Machinery* 23: 555-565.

Schnecke, V. and O. Vornberger. 1997. Hybrid genetic algorithms for constrained placement problems. *IEEE Transactions on Evolutionary Computation* 1: 266-277.

Shayan, E. and L. Al-Hakim. 1999. Improved methods in solution of unequal-sized facilities layout problems by genetic algorithms. *Advanced Manufacturing Processes, Systems, and Technologies*, p 133-141.

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

Sherali, H. D., B. M. P. Fraticelli, and R. D. Meller. 2003. Enhanced model formulations for optimal facility layout. *Operations Research* 51: 629-644.

Sherwani, N. 1993. Algorithms for VLSI physical design automation. Kluwer, Norwell.

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

Skorin-Kapov, J. 1990. Tabu search applied to the quadratic assignment problem. *ORSA Journal on Computing* 2: 33-45.

Tam, K. Y. and S. G. Li. 1991. A hierarchical approach to the facility layout problem. *International Journal of Production Research* 29: 165-184.

Tam, K. Y. 1992. Genetic algorithms, function optimization, and facility layout design. *European Journal of Operational Research* 63: 322-346.

Tam, K. Y. 1992. A simulated annealing algorithm for allocating space to manufacturing cells. *International Journal of Production Research* 30: 63-87.

Tam, K. Y. and S. K. Chan. 1998. Solving facility layout problems with geometric constraints using parallel genetic algorithms: experimentation and findings. *International Journal of Production Research* 36: 3253-3272.

Tate, D. M. and A. E., Smith. 1995. Unequal-area facility layout by genetic search. *IIE Transactions* 27: 465-472.

Tompkins, J. A., J. A. White, Y. A. Bozer, E. H. Frazelle, J. M. A. Tanchoco, and J. Trevino. 1996. *Facilities Planning*. New York, N.Y.: John Wiley.

Tong, X. 1991. SECOT: a sequential construction technique for facility design, *Unpublished Doctoral Dissertation, Department of Industrial Engineering, University of Pittsburgh, Pittsburgh, PA, USA.*

Valenzuela, C. L. and P. Y. Wang. 2001. Breeding Normalized Postfix Expressions for the Facility Layout Problem. *MIC'2001-4th Metaheuristics International Conference*, p 261-265.

Wascher, G. and J. Merker. 1997. A comparative evaluation of heuristics for the adjacency problem in facility layout planning. *International Journal of Production Research* 35: 447-466.

Welgama, P. S. and P. R. Gibson. 1993. A construction algorithm for the machine layout with fixed pick-up and drop-off points. *International Journal of Production Research* 31: 2575-2590.

Welgama, P. S. and P. R. Gibson. 1995. Computer-aided facility layout-a status report. *International Journal of Advanced Manufacturing Technology* 10: 66-77.

Wilhelm, M. R. and T. L. Ward. 1987. Solving quadratic assignment problems by "Simulated Annealing". *IIE Transactions* 19: 107-119.

Yang, T. and B. A. Peters. 1998. Flexible machine layout design for dynamic and uncertain production environments. *European Journal of Operational Research* 108: 49-64.

Appendix A. Problem Instance Used to Demonstrate the BSHs

Number of periods is 3 (i.e., T = 3);

Number of departments is 12 (i.e., N = 12);

Departments are not restricted to horizontal or vertical orientations (i.e., $DeptOrient_{ti} = 0$,

for *t* = 1,..., *T* and *i*, *j* = 1,..., *N*);

Rearrangement cost is 50 for all departments in all periods (i.e., $R_{ti} = 50$, for t = 1,..., 3and i, j = 1,..., 12);

Dept (i)	1	2	3	4	5	6	7	8	9	10	11	12
Sh_{i}	2	2	2	2	2	3	2	2	2	3	2	1
Lng _{ii}	2	3	3	2	3	4	3	2	3	4	3	2

Table A.1: Shorter and longer side lengths of departments

Period	i/j	2	3	4	5	6	7	8	9	10	11	12
	1	9	3	б	1	10	18	5	19	10	13	13
	2	-	9	5	2	13	17	16	8	0	18	16
	3	-	-	0	2	2	7	15	16	18	16	19
	4	-	-	-	8	19	1	17	15	1	15	0
	5	-	-	-	-	10	1	17	15	3	7	12
1	6	-	-	-	-	-	11	13	13	14	13	6
	7	-	-	-	-	-	-	18	6	10	4	14
	8	-	-	-	-	-	-	-	11	19	6	10
	9	-	-	-	-	-	-	-	-	4	11	12
	10	-	-	-	-	-	-	-	-	-	15	3
	11	-	-	-	-	-	-	-	-	-	-	6
	-		-	-			-	-			-	_
	1	4	0	8	4	19	8	5	6	15	0	3
	2	-	0	18	16	7	4	8	0	13	5	9
	3	-	-	3	0	12	5	11	19	2	4	11
	4	-	-	-	0	10	1	16	3	5	2	12
	5	-	-	-	-	13	18	12	1	1	1	12
2	6	-	-	-	-	-	2	2	14	18	5	9
	7	-	-	-	-	-	-	3	12	16	5	5
	8	-	-	-	-	-	-	-	19	3	2	6
	9	-	-	-	-	-	-	-	-	9	13	19
	10	-	-	-	-	-	-	-	-	-	7	10
	11	-	-	-	-	-	-	-	-	-	-	6
	1	3	12	8	17	13	6	15	7	14	5	18
	2	-	3	3	14	19	11	11	1	19	11	3
	3	-	-	18	14	4	16	13	7	7	12	12
	4	-	-	-	б	2	5	13	5	8	10	7
	5	-	-	-	-	4	16	18	4	6	16	14
3	6	-	-	-	-	-	1	9	3	6	2	0
	7	-	-	-	-	-	-	15	0	8	15	12
	8	-	-	-	-	-	-	-	17	11	6	4
	9	-	-	-	-	-	-	-	-	8	11	18
	10	-	-	-	-	-	-	-	-	-	1	11
	11	-	-	-	-	-	-	-	-	-	-	3

Table A.2: Cost to transport materials a unit distance between departments (i.e., the values of F'_{tij})

Appendix B. Problem Instance Used to Demonstrate the Dual Simplex Based Heuristics

Number of periods is 3 (i.e., T = 3);

Number of departments is 12 (i.e., N = 4);

Departments are not restricted to horizontal or vertical orientations (i.e., $DeptOrient_{ti} = 0$,

t = 1, ..., N, i, j = 1, ..., T;

Rearrangement cost is 50 is all departments and periods (i.e., $R_{ti} = 50$, i = 1,..., 3, i, j=1,..., 12);

Dept (i)	1	2	3	4
Sh_{i}	4	5	5	4
Lng _{ii}	5	7	б	4

Table B.1: Shorter and longer side lengths of departments

Period	ij	2	3	4				
	1	9	3	б				
1	2	-	б	5				
	3	-	-	0				
	1	4	0	8				
2	2	-	0	18				
	3	-	-	3				
	1	3	12	8				
3	2	-	3	3				
	3	-	-	18				

Table B.2: Cost to transport materials a unit distance between departments (i.e., the values of F'_{tij})

Appendix C. LP Formulation Used by the Dual Simplex Based Heuristics

In addition to the indexes, parameters, and variables defined in section 4.2 the following parameters and variables are used by the LP formulation used in the dual simplex method and TS/DUAL.

P = Penalty incurred if departments span outside of boundaries of plant floor. The value of P is set to the value of the OFV of solution obtained by solving the problem using the BSH when the plant floor length and width are 3L and 3W correspondingly;

$$M = 3\max(L, W);$$

 r_p_{tij} , h_{ti} , and r_{ti} are variables used by dual simplex based heuristic, defined in section 4.3.2;

 sp_h = The span of departments in horizontal direction in all periods in excess of plant floor length (i.e., sp_h = max(0, max(l_{ti}) - L));

 sp_v = The span of departments in vertical direction in all periods in excess of plant floor length (i.e., sp_v = max(0, max(w_{ti}) - W));

The LP formulation of the problem is as follows.

Minimize total cost =

$$\sum_{t=1}^{T} \sum_{i=1}^{N} \sum_{j>i}^{N} F'_{tij} \left(x_{p_{tij}} + y_{p_{tij}} \right) + \sum_{i=1}^{N} \sum_{t=2}^{T} R_{ti} r_{ti} + P * sp_h + P * sp_v$$
(C.1)

Subject to:

$$x_{ti} + 0.5l_{ti} - x_{tj} + 0.5l_{tj} \le \begin{cases} 0, \text{ If } r_{-}p_{tij} = 1\\ M, \text{ Otherwise} \end{cases} \quad \forall t, i, j > i$$
(C.2)

$$-x_{ti} + 0.5l_{ti} + x_{tj} + 0.5l_{tj} \le \begin{cases} 0, \text{ If } r_{p_{tij}} = 2\\ M, \text{ Otherwise} \end{cases} \quad \forall t, i, j > i$$
(C.3)

$$y_{ti} + 0.5w_{ti} - y_{tj} + 0.5w_{tj} \le \begin{cases} 0, \text{ If } r_{p_{tij}} = 3\\ M, \text{ Otherwise} \end{cases} \quad \forall t, i, j > i$$
(C.4)

$$-y_{ti} + 0.5w_{ti} + y_{tj} + 0.5w_{tj} \le \begin{cases} 0, \text{ If } r_{p_{tij}} = 4\\ M, \text{ Otherwise} \end{cases} \quad \forall t, i, j > i$$
 (C.5)

$$x_{ti} + 0.5l_{ti} - sp_h \le L \quad \forall t, i \tag{C.6}$$

$$-x_{ti} + 0.5l_{ti} \le 0 \quad \forall t, i \tag{C.7}$$

$$y_{ti} + 0.5w_{ti} - sp_v \le W \quad \forall t, i$$
(C.8)

$$-y_{ti} + 0.5w_{ti} \le 0 \quad \forall t, i \tag{C.9}$$

$$x_{ti} - x_{tj} - x_{j} p_{ij} \le 0 \quad \forall t, i, j > i$$
 (C.10)

$$-x_{tj} + x_{tj} - x_{p_{tij}} \le 0 \quad \forall t, i, j > i$$
(C.11)

$$y_{ti} - y_{tj} - y_{-} p_{tij} \le 0 \quad \forall t, i, j > i$$
 (C.12)

$$-y_{ij} + y_{ij} - y_{j} p_{iij} \le 0 \quad \forall t, i, j > i$$
(C.13)

$$l_{ii} \leq \begin{cases} Lng_{ii}, \text{ If } h_{ii} = 1\\ Sh_{ii}, \text{ Otherwise} \end{cases} \quad \forall t, i$$
(C.14)

$$-l_{ii} \leq \begin{cases} -Lng_{ii}, \text{ If } h_{ii} = 1\\ -Sh_{ii}, \text{ Otherwise} \end{cases} \quad \forall t, i$$
(C.15)

$$w_{ti} \leq \begin{cases} Lng_{ti}, \text{ If } h_{ti} = 0\\ Sh_{ti}, \text{ Otherwise} \end{cases} \quad \forall t, i$$
(C.16)

$$-w_{ti} \leq \begin{cases} -Lng_{ti}, \text{ If } h_{ti} = 0\\ -Sh_{ti}, \text{ Otherwise} \end{cases} \quad \forall t, i$$
(C.17)

$$x_{ti} - x_{t-1,i} \le \begin{cases} 0, \text{ If } r_{ti} = 0\\ M, \text{Otherwise} \end{cases} \quad \forall i, t > 1$$
(C.18)

$$-x_{ti} + x_{t-1,i} \le \begin{cases} 0, \text{ If } r_{ti} = 0\\ M, \text{Otherwise} \end{cases} \quad \forall i, t > 1$$
(C.19)

$$y_{ti} - y_{t-1,i} \le \begin{cases} 0, \text{ If } r_{ti} = 0\\ M, \text{Otherwise} \end{cases} \quad \forall i, t > 1$$
(C.20)

$$-y_{ti} + y_{t-1,i} \le \begin{cases} 0, \text{ If } r_{ti} = 0\\ M, \text{Otherwise} \end{cases} \quad \forall i, t > 1$$
(C.21)

$$w_{ti} - w_{t-1,i} \le \begin{cases} 0, \text{ If } r_{ti} = 0\\ M, \text{Otherwise} \end{cases} \quad \forall i, t > 1$$
(C.22)

$$-w_{ti} + w_{t-1,i} \le \begin{cases} 0, \text{ If } r_{ti} = 0\\ M, \text{ Otherwise} \end{cases} \quad \forall i, t > 1$$
(C.23)