

2005

Tabu search heuristics for the dynamic facility layout problem

Wen-Hsing Liu
West Virginia University

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

Recommended Citation

Liu, Wen-Hsing, "Tabu search heuristics for the dynamic facility layout problem" (2005). *Graduate Theses, Dissertations, and Problem Reports*. 4167.
<https://researchrepository.wvu.edu/etd/4167>

This Thesis 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 Thesis 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 Thesis 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.

**Tabu Search Heuristics for the
Dynamic Facility Layout Problem**

Wen-Hsing Liu

**Thesis submitted to the
College of Engineering and Mineral Resources
at West Virginia University
in partial fulfillment of the requirements
for the degree of**

**Master of Science
in
Industrial Engineering**

**Alan R. McKendall Jr., Ph.D., Chair
Majid Jaraiedi, Ph.D.
Wafik H. Iskander, Ph.D.**

Department of Industrial and Management Systems Engineering

**Morgantown, West Virginia
2005**

Keywords: Dynamic Facility Layout Problem, Tabu Search, Meta-heuristics

ABSTRACT

Tabu Search Heuristics for the Dynamic Facility Layout Problem

Wen-Hsing Liu

The facility layout dramatically influences the efficiency of material handling within a manufacturing system. In order to ensure optimal performance within a manufacturing system, the facility layout should reflect changes throughout time. However, the static facility layout problem with constant material flows between departments may not be a realistic scenario because a manufacturing facility is a dynamic system that constantly evolves. In other words, product demand constantly changes over time. As a result, the dynamic facility layout problem (DFLP) considers these changes and is defined as the problem of assigning departments to locations during a multi-period planning horizon such that the sum of the material handling and rearrangement costs is minimized. In this research, tabu search heuristics and a probabilistic tabu search heuristic are developed to solve the DFLP. The proposed tabu search heuristics are a simple tabu search heuristic, a tabu search heuristic with diversification and intensification strategies, and a probabilistic tabu search heuristic. Two data sets taken from the literature are used to test the performances of the proposed heuristics. Computational experiments show that the proposed heuristics out-performed the heuristics presented in the literature with respect to solution quality and computational time.

ACKNOWLEDGEMENTS

I appreciate my advisor, Dr. Alan R. McKendall Jr., for his valuable guidance and continuous support. He introduced me into the new field of Operations Research and inspired me to go further in the academic field. He also put forth a great deal of effort in helping me finish my research, and I could not have completed this thesis without his assistance.

I also would like to thank my committee members, Dr. Wafik Iskander and Dr. Majid Jaraiedi, for their suggestions on this research. They have been valuable to my studying at West Virginia University.

I am grateful to my parents for giving me the opportunity to pursue my graduate education in the United States, and to my sisters, Chin-Yen and Chiung-Fen, who gave me a lot of assistance and encouragement throughout this time.

I also would like to thank Mr. Jin Shang for providing me help in this research. In addition, a lot of thanks go to my friends, Yi-Ting and Vivian, for their emotional support. I am fortunate to have you in my life.

After the long effort, I have finally completed my thesis. The time I have spent in Morgantown has been an extraordinary and delightfully unexpected experience which was full of happiness, sadness, surprises and stories. This is a special journey that I have had in my life and I will never forget it.

TABLE OF CONTENTS

ABSTRACT	ii
ACKNOWLEDGEMENTS	iii
TABLE OF CONTENTS	iv
LIST OF TABLES	vi
LIST OF FIGURES	vii
CHAPTER 1 INTRODUCTION	1
1.1 The Importance of the Facility Layout Problem.....	1
1.2 The Facility Layout Problem	2
1.3 The Static Facility Layout Problem	4
1.4 The Dynamic Facility Layout Problem.....	6
1.5 Objectives of the Thesis.....	8
1.6 Organization of the Thesis	8
CHAPTER 2 LITERATURE REVIEW	10
2.1 Introduction.....	10
2.2 The Static Facility Layout Problem	10
2.2.1 Introduction.....	10
2.2.2 Exact Methods	11
2.2.3 Meta-Heuristics.....	11
2.2.4 Tabu Search Heuristics for the SFLP.....	12
2.3 Dynamic Facility Layout Problem.....	15
2.3.1 Introduction.....	15
2.3.2 Exact Methods	16
2.3.3 Heuristic Methods.....	17
CHAPTER 3 PROBLEM DEFINITION	24
3.1 Problem Definition.....	24
3.2 Problem Assumptions	24
3.3 Mathematical Formulation.....	25
CHAPTER 4 METHODOLOGY	28
4.1 Introduction.....	28

4.2	Basic Tabu Search Heuristic for the DFLP.....	28
4.2.1	Basic Tabu Search Heuristic.....	28
4.2.2	Solution Representation for the DFLP.....	29
4.2.3	Determining the total Cost of the DFLP Solution	30
4.2.4	Neighborhood structure	33
4.2.5	Calculating the Change in Total Cost.....	35
4.2.6	Components of the Basic Tabu Search Heuristic for the DFLP.....	38
4.2.6.1	Tabu list	38
4.2.6.2	Aspiration criterion.....	39
4.2.6.3	Stopping criterion.....	39
4.2.7	Pseudo-code for the Basic TS Heuristic for the DFLP.....	40
4.3	TS Heuristic for the DFLP with Diversification/Intensification Strategies.....	41
4.3.1	Diversification Strategies.....	41
4.3.1.1	Frequency-based Memory	41
4.3.1.2	Dynamic Tabu List Size.....	45
4.3.2	Intensification Strategy	47
4.3.3	Pseudo-code for TS Heuristic with Diversification/Intensification Strategies	49
4.4	The Probabilistic Tabu Search Heuristic for the DFLP	51
CHAPTER 5 COMPUTATIONAL RESULTS.....		54
5.1	Introduction.....	54
5.2	Setting Parameters	55
5.3	Results.....	65
5.3.1	Data Set from Lacksonen and Enscore (1993)	65
5.3.2	Data Set from Balakrishnan and Cheng (2000).....	72
CHAPTER 6 CONCLUSION		81
6.1	Summary of Research.....	81
6.2	Recommendations for Future Research.....	82
REFERENCES.....		83

LIST OF TABLES

Table 4.1 Distances between locations.....	31
Table 4.2 Flows between departments.....	32
Table 4.3 The list of candidate moves.....	34
Table 4.4 The list of candidate moves in iteration 2.....	44
Table 4.5 Dynamic tabu list size.....	46
Table 4.6 The accumulated probability table.....	53
Table 5.1 The total number of initial solutions for data set 1.....	57
Table 5.2 The total number of initial solutions for data set 2.....	57
Table 5.3 Minimum, average, and maximum values for data set 1.....	59
Table 5.4 Minimum, average, and maximum values for data set 2.....	60
Table 5.5 The computational time for data set 1.....	62
Table 5.6 The computational time for data set 2.....	62
Table 5.7 Parameter settings for data set 1.....	64
Table 5.8 Parameter settings for data set 2.....	64
Table 5.9 Solution results for TS heuristics for data set 1.....	67
Table 5.10 Solution results for the PTS heuristic for data set 1.....	68
Table 5.11 Solution results for problems with $N = 6$ in data set 1.....	69
Table 5.12 Solution results for problems with $N = 12$ in data set 1.....	70
Table 5.13 Solution results for problems with $N = 20$ in data set 1.....	71
Table 5.14 Solution results for problems with $N = 30$ in data set 1.....	71
Table 5.15 Solution results for TS heuristics for data set 2.....	74
Table 5.16 Solution results for the PTS heuristic for data set 2.....	75
Table 5.17 Solution results for problems with $N = 6$ in data set 2.....	77
Table 5.18 Solution results for problems with $N = 15$ in data set 2.....	78
Table 5.19 Solution results for problems with $N = 30$ in data set 2.....	79

LIST OF FIGURES

Figure 1.1 Layout plan for the DFLP with 4 departments and 3 periods.....	7
Figure 3.1 Layout configuration.....	24
Figure 4.1 Solution for a DFLP with four departments and two periods.....	30
Figure 4.2 Layout configuration of four departments.....	31
Figure 4.3 the solution before and after the exchange.....	37
Figure 4.4 The updated tabu list for period 1 ($tabu[1][i][k]$) at the end of iteration 1...	39
Figure 4.5 The tabu list for period 1 ($tabu[1][i][k]$) with recency and frequency-based memory structure at the end of iteration 1.....	42
Figure 4.6 The tabu list at the beginning of iteration 2.....	44
Figure 4.7 The updated tabu list at the end of iteration 2.....	45
Figure 4.8 The dynamic tabu list in at the end of iteration 2.....	47

CHAPTER 1

INTRODUCTION

1.1 The Importance of the Facility Layout Problem

Facility layout dramatically influences the efficiency of material handling within a manufacturing system. Therefore, the facility layout is of great concern for manufacturers. An efficient facility layout will improve profit and productivity. Moreover, it has been estimated that materials handling cost is between 20 to 50% of the total operating cost, and effective facility layout planning can reduce the material handling costs by 10 to 30% (Tompkins *et al.*, 2003, p. 10).

Since customer demand is constantly changing, the material handling paths and layout of machines (or departments) are varied constantly. In other words, in order to ensure optimal performance of a facility, the layout should reflect changes to the system that may occur over time. Therefore, the facility layout problem exists when either new plants are built or old plants are modified. Francis *et al.* (1992, p. 32-33) proposed some reasons that may cause the modification of the layout of a facility:

- a. Change of the product design.
- b. The addition or deletion of a product from the product line.
- c. Significant increase or decrease in the demand for a product.
- d. Changes on the design of the process.
- e. The replacement of equipment.
- f. The adoption of new safety standards.
- g. Bottlenecks in production.
- h. Unexplainable delays and idle time.

- i. Excessive temporary storage space.

Therefore, the facility layout problem often occurs and exists for many different reasons. As a result, the facility layout may need to be modified constantly. The decision of the facility layout is made at the strategic level and has a long-lasting effect on the manufacturing system. Once the decision is made, changing the layout of the facility can be very costly. Some of the costs associated with the re-layout of a facility are cost of rearranging the machines, cost of purchasing or leasing equipment for rearranging the machines/department, and the cost associated with the loss of production. Therefore, the importance of the facility layout problem is obvious.

1.2 The Facility Layout Problem

The facility layout problem is to find the most efficient arrangement of departments within a facility. The most commonly used criterion to determine the efficiency of facility layouts is material handling cost minimization. Besides this, a number of objectives are considered important in evaluating a facility layout, and they are as follows, as defined in Francis *et al.* (1992, p. 33-34).

- a. Minimize investment in equipment.
- b. Minimize overall production time.
- c. Utilize the existing space effectively.
- d. Facilitate the manufacturing process and organizational structure.
- e. Maintain flexibility of arrangement and operation.
- f. Minimize variation in types of material handling equipment.
- g. Provide for employee convenience, safety, and comfortable working environment.

The objective of the facility layout problem in this research is to minimize the material handling cost which is based on the material flows between departments and the distances between their locations. However, the objective can be either a distance-based objective (Bozer *et al.*, 1994) or an adjacency-based objective (Heragu and Kusiak, 1991). Objectives (a)-(d) given above are distance-based objectives and objectives (e)-(g) are adjacency-based objectives (McKendall *et al.*, 1999). A distance-based objective is based on the material flows between departments and the distances between their locations, and an adjacency-based objective is based on the ratings of the closeness of the departments. The closeness rating is a value which indicates the preference between adjacent departments. The objective is achieved by maximizing the adjacency score between preferred departments. In retrospect, the objective of the facility layout problem in this research is to minimize the material handling costs which is a distance-based objective.

Facility layout problems can be further classified according to the type of material flows. The materials that flow between departments can be either deterministic or stochastic. Deterministic flow data are fixed and known with certainty. In contrast, when the material flows are not known with certainty, the flow is a random variable and may be represented as a probability distribution. This type of material flow data is defined to be stochastic. Kouvelis *et al.* (1992) presented the facility layout problem with stochastic flow data. However, the flow data for the facility layout problem presented in this research are deterministic.

Furthermore, the most commonly used distance measures for the facility layout problem are rectilinear and Euclidean. The rectilinear distance between two points (x_1, y_1) and (x_2, y_2) is defined as $|x_1 - x_2| + |y_1 - y_2|$, and the Euclidean distance between the two

points is defined as $\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$. Other distance measures can also be used to determine the distances between two departments. Nevertheless, in this research the rectilinear distance measure is used to determine the distance between two departments.

The output of the facility layout problem can be represented using a block layout. A block layout specifies the relative location and size of each department within a facility and can be represented in either a discrete or continuous fashion. A discrete block layout representation uses a collection of grids to represent the locations of departments, and a continuous representation uses the centroids, areas, and the widths (or lengths) of the departments to specify the exact locations of the departments. The layout representation in this research uses the discrete block layout.

The facility layout problem can be either static or dynamic. The static facility layout problem (SFLP) considers the layout for a single period. In other words, the material flow data are fixed. However, the dynamic facility layout problem (DFLP) considers the dynamic nature of the facility layout problem (i.e., the material flow data change over time). More specifically, the DFLP solves the facility layout problem for several periods in a planning horizon.

1.3 The Static Facility Layout Problem

The static facility layout problem (SFLP), considering the discrete representation of the layout, is to assign a set of n departments to a set of n locations within a facility for a given time horizon with respect to minimizing material handling cost. This type of SFLP is formulated as a quadratic assignment problem (QAP). The following QAP formulation for the SFLP is adopted from Koopmans and Beckman (1957).

$$\text{Minimize } \sum_{i=1}^N \sum_{j=1}^N \sum_{k=1}^N \sum_{l=1}^N F_{ik} D_{jl} X_{ij} X_{kl} \quad (1)$$

Subject to

$$\sum_{j=1}^N X_{ij} = 1, \quad i = 1, \dots, N, \quad (2)$$

$$\sum_{i=1}^N X_{ij} = 1, \quad j = 1, \dots, N, \quad (3)$$

$$X_{ij} = \{0, 1\}, \quad i, j = 1, \dots, N, \quad (4)$$

where

i, k = Departments in the layout.

j, l = Locations in the layout.

N = Number of departments and locations.

D_{jl} = Distance between location j and l .

F_{ik} = Flow cost between departments i and k .

$X_{ij} = 1$, if department i is assigned to location j , and

0, otherwise.

The objective function (1) minimizes the total material handling cost. Constraint set (2) ensures that every department is assigned to one location, and constraint set (3) ensures that every location is assigned to one department. Last, constraint set (4) gives the restrictions on the decision variables.

When the facility layout problem has only a single period or is under the static environment, the flow of materials between departments is constant. However, this may not be a realistic scenario due to the fact that the flows of materials between departments are constantly changing due to the changes described before (e.g., change in the product

design, the addition or deletion of a product from the product line, etc.). Therefore, the purpose of this research is to examine the dynamic facility layout problem.

1.4 The Dynamic Facility Layout Problem

Because of the dynamic environment of the manufacturing system, some factors such as customer demand or change in production equipment, etc., can cause fluctuation in the material flow between departments. The changes in material flow between departments may result in the increase of the material handling cost in the existing layout. Thus, in order to maintain the efficiency of material flow, it may be necessary to modify the facility layout in different periods which results in the DFLP. The dynamic facility layout problem (DFLP) is the problem of assigning departments to locations during a multi-period planning horizon such that the sum of material handling and rearrangement cost is minimized. The costs considered in the DFLP are the material handling and the rearrangement costs. The material handling cost is the sum of the product of flow costs between pairs of departments and the distances between their locations. The rearrangement cost is the relocation cost of the departments (e.g. the fixed cost for installing a department, the transportation cost for the facilities, etc.), and it occurs when the locations of the departments are changed in consecutive periods.

The rearrangement cost is the cost of rearranging departments. If rearrangement costs are much less than the material handling cost, solving the dynamic facility layout problem is not necessary. In other words, when rearrangement costs are negligible, the DFLP could be solved by solving the SFLP for each period. If rearrangement costs are relatively large, the problem is also solved as a series of SFLP, and the layout which gives the minimum cost is used for each period. Often times, differences between the

material handling and rearrangement costs are small enough such that the DFLP cannot be solved as a series of SFLPs. Therefore, there is a trade-off between minimizing the material handling and rearrangement costs.

An example of a series of layouts for a DFLP with 4 departments and 3 periods is given in Figure 1.1. In period 1 ($t = 1$), departments 1, 2, 3, and 4 are assigned to locations 3, 1, 2, and 4, respectively. Based on the arrangement of departments (distances between pairs of departments) and the material flows between pairs of department, the material handling cost may be obtained for the first period. Similarly, the material handling costs may be obtained for periods 2 and 3. Since departments 2 and 4 relocate in period 2, there is rearrangement cost associated with this relocation. In contrast, there is no rearrangement cost in period 3, since there is no rearrangement of departments.

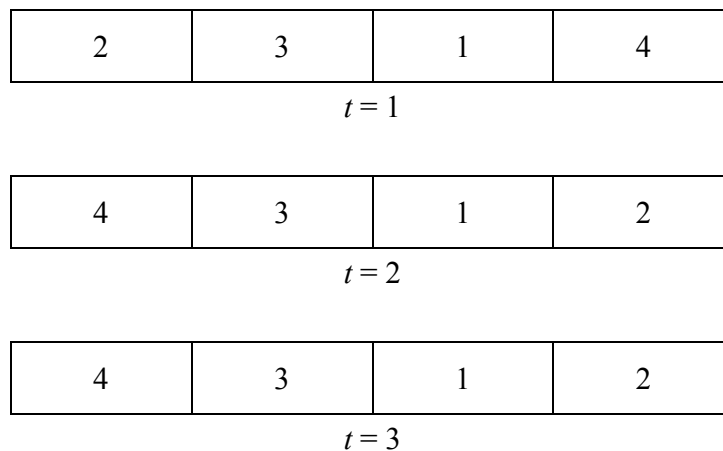


Figure 1.1 Layout plan for a DFLP with 4 departments and 3 periods.

In a DFLP, it is extremely hard to obtain the optimal solution as the number of departments and periods increase. For example, consider a DFLP with four departments and three periods ($N = 4, T = 3$). There are $4! = 24$ possible layouts for each period. Thus, the total number of possible layouts (solutions) is $(4!)^3 = 13,824$. Also, consider the

example of a DFLP with six departments and four time periods ($N = 6, T = 4$). There are $6! = 720$ possible layouts for each period. Hence, the total number of possible layouts is $(6!)^4 = 26.87 \times 10^{10}$. As a result, these two examples show that the number of layouts increases dramatically with a slight increase in the number of departments and periods. Therefore, it is very hard to obtain optimal solutions for even small sized DFLP instances in reasonable time using exact methods. That is why heuristics are often used to obtain “good” solutions to the DFLP in a reasonable time.

1.5 Objectives of the Thesis

The objectives of this research are given as follows:

1. To develop a simple tabu search heuristic for the DFLP.
2. To develop a tabu search heuristic with diversification and intensification strategies for the DFLP.
3. To develop a probabilistic tabu search heuristic for the DFLP.
4. To test the performance of the tabu search heuristics by solving test problems from two data sets taken from the literature.

1.6 Organization of the Thesis

In Chapter 2, a literature review for the SFLP and DFLP is given. In Chapter 3, the problem definition, assumptions and mathematical formulation for DFLP are presented. In Chapter 4, a simple tabu search heuristic, a tabu search heuristic with diversification and intensification strategies, and a probabilistic tabu search heuristic are described for solving the DFLP. In Chapter 5, the computational experiments are conducted to test the performances of the heuristics, and the parameter settings and results generated from the

proposed tabu search heuristics are presented. Finally, the conclusions and recommendations for future research are given in Chapter 6.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

A number of papers have been published on solving the static and dynamic facility layout problems. Most of the research is devoted to the SFLP or the QAP. Since it is very difficult to find optimal solutions in reasonable time for large size problems, most of the research focuses on heuristic approaches. This chapter reviews exact methods and meta-heuristics, including tabu search heuristics, for the SFLP. Also, exact methods and heuristic methods are reviewed for the DFLP.

2.2 The Static Facility Layout Problem

2.2.1 Introduction

Koopmans and Beckmann (1957) first presented the quadratic assignment problem (QAP). The authors first introduced the quadratic assignment problem which is to assign plants to locations to maximize total net revenue. Armour and Buffa (1963) presented a new algorithm, which determined how an initial solution to the QAP can be improved using a pairwise exchange heuristic. Buffa *et al.* (1964) improved this heuristic and called the improved heuristic *CRAFT* (Computerized Relative Allocation of Facilities Technique). The method starts with an initial layout and evaluates all pairs of departments in the neighborhood of the initial solution. The corresponding layout for the pair of departments with the best objective function value (i.e., most reduction in cost) is selected for exchange. Then the layout is updated according to the best exchange, and this

layout becomes the starting layout at the next iteration. These steps are repeated until no better solution is found.

2.2.2 Exact Methods

Some exact methods used for solving the SFLP are branch and bound algorithms presented by Gilmore (1962), Lawler (1963) and Kaku and Thompson (1986). Also, Bazaraa and Sherali (1980) and Burkard and Bonninger (1983) developed cutting plane algorithms for solving the SFLP.

2.2.3 Meta-Heuristics

Meta-heuristics such as simulated annealing (SA), tabu search (TS), genetic algorithms (GA), and ant systems are used for solving the SFLP. Burkard and Rendl (1984) were the first to apply the simulated annealing (SA) algorithm to the SFLP, and Wilhelm and Ward (1987) presented a SA algorithm for the SFLP. Also, Heragu and Alfa (1992) used a hybrid SA algorithm to solve SFLP, and the results show that the hybrid SA algorithm outperformed Wilhelm and Ward (1987) SA algorithm. In addition, Fleurent and Ferland (1994), Tate and Smith (1995), Suresh *et al.* (1995), and Ahuja *et al.* (2000) used genetic algorithms (GA) for solving the SFLP. Gambardella *et al.* (1999) presented hybrid ant systems (HAS) to solve the QAP. Since the proposed heuristics are tabu search heuristics, the papers that have applied tabu search heuristics to solve SFLP are reviewed in the following section.

2.2.4 Tabu Search Heuristics for the SFLP

Skorin-Kapov (1990) was the first to apply the tabu search heuristic to solve the SFLP. The author described the problem as a quadratic assignment problem (QAP) and proposed the tabu-navigation algorithm to solve the QAP. Aspiration criterion and tabu list are used in the proposed heuristic. The parameters used are the length of the tabu list denoted as *tabu_size* and the maximum number of iterations denoted as *max_iter*. After performing *max_iter* iterations, one of the following steps is selected: (i) Restart from the solution given by the construction algorithm with new values for *tabu_size* and *max_iter*; (ii) Restart from the best solution obtained so far with new values for *tabu_size* and/or *max_iter*; (iii) Invoke long term memory: Restart the procedure from the beginning of the construction phase, penalizing the moves performed so far; (iv) Stop the procedure. The long term memory is used to record moves that occurred in the past in order to penalize them in the construction phase. Computational experiments with different parameter values and different strategies have been performed for test problems taken from the literature and some randomly generated test problems with the number of departments (n) varied between 42 and 90. The computational results show that tabu search heuristic outperformed simulated annealing algorithm with respect to solution quality.

Skorin-Kapov (1994) modified the tabu search heuristic from Skorin-Kapov (1990) to solve the QAP. The differences between the modified tabu search and the tabu search in Skorin-Kapov (1990) are to redefine the evaluation function and its domain, new intensification and diversification strategies, and to change the composition of the tabu list. The computational experiment is conducted by the test problems with the number of departments (n) varied between 42 and 90 taken from Skorin-Kapov (1990). This tabu search obtained better results for all the test problems.

Chiang and Kouvelis (1996) presented a new implementation of tabu search heuristic to solve the QAP. The tabu search heuristic includes a recency-based and frequency-based memory structure, as well as diversification and intensification strategies. A two dimensional array called tabu list is used to keep track of tabu status and frequency of moves. Therefore, the tabu list employs the recency-based and frequency-based memory structure. The diversification strategy includes dynamic tabu list size and a penalty function for nonimproving moves. The dynamic tabu list size strategy used dynamic tabu length to diversify the search. The dynamic tabu length varies according to the percentage improvement from the total cost of the last move and the current move. The diversification strategies also use a penalty function to penalize nonimproving moves that have been visited. The intensification strategy used the method of fixing departments and freeing departments to intensify the search region. The pairs of departments are fixed if it reduces more than a certain percentage from the total cost of the best found solution so far (i.e., the location of the pair of departments are not allowed to change until they are free). The pairs of departments are freed when an exchange of the fixed department and other free department yields a percentage improvement better than the fixed pairs of departments. The proposed tabu search heuristic is tested using the test problems taken from the literature and compared with the tabu-navigation in Skorin-Kapov (1990), extension of tabu-navigation in Skorin-Kapov (1994), and Taillard (1991). Computational results show that the proposed tabu search heuristic outperformed the other algorithms.

Chiang and Chiang (1998) presented a tabu search heuristic, a probabilistic tabu search heuristic, a simulated annealing heuristic, and a hybrid tabu search heuristic to solve the QAP. The proposed tabu search heuristic started with a randomly generated

initial solution. Tabu list and aspiration criteria were used in the tabu search heuristic to escape from a local optimal solution. The tabu search heuristic includes a frequency-based memory structure, avoidance list, dynamic tabu size, and a penalty function. The frequency-based memory structure was used to record the number of times a move was selected for exchange, and the avoidance list recorded the worst candidate move. This move was not allowed in the candidate move list for a certain number of iterations. Also, a dynamic tabu size was implemented with dynamic tabu length varied from a lower bound to an upper bound. In addition, a penalty function used the information from the frequency-based memory structure to diversify the search process. The probabilistic tabu search (PTS) heuristic is a modification of the tabu search heuristic. The difference between the tabu search and PTS heuristic is how the move is selected. The PTS heuristic randomly selected a move from the candidate list of moves for exchange. The simulated annealing heuristic is a memoryless procedure in which the search history is not recorded. The hybrid tabu search approach combines the tabu search and simulated annealing. The proposed heuristics are tested using the test problems taken from Nugent *et al.* (1968), Golany and Rosenblatt (1989), and Skorin-Kapov (1990). Computational results show that the hybrid tabu search approach which combines the advantages of the tabu search and simulated annealing heuristics outperformed the other heuristics.

Lim *et al.* (2004) used a probabilistic tabu search approach to solve the crane scheduling problem. The initial solution is generated from either a greedy method or a random crane-job assignment. The similar ideas of short-term and long-term memory strategies of Chiang and Chiang (1998) are implemented. The strategy for a probabilistic move selection is also used in the tabu search approach.

Some survey papers are presented for SFLP. Kusiak and Heragu (1987) presented a survey paper for the facility layout problem. Twelve heuristic algorithms are compared on the basis of their performance. The author attempted to include almost all optimal and suboptimal algorithms which solve the facility layout problem. The optimal algorithms are branch and bound algorithms and cutting plane algorithms. The suboptimal algorithms include construction algorithms, improvement algorithms, hybrid algorithms, and graph-theoretic algorithms. In addition, Meller and Gau (1996) provided a review of the facility layout problem. This review discussed the extensions of the facility layout problem including dynamic layout, stochastic layout, and multiple objective criteria.

2.3 Dynamic Facility Layout Problem

2.3.1 Introduction

Rosenblatt (1986) first presented the dynamic nature of the plant layout problem and defined the DFLP. Afterwards, Conway and Venkataramanan (1994) presented a genetic algorithm to solve the DFLP. The only tabu search heuristic presented for the DFLP was presented by Kaku and Mazzola (1997), and Baykasoglu and Gindy (2001) first applied the simulated annealing algorithm to the DFLP. In addition, Lacksonen and Ensore (1993), and Balakrishnan and Cheng (2000) presented heuristics for the DFLP and also proposed test problems for the DFLP. Both data sets are used to test the performances of the proposed heuristics in this research. In the following section, a literature review for the DFLP is given.

2.3.2 Exact Methods

Rosenblatt (1986) first presented the dynamic nature of the plant layout problem and solved the DFLP by using a dynamic programming approach. A dynamic programming formulation is presented and solved by both exact and heuristic methods. A dynamic programming approach is applied by using the period as stages and the specific layout as states. A good upper bound on the objective function value and the best solution for the SFLP in each period are obtained to reduce the number of possible layouts to be evaluated and the possible solutions still provided the optimal solution. A recursive formulation is developed to consider the total cost of the possible layouts in each period. The global optimal solution is the combination of layouts with the minimum total cost.

Since the computational time increases dramatically with the number of states in a dynamic programming problem, a heuristic procedure is practical for large size problems. Rosenblatt (1986) also presented two approaches in the paper. The first approach is to solve the SFLP optimally for all periods. The set of layouts considered in each period is just the optimal solution for each SFLP. Thus, the maximum number of layouts (states) in each period (stage) is the number of periods (n). The heuristic procedure is similar to the heuristic presented by Ballou (1968) for the warehouse location problem. The second approach is to generate the set of layouts (solutions) for each period by using computerized approaches, such as CRAFT (Buffa *et al.*, 1964), COFAD (Tompkins and Reed, 1976), or randomly generating algorithms. An example of a DFLP with six departments and five periods is considered and a set of 30 test problems are solved by Ballou's method and the randomly generated layout approach. The results were compared with the optimal solution and the average errors for both methods are small.

Lacksonen and Enscore (1993) modified five algorithms for the SFLP to solve the DFLP. Two of them are exact methods, which are the dynamic programming approach and branch and bound. A 0/1 formulation extended from the QAP model is presented. The dynamic programming formulation is based on Rosenblatt's (1986) method. The pairwise exchange algorithm is applied to the flow data for each period to generate possible layouts (states) for each period. Furthermore, hybrid states are obtained by exchanging the locations of departments between the best layouts of consecutive periods. The branch and bound algorithm taken from Pardalos and Crouse (1989) used a cutting plane algorithm to obtain an upper bound on the total cost of the solution for the QAP. Two modifications for the branch and bound are made to solve the DFLP. In the first one, the departments are only permitted to be assigned to the proper period. In the second one, the lower bound calculation is revised by adding the estimated cost of all periods which do not have any assignments yet made. A series of test problems with 6, 12, 20, and 30 departments each with 3 and 5 periods was developed to determine the effectiveness of the five algorithms.

2.3.3 Heuristic Methods

Lacksonen and Enscore (1993) modified algorithms of for the SFLP to solve the DFLP. Three of them are heuristics, which are CRAFT, cutting planes, and cut trees. CRAFT starts with an initial solution and exchanges pairs of departments to minimize the total cost. The modification of CRAFT is to consider pairs for exchange for all periods. The cutting planes taken from Burkard and Bonniger (1983) for QAP are cutting planes with an exchange routine. The routine starts with a random solution and an assignment routine finds the estimated best solution such that all departments moving to new

locations at each iteration. Since the optimal solution may be eliminated by this “cut”, each iteration ends with an exchange routine. To solve the DFLP, the cutting plane portion assumes the location for each department for each period is the same. Then, only the exchange routine is used to consider rearrangements. The cut trees (Gomory and Hu, 1961) are the graphical layout techniques with spanning trees. To apply on the DFLP, each department of each period is represented by a node. The arcs are added with the rearrangement cost between the nodes representing the same department in consecutive periods. A series of test problems with 6, 12, 20, and 30 departments each with 3 and 5 periods was developed to determine the effectiveness of the five algorithms. Computational results show that the cutting plane algorithm performs better for this set of test problems.

Urban (1993) presented a steepest-descent pairwise-interchange procedure for the DFLP. The steepest-descent pairwise interchange procedure solves the DFLP using the material handling cost with forecast windows and rearrangement cost. Only the initial layout of the first period is given in the heuristic. When the length of the forecast window is equal to 1, the layout for period 1 is developed by using the material flow data for period 1. The layout for period 2 is developed by using the flow data for period 2, and so on. The layout obtained from period 1 is used as the initial layout for period 2, etc. When the length of the forecast window is equal to 2, the material flow data for periods 1 and 2 are used to determine the layout for period 1. The material flow data for periods 2 and 3 are used to determine the layout for period 2, and so on. Fifty-two test problems are generated to evaluate the accuracy of the heuristics by comparing with the results of Ballou’s (1968) heuristic. Further analysis was conducted to test the accuracy of the heuristic of large size problems by using data from Nugent *et al.* (1968). The test

problems are problems with 6, 8, 12, 15, 20 and 30 departments each with 4, 8, 12, 16, and 20 periods. The results show that the heuristic proposed performed as well as Ballou's (1968) heuristic.

Conway and Venkataramanan (1994) presented a genetic algorithm to solve a DFLP. This problem is called the constrained DFLP which decides the facility layout restricted by a budget for total rearrangement costs over entire finite horizon such that the total costs of layout rearrangements and material flow between departments during the planning horizon is minimized. This is a combinatorial problem and it is solved by a genetic search algorithm. Computational results are presented for two sample problems from the literature.

Kaku and Mazzola (1997) presented a tabu search heuristic for the DFLP. First, a basic tabu search procedure for DFLP is presented. The basic tabu search procedure for the DFLP includes four steps: Step 1: Initialize: Obtain the initial solution and initialize the parameters and iteration counter. Set the initial solution as the current solution. Step 2: Neighborhood search: Evaluate each neighbor (or move) of the current solution. The move which is non-tabu or overrides the tabu restriction with the best objective function value is selected as the best admissible move. The corresponding solution of this move becomes the current solution. Step 3: Update the tabu list, current solution, and best found solution so far (if necessary). Step 4: Stopping rules: Increase iteration counter by 1. If the stopping criteria are met, terminate the procedure. Otherwise, go to Step 2. Two stopping criteria are used in the DFLP tabu search heuristic. The first criterion is the maximum number of iterations ($ITER_{max}$), and the second criterion is the maximum number of consecutive iterations without improvement ($NOLMP_{max}$). Also, diversification and intensification strategies are used in the DFLP tabu search heuristic.

The diversification strategy starts the DFLP tabu search heuristic with a specified number of initial solutions using a construction algorithm for QAP obtained from Kaku *et al.* (1991). The intensification strategy is to adjust the tabu length (L_{TABU}) during the search process to allow a more intensive search. The tabu length is modified when both of the following conditions are satisfied: the current iteration number is more than $2/3 * ITER_{max}$ and the number of nonimproving iterations is more than $NOLMP_{max}$. Then, the tabu length is defined as $L_{TABU}/2$. Therefore, the tabu length in the tabu search heuristic is a dynamic tabu length.

The DFLP tabu search heuristic is a two-stage procedure. In stage one, the basic tabu search heuristic with a diversification strategy is applied. And in stage two, the intensification strategy is started and the tabu length (L_{TABU}) is modified. The DFLP tabu search procedure is the same as the basic tabu search procedure in Step 1 and Step 2. Step 3 and Step 4 are modified as follows. Step 3: Update the tabu list, current solution, and best found solution so far (if necessary); and adjust for intensification search. If the current iteration number $> 2/3 * ITER_{max}$ and the number of nonimproving iterations $> NOLMP_{max}$, the tabu length is modified to $L_{TABU}/2$, and set the nonimproving iteration counter to zero. Step 4: Stopping rules: If the current iteration number is more than $ITER_{max}$ or if tabu length = $L_{TABU}/2$ and the number of nonimproving iterations is more than $NOLMP_{max}$, terminate the procedure. Otherwise, go back to Step 2. The DFLP tabu search heuristic was tested for the 32 test problems taken from Lacksonen and Ensore (1993) and compared with the cutting planes in Lacksonen and Ensore (1993) and the heuristic presented in Urban (1993). Computational results show that the DFLP tabu search heuristic generated improved solutions for over one-third of the 32 test problems and matched solution quality on an additional 50 percent of the 32 test problems.

Balakrishnan and Cheng (2000) presented a paper dealing with the dynamic facility layout problem by using an improved genetic algorithm. The nested loop genetic algorithm differs from the existing implementations in three ways: a different crossover operator, using mutation, and using a new generational replacement strategy to help increase population diversity. Computational results are presented for a number of test problems which are randomly generated by a specific setting. The total is 48 test problems with 6, 15 and 30 departments with each 5 and 10 periods. The results show that the improved GA is effective.

Balakrishnan *et al.* (2000) presented an improved dynamic pairwise exchange heuristic for the DFLP. The authors presented an improved dynamic pairwise exchange heuristic which is based on Urban's (1993) heuristic by using the time windows concept. There are two improvements proposed for Urban's (1993) heuristic. The first one is a backward-pass method. Initially Urban's heuristic is used to solve the DFLP. Then a backward pass pairwise exchange is performed on each solution from Urban's (1993) forward pass heuristic. The best solution is selected. The second one is to combine Urban's (1993) heuristic with dynamic programming. Using Urban's heuristic, first solve the DFLP. The result generated is the initial solution for a dynamic programming approach. The layouts obtained from Urban's (1993) heuristic are used as the states in Rosenblatt's dynamic programming procedure. Computational results are presented for test problems with 6, 15, and 30 departments with each 5 and 10 periods. The data set is generated based on the method used in Balakrishnan *et al.* (1992). The proposed heuristics for Urban's procedure and Rosenblatt's method are compared. The computational results show that the proposed method is effective and efficient.

Baykasoglu and Gindy (2001) presented a simulated annealing (SA) algorithm to solve the DFLP. The simulated annealing algorithm is a stochastic search method. It has the capability to get the global optimal by accepting the worse solution with the probability. The acceptance probability is determined by a temperature parameter which decreased during the SA procedure. The cooling schedule is also defined in the paper. Computational experiments are tested by the test problems taken from Balakrishnan and Cheng (2000). The results show that the SA algorithm performs better than the GA.

Balakrishnan *et al.* (2003) presented a hybrid genetic algorithm (GA) to solve the DFLP. The proposed genetic algorithm modified the weakness of the existing GAs in DFLP. Dynamic programming is then used in the crossover operator to create offspring and the CRAFT is used in mutation. Computational experiment is conducted to compare the proposed algorithm with the genetic algorithms in Balakrishnan and Cheng (2000) and the simulated annealing algorithm in Baykasoglu and Gindy (2001). The result shows that the proposed hybrid genetic algorithm provides better quality solutions.

Erel *et al.* (2003) presented a new heuristic to solve the dynamic layout problem. The authors proposed a new heuristic scheme which includes three phases. The first phase is to identify a viable set of layouts and the second phase uses dynamic programming to solve the shortest path problem over the viable set. The third phase is to seek local improvement of the solution obtained in the second phase. Computational results are presented for 48 test problems taken from Balakrishnan and Chang (2000). The proposed heuristic can solve the DFLP reasonably fast and generate the same quality solutions with the other methods.

McKendall and Shang (2005) developed three hybrid ant systems (HASs) to solve the dynamic facility layout problem. The HASs are the modification of the HAS for the

QAP (Gambardella *et al.*, 1999) to solve the DFLP. The first HAS heuristic (HAS I) is a direct application of the HAS-QAP heuristic for the DFLP. The second HAS heuristic (HAS II) is like HAS I, except that a SA heuristic is used as the local search heuristic instead of the random descent pairwise exchange heuristic. The third HAS heuristic (HAS III) is like HAS I, except that the random descent pairwise exchange heuristic has a look-ahead/look-back strategy. The HASs are tested using two data sets taken from Lacksonen and Ensore (1993) and Balakrishnan and Cheng (2000). The results show that the HASs are efficient techniques for solving the DFLP.

McKendall *et al.* (2005) presented two simulated annealing (SA) heuristics to solve the dynamic facility layout problem. The first SA heuristic (i.e. the SA I heuristic) is a straightforward implementation of the SA heuristic. The second SA heuristic (i.e. the SA II heuristic) combines the SA I heuristic with a look-ahead and look-back strategy. The SA heuristics performed well for the data set taken from Balakrishnan and Cheng (2000). The results obtained show that the SA heuristics are effective for the DFLP.

Balakrishnan and Cheng (1998) presented a comprehensive review of the dynamic facility layout problem. They categorize the DFLP into equal size and unequal departments. The type of problems with equal size departments are classified as deterministic material flow and stochastic material flow problems. The algorithms with deterministic flow are dynamic programming approaches, pairwise-interchange heuristics, genetic algorithms, tabu-search, CRAFT, cutting planes, branch and bound, and cut trees. The algorithms with stochastic material flow are branch and bound, Markov process and simulation. The category of problems with unequal size departments used linear programming and mixed integer programming.

CHAPTER 3

PROBLEM DEFINITION

3.1 Problem Definition

The DFLP is the problem of assigning departments to locations during a multi-period planning horizon such that the sum of material handling and rearrangement costs is minimized. The material handling cost is derived from the sum of the product of material flows between departments and the distances between their locations. The rearrangement cost is the relocation cost of a department, and it is incurred when the location of a department in a period is different from the location of the department in the preceding and/or succeeding periods. The solution to the DFLP is represented as a layout plan which is a series of layouts for each period in the planning horizon.

3.2 Problem Assumptions

The assumptions for the DFLP are as follows:

1. The layout configuration is given. See Figure 3.1 for an example of a 2×3 layout configuration with 6 locations.

Location 1	Location 2	Location 3
Location 4	Location 5	Location 6

Figure 3.1 Layout configuration.

2. All departments and locations are of equal size.

3. The flow between departments is deterministic and dynamic.
4. The distances between locations are given. For example, in Figure 3.1 the distance between locations 1 and 3 is 2 distance units and between locations 1 and 6 is 3 distance units.
5. The layout representation is discrete.

3.3 Mathematical Formulation

The mathematical model given below for the DFLP is adopted from Balakrishnan *et al.* (1992).

$$\text{Min } \sum_{t=2}^T \sum_{i=1}^N \sum_{j=1}^N \sum_{l=1}^N A_{ijl} X_{t-1,ij} X_{til} + \sum_{t=1}^T \sum_{i=1}^N \sum_{j=1}^N \sum_{k=1}^N \sum_{l=1}^N C_{ijkl} X_{ij} X_{tkl} \quad (1)$$

Subject to

$$\sum_{j=1}^N X_{ij} = 1, \quad i = 1, \dots, N, \quad t = 1, \dots, T \quad (2)$$

$$\sum_{i=1}^N X_{ij} = 1, \quad j = 1, \dots, N, \quad t = 1, \dots, T \quad (3)$$

$$X_{ij} = \{0, 1\}, \quad i, j = 1, \dots, N, \quad t = 1, \dots, T \quad (4)$$

where

i, k = Departments in the layout.

j, l = Locations in the layout.

t = Time periods.

T = Number of periods in the planning horizon.

N = Number of departments and locations.

A_{ijl} = Cost of rearranging department i from location j to l in period t .

D_{jl} = Distance between locations j and l .

F_{ik} = Flow from department i to department k in period t .

C_{ijkl} = Cost of material flow from department i at location j to department k at location l in period t .

$$= F_{ik} * D_{jl}.$$

X_{ij} = 1, if department i is assigned to location j in period t .

0, otherwise.

The objective function (1) minimizes the sum of the rearrangement and material handling cost during the planning horizon. Constraint set (2) ensures every department is assigned to one location in each period, and constraint set (3) ensures every location is assigned to one department in each period. Last, constraint set (4) gives the restrictions on the decision variables.

In order to linearize the nonlinear binary integer programming model, the nonlinear term in the objective function is linearized by introducing two new binary variables. The linearized objective function (1') is substituted for the objective function (1) and constraint sets (5), (6), (7), (8), and (9) are added to the model.

$$\text{Min } \sum_{t=2}^T \sum_{i=1}^N \sum_{j=1}^N \sum_{l=1}^N A_{ijl} Z_{t-1,ijl} + \sum_{t=1}^T \sum_{i=1}^N \sum_{j=1}^N \sum_{k=1}^N \sum_{l=1}^N C_{ijkl} W_{ijkl} \quad (1')$$

$$X_{ij} + X_{kl} - 1 \leq W_{ijkl} \quad \forall t, i, j, k, l \quad i \neq k, j \neq l \quad (5)$$

$$X_{ij} + X_{kl} \geq 2W_{ijkl} \quad \forall t, i, j, k, l \quad i \neq k, j \neq l \quad (6)$$

$$X_{t-1,ij} + X_{til} - 1 \leq Z_{t-1,tijl} \quad \forall i, j, l \quad t = 2, \dots, T \quad (7)$$

$$X_{t-1,ij} + X_{til} \geq 2Z_{t-1,tijl} \quad \forall i, j, l \quad t = 2, \dots, T \quad (8)$$

$$W_{ijkl} = \{0,1\} \quad Z_{t-1,tijl} = \{0,1\} \quad \forall t, i, j, k, l \quad i \neq k, j \neq l \quad (9)$$

Since the DFLP is a minimal problem and the objective function coefficients are not negative, constraints (6) and (8) are not necessary in this formulation. Therefore, only constraints (5), (7) and (9) are added to the formulation for DFLP. The full linearized DFLP model is given as follows:

$$\text{Min} \sum_{t=2}^T \sum_{i=1}^N \sum_{j=1}^N \sum_{l=1}^N A_{ijl} Z_{t-1,tijl} + \sum_{t=1}^T \sum_{i=1}^N \sum_{j=1}^N \sum_{k=1}^N \sum_{l=1}^N C_{ijkl} W_{ijkl} \quad (1')$$

Subject to

$$\sum_{j=1}^N X_{ij} = 1, \quad i = 1, \dots, N, \quad t = 1, \dots, T \quad (2)$$

$$\sum_{i=1}^N X_{ij} = 1, \quad j = 1, \dots, N, \quad t = 1, \dots, T \quad (3)$$

$$X_{ij} + X_{ikl} - 1 \leq W_{ijkl} \quad \forall t, i, j, k, l \quad i \neq k, j \neq l \quad (5)$$

$$X_{t-1,ij} + X_{til} - 1 \leq Z_{t-1,tijl} \quad \forall i, j, l \quad t = 2, \dots, T \quad (7)$$

$$X_{ij} = \{0,1\}, \quad \forall t, i, j \quad (4)$$

$$W_{ijkl} = \{0,1\} \quad Z_{t-1,tijl} = \{0,1\} \quad \forall t, i, j, k, l \quad i \neq k, j \neq l \quad (9)$$

CHAPTER 4

METHODOLOGY

4.1 Introduction

Tabu search (TS) was introduced by Glover (1986). Some applications of the TS heuristic were successfully used for solving the SFLP and DFLP. Skorin-Kapov (1990) was the first to apply the TS heuristic to the QAP (SFLP). Chiang and Kouvelis (1996) presented a new implementation including dynamic tabu list size, a penalty function, and intensification strategies for the QAP. Chiang and Chiang (1998) presented a tabu search heuristic, a probabilistic TS heuristic, and a hybrid TS heuristic to solve the SFLP. One of the TS heuristics for the DFLP presented in this research is a modification of the TS heuristic presented in Chiang and Kouvelis (1996) for the QAP. The other one is a modification of the probabilistic TS heuristic presented by Chiang and Chiang (1998). Kaku & Mazzola (1997) were the only ones to present a TS heuristic for the DFLP. Therefore, the proposed tabu search heuristics presented in this research are: a simple tabu search heuristic, tabu search heuristic with frequency-based memory as well as diversification and intensification strategies, and a probabilistic tabu search heuristic.

4.2 Basic Tabu Search Heuristic for the DFLP

4.2.1 Basic Tabu Search Heuristic

The basic idea of the tabu search heuristic is to restrict some (most recent) moves to prevent cycling and to accept non-improving moves to escape from a local optimum in search of the global optimum. The tabu search heuristic starts with an initial solution and performs a neighborhood search of the current solution. The local neighborhood search

technique most commonly used in the tabu search heuristic is the steepest descent pairwise exchange heuristic. Some candidate solutions will be generated from the neighborhood search technique by exchanging pairs of the departments. All possible exchanges are considered and defined as the neighborhood of the current solution. Then evaluate each neighbor (or move), and choose the best admissible move. The best admissible move will be defined later. This move is defined as tabu (tabu restricted) for the next TL -iterations where TL is the tabu length, and it is recorded in the tabu list, to avoid cycling back to a local optimum. The admissible move is either a move that is non-tabu or is tabu and has an objective function value better than the best solution found so far. The solution obtained from the best admissible move becomes the current solution and is the starting solution at the next iteration. This procedure is repeated until a stopping criterion is met.

4.2.2 Solution Representation for the DFLP

The solution representation for the DFLP is defined as follows:

$$\pi = (\pi^1, \pi^2, \dots, \pi^T)$$

where

π = solution for the DFLP

π^t = layout in period t .

$\pi^t = (\pi^t(1), \pi^t(2), \dots, \pi^t(N))$

$\pi^t(i)$ = location of department i in period t .

T = number of periods

N = number of departments

For example, the solution of the DFLP with four departments and two periods ($N= 4$ and $T = 2$) given in Figure 4.1 has the following solution representation $\pi = ((3, 1, 2, 4), (3, 4, 2, 1))$.

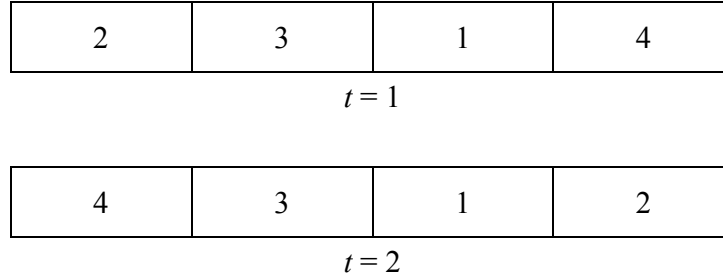


Figure 4.1 Solution for a DFLP with four departments and two periods.

4.2.3 Determining the total Cost of the DFLP Solution

The total cost for the DFLP solution is the sum of material handling and rearrangement costs. The total material handling cost ($MH(\pi)$) is calculated as follows:

$$MH(\pi) = \sum_{t=1}^T \sum_{i=1}^N \sum_{k=i+1}^N W_{ik} * d(\pi^t(i), \pi^t(k))$$

where

F_{ik} = flow from department i to k in period t ,

W_{ik} = weight between department i and k in period t

$$= F_{ik} + F_{ki},$$

$d(j,l)$ = distance between location j and l .

The rearrangement cost occurs when a department is moved (i.e. layout between consecutive periods changes). In other words, the rearrangement cost is generated when the location of a department in the current period is different from that in the preceding or succeeding period. Thus, the rearrangement cost ($RA(\pi)$) is calculated as follows:

$$RA(\pi) = \sum_{t=2}^T \sum_{i \in I_t} A_{ii}$$

where

I_t = all the departments in period t with different locations in period $t-1$ and

A_{ii} = rearrangement cost of department i in period t .

The total cost for the DFLP solution is calculated as follows:

$$TC(\pi) = MH(\pi) + RA(\pi)$$

$$= \sum_{t=1}^T \sum_{i=1}^N \sum_{k=i+1}^N W_{tik} * d(\pi^t(i), \pi^t(k)) + \sum_{t=2}^T \sum_{i \in I_t} A_{ii}$$

Now a solution for a DFLP with 4 departments and 2 periods ($N = 4$ and $T = 2$) is used to illustrate the calculation of the total cost. The problem instance is adopted from Lacksonen and Ensore (1993). Figure 4.2 shows the layout configuration. The distances between locations and the flows between departments for each period are given in Tables 4.1 and 4.2, respectively. The rearrangement cost is 10 for each department.

Location 1	Location 2	Location 3	Location 4
------------	------------	------------	------------

Figure 4.2 Layout configuration of four departments.

		To			
		1	2	3	4
From	1	0	1	2	3
	2	1	0	1	2
	3	2	1	0	1
	4	3	2	1	0

Table 4.1 Distances between locations.

		To			
		1	2	3	4
From	1	0	0	0	0
	2	10	0	4	0
	3	4	4	0	0
	4	0	2	6	0

$t = 1$

		To			
		1	2	3	4
From	1	0	0	0	0
	2	6	0	0	0
	3	2	6	0	0
	4	11	6	5	0

$t = 2$

Table 4.2 Flows between departments.

Given the solution $\pi = ((3, 1, 2, 4), (3, 4, 2, 1))$ where $\pi^1 = (3, 1, 2, 4)$ and $\pi^1(1) = 3$ which indicates that department 1 is assigned to location 3, $\pi^1(2) = 1$, $\pi^1(3) = 2$, and $\pi^1(4) = 4$. Also, $\pi^2 = (3, 4, 2, 1)$ and $\pi^2(1) = 3$, $\pi^2(2) = 4$, $\pi^2(3) = 2$, and $\pi^2(4) = 1$. The weight matrix for each period is obtained as follows.

$$W_{1ik} = F_{1ik} + F_{1ki} = \begin{bmatrix} 0 & 10 & 4 & 0 \\ 10 & 0 & 8 & 2 \\ 4 & 8 & 0 & 6 \\ 0 & 2 & 6 & 0 \end{bmatrix} \quad \text{and} \quad W_{2ik} = F_{2ik} + F_{2ki} = \begin{bmatrix} 0 & 6 & 2 & 11 \\ 6 & 0 & 6 & 6 \\ 2 & 6 & 0 & 5 \\ 11 & 6 & 5 & 0 \end{bmatrix}.$$

Hence, the sum of the material handling costs is calculated as follows.

$$\begin{aligned} MH(\pi) &= MH(\pi^1) + MH(\pi^2) \\ &= W_{112} * d(\pi^1(1), \pi^1(2)) + W_{113} * d(\pi^1(1), \pi^1(3)) + W_{114} * d(\pi^1(1), \pi^1(4)) \\ &\quad + W_{123} * d(\pi^1(2), \pi^1(3)) + W_{124} * d(\pi^1(2), \pi^1(4)) + W_{134} * d(\pi^1(3), \pi^1(4)) \end{aligned}$$

$$\begin{aligned}
& + W_{212} * d(\pi^2(1), \pi^2(2)) + \dots + W_{234} * d(\pi^2(3), \pi^2(4)) \\
& = 10 * d(3, 1) + 4 * d(3, 2) + 0 * d(3, 4) + 8 * d(1, 2) + 2 * d(1, 4) + 6 * d(2, 4) \\
& \quad + 6 * d(3, 4) + \dots + 5 * d(2, 1) \\
& = 115
\end{aligned}$$

Also, rearrangement cost needs to be calculated since the locations of departments 2 and 4 are rearranged in period 2. Therefore, $I_2 = \{2, 4\}$ and

$$RA(\pi) = A_{22} + A_{24} = 10 + 10 = 20.$$

Hence, the total cost of the solution is

$$TC(\pi) = MH(\pi) + RA(\pi) = 115 + 20 = 135.$$

4.2.4 Neighborhood structure

The local search technique embedded within the tabu search heuristic is the steepest descent pairwise exchange heuristic. The basic idea of the heuristic is to find the best solution (or move) in the pairwise exchanges neighborhood of the current solution. Therefore, all possible pairwise exchanges in each period are considered. Each exchange is defined as a move. The input data to the heuristic are an initial layout plan, the flows between departments for each period, the distances between locations, and the rearrangement costs for each department.

The steepest descent pairwise exchange heuristic searches all the candidate moves in the neighborhood of the current solution, and selects the best move for exchange. The candidate moves for the DFLP are the pairwise exchanges of departments. If there are N departments and T periods in the layout, the number of moves in a period is $C_2^N = N(N-1)/2$. The total number of moves is $C_2^N T = N(N-1)T/2$. Next, compute

the total cost for each move, and the best admissible move is selected. A best admissible move is a move which is either non-tabu or tabu restricted such that the total cost of the neighboring solution obtained by performing the move is better than the best solution found so far (aspiration criterion). The best admissible move is the most profitable (or least cost) admissible move. The best admissible move is selected and its corresponding solution becomes the current solution, which is used as the starting solution at the next iteration. The details for the tabu restriction and aspiration criterion will be explained in Section 4.2.6.

Moves (t, u, v)	Tabu or non-tabu	$TC(\pi')$	Change in $TC(\pi)$
(1, 1, 2)	non-tabu	141	-6
(1, 1, 3)	non-tabu	147	-12
(1, 1, 4)	non-tabu	151	-16
(1, 2, 3)	non-tabu	143	-8
(1, 2, 4)*	non-tabu	107	28
(1, 3, 4)	non-tabu	157	-22
(2, 1, 2)	non-tabu	146	-11
(2, 1, 3)	non-tabu	149	-14
(2, 1, 4)	non-tabu	145	-10
(2, 2, 3)	non-tabu	143	-8
(2, 2, 4)	non-tabu	109	26
(2, 3, 4)	non-tabu	136	-1

*Best admissible move

Table 4.3 The list of candidate moves.

An example of the list of candidate moves for the DFLP instance given in Section 4.2.3 is given in Table 4.3. Recall, the current solution is $\pi = ((3, 1, 2, 4), (3, 4, 2, 1))$.

Since there are 4 departments and 2 periods ($N = 4$ and $T = 2$), the total number of moves in the layout is $C_2^4 * 2 = 4(3)(2)/2 = 12$. The first move in the list considers exchanging the locations of departments $u = 1$ and $v = 2$ in period 1 ($t = 1$). This move corresponds to the solution $\pi' = ((1, 3, 2, 4), (3, 4, 2, 1))$. The total cost of this solution is 141 and can be obtained using the technique given in Section 4.2.3. A more efficient technique is given in the next section. Therefore, the change in total cost is -6, which is the total cost of the current solution π minus the solution obtained after performing the move π' ($\Delta TC_{12}(\pi^1) = TC(\pi) - TC(\pi') = -6$, i.e., change in total cost after exchanging the locations of departments 1 and 2 in period 1). The change in total cost is obtained for all solutions in the neighborhood of the current solution, and the best admissible move is selected, which is (1, 2, 4). In other words, the best move is selected which is not tabu, since this is the first iteration (i.e., no moves are tabu, since no moves have been previously performed). Tabu moves will be considered in the next section.

4.2.5 Calculating the Change in Total Cost

Once the objective function value (OFV) is obtained for the initial solution, the OFV of the neighboring solutions can be obtained efficiently by calculating the change in total cost. The change in total cost is the sum of the change in material handling and rearrangement costs. Assume the locations of departments u and v are selected to exchange in period t . Since only the layout in period t is changed and the others are the same, only the change in period t is considered. The change in material handling cost is calculated as follows:

$$\Delta MH_{uv}(\pi^t) = \sum_{\substack{i=1 \\ i \neq u, v}}^N (W_{iu} - W_{iv}) * [d(\pi^t(i), \pi^t(u)) - d(\pi^t(i), \pi^t(v))]$$

The change in rearrangement cost is only related to the departments selected for exchange (departments u and v) in periods $t-1$, t , and $t+1$. For example, if the location of department u in period $t-1$ is different from that in period t , then the rearrangement cost A_{tu} occurs. If the location of department u is changed in period t , then the rearrangement cost $A_{(t+1)u}$ in period $t+1$ also needs to be considered. When computing the partial rearrangement cost before and after exchanging the departments, the difference between them is defined as the change in rearrangement cost. Therefore, the change in rearrangement cost $\Delta RA_{uv}(\pi^t)$ is the change of the rearrangement cost of departments u and v before and after the exchange.

$$\Delta RA_{uv}(\pi^t) = \left(\sum_{i \in I_t^b} A_{ii} + \sum_{i \in I_{t+1}^b} A_{(t+1)i} \right) - \left(\sum_{i \in I_t^a} A_{ii} + \sum_{i \in I_{t+1}^a} A_{(t+1)i} \right)$$

where

I_t^b = all departments (u and/or v) in period t (for $t \geq 2$) with different locations in period $t-1$ before the exchange.

I_t^a = all departments (u and/or v) in period t (for $t \geq 2$) with different locations in period $t-1$ after the exchange.

The change in total cost for the DFLP solution is calculated as follows:

$$\Delta TC_{uv}(\pi^t) = \Delta MH_{uv}(\pi^t) + \Delta RA_{uv}(\pi^t)$$

An example is used to illustrate the calculations of the change in total cost. The first move in the list given in Table 4.3 considers exchanging the locations of departments 1 and 2 in period 1 where the current solution $\pi = ((3, 1, 2, 4), (3, 4, 2, 1))$. Let $u = 1$, $v = 2$ and $t = 1$. Therefore, the change in material handling cost is calculated as follows.

$$\begin{aligned}
\Delta MH_{12}(\pi^1) &= (W_{131}-W_{132})*[d(\pi^1(3), \pi^1(1)) - d(\pi^1(3), \pi^1(2))] \\
&\quad + (W_{141}-W_{142})*[d(\pi^1(4), \pi^1(1)) - d(\pi^1(4), \pi^1(2))] \\
&= (4-8)*[d(2, 3)- d(2, 1)] + (0-2)*[d(4, 3)- d(4, 1)] \\
&= 4
\end{aligned}$$

The change in rearrangement cost is obtained as follows. Since I_t^b and I_t^a are defined for $t \geq 2$, $I_1^b = I_1^a = \phi$. However, $I_2^b = \{2\}$, since the location of department 2 is different before the exchange. In addition, $I_2^a = \{1,2\}$, since both the locations of departments 1 and 2 are different after the exchange. See Figure 4.3 for the solution before and after the exchange. Hence,

$$\begin{aligned}
\Delta RA_{12}(\pi^1) &= \left(\sum_{i \in I_2^b} A_{2i} - \sum_{i \in I_2^a} A_{2i} \right) \\
&= A_{22} - (A_{21} + A_{22}) \\
&= 10 - (10 + 10) = -10.
\end{aligned}$$

$$\begin{array}{ll}
\pi^b = ((3, \boxed{1}, 2, 4), & \pi^a = ((\boxed{1}, \boxed{3}, 2, 4), \\
(3, \boxed{4}, 2, 1)) & (\boxed{3}, \boxed{4}, 2, 1))
\end{array}$$

Figure 4.3 the solution before and after the exchange.

Therefore, the change in total cost is

$$\Delta TC_{12}(\pi^1) = \Delta MH_{12}(\pi^1) + \Delta RA_{12}(\pi^1) = 4 + (-10) = -6.$$

In other words, the change in total cost after exchanging the locations of departments 1 and 2 in period 1 is -6. Hence, the total cost of the corresponding solution is

$$TC(\pi') = TC(\pi) - \Delta TC_{12}(\pi^1) = 135 - (-6) = 141.$$

More specifically, the total cost of the solution after the exchange $\pi^a = ((1, 3, 2, 4), (3, 4, 2, 1))$ is 141.

4.2.6 Components of the Basic Tabu Search Heuristic for the DFLP

4.2.6.1 Tabu list

After performing one iteration of the local search (steepest descent) technique, the best admissible move is defined as tabu (tabu restricted) for a certain number of iterations (tabu length or duration). The tabu list is used to keep track of the tabu moves. More specifically, the tabu list is a set of two-dimensional arrays denoted as *tabu* for each period, which is used to keep track of the status of the tabu restriction. If the exchange of the locations of departments *i* and *k* in period *t* is the current move, then this move is defined as tabu, and set $tabu[t][i][k] = cur_iter + TL$, where *cur_iter* is the current iteration number and *TL* represents the tabu list size (tabu length) which is usually fixed. The value of $tabu[t][i][k]$ where $i < k$ is recorded in the *i*th row and *k*th column of the upper-triangle of the *t*th *tabu* array (i.e., the two-dimensional array for period *t*). The lower-triangle of the *tabu* array is used to keep track of the frequency of the moves and will be explained in Section 4.3.1.1. In addition, if $tabu[t][i][k] < cur_iter$ for $i < k$, then the move to exchange locations of departments *i* and *k* in period *t* is not tabu restricted. Otherwise, the move is tabu restricted.

For example, assume *TL* is 3 and the locations of departments 2 and 4 in period 1 are exchanged in iteration 1, the value of $tabu[1][2][4] = cur_iter + TL = 1 + 3 = 4$ after updating the tabu list. This indicates that the move exchanging the locations of departments 2 and 4 in period 1 is tabu (not allowed to exchange) in iterations 2, 3 and 4. The updated tabu list is shown in Figure 4.4. Thus, the tabu list keeps track of the most

recent moves (i.e., has a short-term memory structure), and its purpose is to prevent cycling.

	1	2	3	4
1				
2				4
3				
4				
	$t = 1$			

Figure 4.4 The updated tabu list for period 1 ($tabu[1][i][k]$) at the end of iteration 1.

4.2.6.2 Aspiration criterion

The aspiration criterion is a short term memory strategy used in the tabu search heuristic to override the tabu restriction. If a move is tabu in the tabu list and produces the solution with better objective function value compared to that of the best found solution so far, the move is admissible. In other words, the tabu restriction is overridden even through the move is tabu.

4.2.6.3 Stopping criterion

The stopping criterion for the proposed tabu search heuristic is computational time (i.e., total run time of the heuristic). This stopping criterion is used to compare the performances of the proposed tabu search heuristics with the same computational time. The detailed settings of the heuristic parameters are explained in Section 5.2. Other commonly used stopping criteria are maximum number of iterations performed and maximum number of consecutive iterations without an improvement.

4.2.7 Pseudo-code for the Basic TS Heuristic for the DFLP

The pseudo-code for the basic TS heuristic for the DFLP is as follows.

Step 0: Input data and initialize heuristic parameters and counters.

Input the following data: the distances between locations (D_{jl}), flows between departments in each period (F_{tik}), and the rearrangement costs for each department (A_{ti}). Next, initialize the tabu list $tabu[t][i][k]$ and the current time cur_time . Also, set tabu length TL , computation time CPU_time , and iteration counter c ($c = 1$).

Step 1: Obtain an initial solution π and calculate its total cost $TC(\pi)$.

The initial layout for period 1 is obtained by assigning department 1 to location 1, department 2 to location 2, and so on. This layout is also used for the other periods. Hence, an initial solution π is constructed. Also, determine the total cost of the solution $TC(\pi)$. Next, set the best found solution so far $\pi_{best} = \pi$ and $TC(\pi_{best}) = TC(\pi)$.

Step 2: Evaluate the neighborhood of the current solution π and select the best admissible move (p, u, v).

Generate all possible moves (t, i, k), where $i < k$, in the neighborhood of π by considering all possible pairwise exchanges between department locations in each period. Evaluate all moves by obtaining the change in total cost of each move $\Delta TC_{ik}(\pi^t) = \Delta MH_{ik}(\pi^t) + \Delta RA_{ik}(\pi^t)$, and the total cost of the solution (π^t) obtained by performing the move (t, i, k) is $TC(\pi^t) = TC(\pi) - \Delta TC_{ik}(\pi^t)$. If the move is either non-tabu or is tabu and $TC(\pi^t) < TC(\pi_{best})$, then the move is an admissible move. Select the best admissible move (p, u, v) with respect to total

cost. This move gives the best admissible solution in the neighborhood of π denoted as π'_{best} .

Step 3: Update the tabu list and current solution.

Set $tabu[p][u][v] = c + TL$, $\pi = \pi'_{best}$, and $TC(\pi) = TC(\pi'_{best})$.

Step 4: Update the best found solution so far.

If $TC(\pi) < TC(\pi_{best})$, set $\pi_{best} = \pi$ and $TC(\pi_{best}) = TC(\pi)$.

Step 5: Update heuristic counter and check stopping criterion.

Set iteration counter $c = c + 1$ and check the current time cur_time . If the $cur_time \geq CPU_time$, then, terminate the heuristic. Otherwise, go to Step 2.

4.3 TS Heuristic for the DFLP with Diversification/Intensification Strategies

4.3.1 Diversification Strategies

The diversification strategies used in one of the proposed heuristics include the frequency-based memory structure with a penalty function for non-improving moves and the dynamic tabu list size as a recency-based memory strategy. Each of the strategies is discussed in details below.

4.3.1.1 Frequency-based Memory

The frequency-based memory structure is employed to keep track of the frequency of the moves. The information is recorded in the lower-triangle of the *tabu* arrays ($tabu[t][i][k]$ where $i > k$). That is, the value $tabu[t][i][k]$ where $i > k$ is the number of times that the locations of departments i and k in period t have been selected as the best admissible move. Recall, in Figure 4.4, the tabu list after performing the move of exchanging the locations of departments 2 and 4 in period 1 is given. More specifically,

the move is tabu for iteration 2, 3, and 4, since the current iteration is 1 and $TL = 3$. For this example, the tabu list $tabu[1][i][k]$ which uses recency and frequency-based memory is given in Figure 4.5. In Figure 4.5, $tabu[1][4][2]$ is 1, which indicates departments 2 and 4 in period 1 have been exchanged once.

	1	2	3	4
1				
2				4
3				
4		1		
	$t = 1$			

Figure 4.5 The tabu list for period 1 ($tabu[1][i][k]$) with recency and frequency-based memory structure at the end of iteration 1.

In addition, a penalty function is used to penalize the non-improving moves. More specifically, the penalty function uses the frequency of a move recorded in the lower-triangle of the *tabu* arrays to penalize a non-improving move. If the non-improving move has been performed frequently in the search process, the penalty function can force the search into other regions by assigning a large penalty to the move using a penalty function. A penalty function is added to the total cost of non-improving moves. If the move is an improvement (i.e., $\Delta TC_{ik}(\pi^t) > 0$) it will not be penalized. Hence, the penalty function for the move of exchanging the locations of the departments i and k in period t is defined as follows.

$$P(t, i, k) = \begin{cases} 0 & \Delta TC_{ik}(\pi') > 0 \text{ (i.e., } TC(\pi') < TC(\pi) \text{)}, \\ a * tabu[t][i][k] & \text{Otherwise.} \end{cases}$$

where

a = penalty value which is a parameter.

Recall, $tabu[t][i][k]$, where $i > k$, is the number of times that departments i and k in period t has been exchanged. If the best admissible move is an improving move, the penalty function is not considered. Otherwise, the penalty function is considered. More specifically, if the best admissible move is a non-improving move, then the total cost of each move is recalculated denoted as $TCp(\pi')$, and is the sum of total cost and penalty function (i.e., $TCp(\pi') = TC(\pi') + P(t, i, k)$). Considering the previous example, at the beginning of iteration 2, the tabu list and the list of candidate moves given in Figure 4.6 and Table 4.4 are used to illustrate how the best admissible move is selected using frequency-based memory. Let $a = 5$. At the start of iteration 2, the current solution is $\pi = ((3, 4, 2, 1), (3, 4, 2, 1))$ where the total cost is 107, which was obtained from Table 4.3 and Section 4.2.4. Also, the total cost of the best found solution is 107. For move (1, 2, 4), the penalty is 5 (i.e., $P(t, i, k) = 5 * tabu[1][4][2] = 5$) since it is a non-improving move and $tabu[1][4][2] = 1$. The move (2, 3, 4) is selected as the best admissible move which is not tabu restricted and is a non-improving solution where $tabu[2][4][3] = 0$. Since this non-improving move had not been performed (i.e., $tabu[2][4][3] = 0$), there is no penalty. Hence, when updating the tabu list, $tabu[2][3][4] = cur_iter + TL = 2 + 3 = 5$ and $tabu[2][4][3] = 1$. The tabu list after updating the move is given in Figure 4.7.

	1	2	3	4
1				
2				4
3				
4		1		
	$t = 1$			

	1	2	3	4
1				
2				
3				
4				
	$t = 2$			

Figure 4.6 The tabu list at the beginning of iteration 2.

Moves (t, i, k)	Tabu or non-tabu	$\Delta TC_{ik}(\pi^t)$	$TC(\pi')$	$P(t, i, k)$	$TCp(\pi')$
(1, 1, 2)	non-tabu	-14	121	0	121
(1, 1, 3)	non-tabu	-28	135	0	135
(1, 1, 4)	non-tabu	-36	143	0	143
(1, 2, 3)	non-tabu	-28	135	0	135
(1, 2, 4)	tabu	-28	135	5	140
(1, 3, 4)	non-tabu	-30	137	0	137
(2, 1, 2)	non-tabu	-21	128	0	128
(2, 1, 3)	non-tabu	-14	121	0	121
(2, 1, 4)	non-tabu	-20	127	0	127
(2, 2, 3)	non-tabu	-18	125	0	125
(2, 2, 4)	non-tabu	-14	121	0	121
(2, 3, 4)*	non-tabu	-11	118	0	118

*Best admissible move

Table 4.4 The list of candidate moves in iteration 2.

	1	2	3	4
1				
2				4
3				
4		1		
	$t = 1$			

	1	2	3	4
1				
2				
3				5
4			1	
	$t = 2$			

Figure 4.7 The updated tabu list at the end of iteration 2.

4.3.1.2 Dynamic Tabu List Size

The dynamic tabu list size (or tabu length) is a recency-based memory strategy used to diversify the search. The dynamic tabu length, which is denoted as TL_d , varies between a lower bound (LB) and an upper bound (UB). It varies from one iteration to another depending on the percentage reduction of the total cost of the best admissible solution ($TC(\pi'_{best})$) from the current solution ($TC(\pi)$). The percentage reduction ($PR(\pi)$) is calculated as follow:

$$PR(\pi) = [TC(\pi) - TC(\pi'_{best})] * 100\% / TC(\pi).$$

If $PR(\pi) \geq \alpha\%$, then the total cost of the best admissible solution $TC(\pi'_{best})$ is a significant improvement over the total cost of the current solution $TC(\pi)$. If the best admissible solution results in a significant improvement, TL_d is set to the upper bound (UB). However, if, the best admissible solution results in a relatively large improvement

(i.e., $PR(\pi) \geq \beta\% > \alpha\%$), TL_d is set to a relatively large value ($TL_\beta = 2NT$). If $PR(\pi) = 0\%$, then TL_d is set to the lower bound (LB). When the best admissible solution results in an improvement between 0 and $\alpha\%$ (i.e., $0\% < PR(\pi) < \alpha\%$), TL_d is set somewhere between LB and UB , and its distance from LB is in proportion to the improvement of this solution. If π'_{best} is not an improving solution (i.e., $PR(\pi) < 0\%$), TL_d is not changed, which implies TL_d is the same as the TL_d in the pervious iteration. A summary of how TL_d is set is given in Table 4.5.

% Reduction of the total cost ($PR(\pi)$)	Dynamic tabu list size (TL_d)
$\geq \beta\%$	$TL_\beta = 2NT$
$\alpha\% \leq PR(\pi) < \beta\%$	UB
$0\% \leq PR(\pi) < \alpha\%$	$LB + (UB-LB)* PR(\pi)/\alpha\%$
$PR(\pi) < 0\%$	TL_d in the pervious iteration

Table 4.5 Dynamic tabu list size.

The previous example is used to illustrate the determination of the dynamic tabu list size (TL_d). Let $UB = 6$, $LB = 2$, and $\alpha\% = 25\%$. Recall, the total cost of the initial solution is 135 (in Section 4.2.3), and the total cost of the best admissible move in iteration 1 and 2 is 107 (in Section 4.2.4) and 118 (in Section 4.3.1.1), respectively. In iteration 1,

$$PR(\pi) = [TC(\pi) - TC(\pi'_{best})] * 100\% / TC(\pi) = (135 - 107) * 100\% / 135 = 20.7\%.$$

Since $0\% \leq PR(\pi) = 20.7\% < \alpha\% = 25\%$,

$$TL_d = LB + (UB - LB) * PR(\pi) / \alpha\% = 2 + (6 - 2) * 20.7\% / 25\% = 5.3 \cong 5.$$

In iteration 2,

$$PR(\pi) = [TC(\pi) - TC(\pi'_{best})] * 100\% / TC(\pi) = (107 - 118) * 100\% / 107 = -10.3\%.$$

Since $PR(\pi) = -10.3\% < 0\%$, $TL_d = TL_d$ in the pervious iteration (iteration 1) which is 5 (i.e., $TL_d = 5$). The updated dynamic tabu list at the end of iteration 2 is given in Figure 4.8.

	1	2	3	4
1				
2				6
3				
4		1		

$t = 1$

	1	2	3	4
1				
2				
3				7
4			1	

$t = 2$

Figure 4.8 The dynamic tabu list in at the end of iteration 2.

4.3.2 Intensification Strategy

The basic idea of the intensification strategy is to search the neighborhoods of good solutions more in depth. The way to intensify the search is to fix pairs of departments which result in large percentage reductions (at least $\gamma\%$) from the total cost of the best solution found so far, after exchanging their locations in specific periods. In other words, if the percentage reduction of the total cost of the best admissible solution ($TC(\pi'_{best})$) is equal to or more than $\gamma\%$ from the total cost of the best found solution so far ($TC(\pi_{best})$), the pair of departments (u, v) of the best admissible move is fixed in period p . To fix the pair of departments (u, v) , the locations of the departments in period p is not allowed to

change until they are freed. The percentage reduction of the total cost of the best admissible solution from the total cost of the best found so far ($PR(\pi_{best})$) is calculated as follows:

$$PR(\pi_{best}) = [TC(\pi_{best}) - TC(\pi'_{best})] * 100\% / TC(\pi_{best}).$$

If $PR(\pi_{best}) \geq \gamma\%$, then the total cost of the best admissible solution $TC(\pi'_{best})$ is a large reduction over the total cost of the best solution found so far $TC(\pi_{best})$, and the pair of departments of the best admissible move (p, u, v) is fixed (i.e., departments u and v are not allowed to change in period p) until freed.

The intensification strategy starts after a certain number of iterations (η) have been performed, since large percentage reduction moves are generated in the first η iterations. Hence, intensification is invoked after obtaining “good” solutions. As stated previously, the fixed departments (u, v) in period p are allowed to change their locations if the departments are freed. The fixed departments can be freed only when the exchange of the locations of the fixed departments with other departments produces an improvement in the total cost of the best found solution so far. This is different from the way Chiang and Kouvelis (1996) freed the fixed departments. In contrast, they freed the fixed departments only when the exchange of the locations of the fixed departments with other free departments produces a percentage reduction better than that at which the departments were fixed.

The previous example is used to illustrate the intensification strategy. Let $\eta = 4$ and $\gamma\% = 2\%$. Therefore, the TS heuristic performs four iterations until intensification is invoked at iteration 5. At iteration 5, the total cost of the best solution found so far is 107 (i.e., $TC(\pi_{best}) = 107$), and the total cost of the best admissible solution is 103 (i.e., $TC(\pi'_{best}) = 103$). The best admissible solution is obtained from the best admissible

move (2, 1, 2), which is to exchange the locations of departments 1 and 2 in period 2.

Then the percentage reduction is calculated as follows:

$$PR(\pi_{best}) = [TC(\pi_{best}) - TC(\pi'_{best})] * 100\% / TC(\pi_{best}) = (107 - 103) * 100\% / 107 = 3.7\%.$$

Since $PR(\pi_{best}) = 3.7\% \geq \gamma\% = 2\%$, departments 1 and 2 in period 2 are fixed until the exchange of the locations of departments 1 and 2 with other free departments yield a solution better than the best found solution so far.

4.3.3 Pseudo-code for TS Heuristic with Diversification/Intensification Strategies

The pseudo-code for the TS heuristic with the diversification and intensification strategies mentioned above for the DFLP is given below.

Step 0: Input data and initialize heuristic parameters and counters.

Input the following data: the distances between locations (D_{jl}), flows between departments in each period (F_{tik}), and the rearrangement costs for each department (A_{ti}). Next, initialize the tabu list $tabu[t][i][k]$ and the current time cur_time . Also, set penalty value a , upper bound UB , lower bound LB , initial tabu length $TL_d = LB$, significant improvement from the total cost of current solution $\alpha\%$, relatively large improvement from the total cost of current solution $\beta\%$, dynamic tabu length of $\beta\%$ reduction $TL_\beta = 2NT$, the number of iterations after intensification is invoked η , large percentage reduction from the total cost of the best found solution so far $\gamma\%$, computation time CPU_time , and iteration counter c ($c = 1$).

Step 1: Obtain an initial solution π and calculate its total cost $TC(\pi)$.

The initial solution π and the total cost $TC(\pi)$ are obtained as discussed in Step 1 of the basic TS heuristic. Next, set the best found solution $\pi_{best} = \pi$ and $TC(\pi_{best}) = TC(\pi)$.

Step 2: Evaluate the neighborhood of the current solution π and select the best admissible move (p, u, v) .

Generate all possible moves (t, i, k) , where $i < k$, in the neighborhood of π by considering all possible pairwise exchanges between department locations in each period. Evaluate all moves by obtaining the change in total cost of each move $\Delta TC_{ik}(\pi') = \Delta MH_{ik}(\pi') + \Delta RA_{ik}(\pi')$, and the total cost of the solution (π') obtained by performing the move (t, i, k) is $TC(\pi') = TC(\pi) - \Delta TC_{ik}(\pi')$. Also, the sum of total cost and penalty function $TCp(\pi') = TC(\pi') + P(t, i, k)$ is obtained. If $\Delta TC_{ik}(\pi') > 0$, the penalty function $P(t, i, k) = 0$. Otherwise, the penalty function $P(t, i, k) = a * tabu[t][k][i]$. If the move (t, i, k) satisfies one of the following restrictions, the move is defined as an admissible move.

- a. If the move (t, i, k) is non-tabu and departments i or k are not fixed in period t , then the move is an admissible move.
- b. If the move (t, i, k) is tabu and $TCp(\pi') < TC(\pi_{best})$, then the move is an admissible move.

Select the best admissible move (p, u, v) with respect to $TCp(\pi')$. This move gives the best admissible solution in the neighborhood of π denoted as π'_{best} .

Step 3: Update the dynamic tabu list size TL_d .

The dynamic tabu length TL_d is obtained by calculating the percentage reduction of the total cost of the best admissible solution from the current solution using $PR(\pi) = [TC(\pi) - TC(\pi'_{best})] * 100\% / TC(\pi)$. Set the dynamic tabu list size (TL_d) as follows.

- a. If $PR(\pi) \geq \beta\%$, set $TL_d = TL_\beta$.

- b. If $\alpha\% \leq PR(\pi) < \beta\%$, set $TL_d = UB$.
- c. If $0\% \leq PR(\pi) < \alpha\%$, set $TL_d = LB + (UB-LB) * PR(\pi) / \alpha\%$.
- d. If $PR(\pi) < 0\%$, set $TL_d = TL_d$ in the pervious iteration.

Step 4: Invoke the intensification strategy.

If $c > \eta$, invoke intensification strategy. Otherwise, go to Step 5.

Step 4.1: If the total cost of best admissible solution $TC(\pi'_{best}) < TC(\pi_{best})$, free the departments u and v in period p if necessary.

Step 4.2: Calculate the percentage reduction of the total cost of the best admissible solution from the total cost of the best found so far which is $PR(\pi_{best}) = [TC(\pi_{best}) - TC(\pi'_{best})] * 100\% / TC(\pi_{best})$.

Step 4.3: If the reduction $PR(\pi_{best}) \geq \gamma\%$, fix departments u and v in period p .

Step 5: Update the tabu list and current solution.

Set $tabu[p][u][v] = c + TL_d$, $tabu[p][v][u] = tabu[p][v][u] + 1$, $\pi = \pi'_{best}$, and $TC(\pi) = TC(\pi'_{best})$.

Step 6: Update the best found solution so far.

If $TC(\pi) < TC(\pi_{best})$, set $\pi_{best} = \pi$ and $TC(\pi_{best}) = TC(\pi)$.

Step 7: Update heuristic counter and check stopping criterion.

Set iteration counter $c = c + 1$ and check the current time cur_time . If the $cur_time \geq CPU_time$, then, terminate the heuristic. Otherwise, go to Step 2.

4.4 The Probabilistic Tabu Search Heuristic for the DFLP

The probabilistic tabu search (PTS) heuristic is a modification of the basic tabu search heuristic (TS) presented in Section 4.2.7, which randomly selects an admissible move for exchange. Chiang and Chiang (1998) presented a PTS heuristic to solve the

SFLP. Also, Lim *et al.* (2003) presented a PTS heuristic to solve the crane scheduling problem. The proposed PTS heuristic for the DFLP is a modification of the PTS heuristics presented by Chiang and Chiang (1998) and Lim *et al.* (2003) for the DFLP.

The only difference between the basic TS and PTS heuristics is how the admissible move is selected. As before, all possible moves in the neighborhood of the current solution are evaluated. However, the PTS heuristic selects the top M admissible moves as candidate moves and puts them in order from best to worst in the candidate list. If the best admissible move in the candidate list yields a solution with a total cost less than that of the best found solution so far, the best found solution is updated. Then the following procedure is used to select a move from the candidate list.

Step 1: Consider the first move in the candidate list as the current move.

Step 2: Accept the current move with probability p . If the move is accepted, then this move is selected as the admissible move for exchange, and exit the procedure, since an admissible move was selected. Otherwise, go to step 3.

Step 3: Go to the next move in the candidate list and set as the current move. If there is no more candidate moves in the list, select the best move from the list of candidate moves as the admissible move for exchange and exit the procedure. Otherwise go to step 2.

The probability to accept a move in the above procedure is p . Therefore, the probability of selecting the first move in the candidate list as the admissible move is p . If the first move is rejected, the probability of selecting the second move as the admissible move is $p(1-p)$. If the $(i-1)$ th move is rejected, the probability of selecting the i th move as the admissible move is $p(1-p)^{i-1}$. The accumulated probability ($AP(i)$) for each move in the candidate list is calculated by using the following formulas:

$$AP(i)=\begin{cases} 0, & \text{for } i > M \text{ or } i < 1, \\ p(1-p)^{M-i}, & \text{for } i = M, \\ AP(i+1) + p(1-p)^{i-1}, & \text{for } 2 \leq i < M, \\ 1, & i = 1. \end{cases}$$

A more efficient way of selecting the admissible move is to use the accumulated probability ($AP(i)$) formulas. More specifically, a random number x between 0 and 1 is generated to select the admissible move for exchange. If $AP(i+1) < x \leq AP(i)$ then the i th move in the candidate list is selected as the admissible move with probability $p(1-p)^{i-1}$. For example, given $M = 10$ and $p = 0.33$, the accumulated probability ($AP(i)$) for each move is listed in Table 4.6. If $x = 0.45$, then $0.430672 < x = 0.45 \leq 0.651772$. Therefore, $i = 2$, which means that the second move is selected as the admissible move.

The i th Move	$AP(i)$
1	1
2	0.651772
3	0.430672
4	0.282535
5	0.183283
6	0.116784
7	0.072232
8	0.042379
9	0.022378
10	0.008978

Table 4.6 The accumulated probability table.

CHAPTER 5

COMPUTATIONAL RESULTS

5.1 Introduction

The proposed tabu search heuristics were tested on two data sets given by Lacksonen and Enscore (1993) and Balakrishnan and Cheng (2000). The first data set which is taken from the Lacksonen and Enscore (1993) contains test problems with 6, 12, 20, and 30 departments (i.e., $N = 6, 12, 20,$ and 30) each with 3 and 5 periods (i.e., $T = 3$ and 5). Each problem instance includes four test problems. Therefore, there are 32 test problems in this data set. More specifically, the data set was developed based on six factors: number of departments, number of time periods, ratio of rearrangement costs and flow costs, percentage of new departments in a period, percentage of department pairs with positive flows, and maximum flow change of a department pair in consecutive period. For some problem instances, new departments are introduced to replace the existing departments, and the cost to replace a department is equal to the rearrangement cost of that department, regardless of whether the location is the same as the previous department. Also, rearrangement costs are the same for each department in each period within a test problem.

The second data set which is taken from Balakrishnan and Cheng (2000) includes test problems with 6, 15, and 30 departments (i.e., $N = 6, 15,$ and 30) each with 5 and 10 periods (i.e., $T = 5$ and 10). Each problem instance contains eight test problems. Therefore, there are 48 test problems in this data set. More specifically, the sum of flows within a period is constant during the layout planning horizon for each test problem. However, the rearrangement costs are different according to the functions of the

departments, and the average rearrangement cost for a department was set to be 15% of the average material handling cost for the department. In addition, the five period problems use the data of the first five periods from the corresponding ten period problems.

5.2 Setting Parameters

The settings for each parameter in the proposed tabu search heuristics are given below:

1. Tabu length (TL)

For data set 1, Kaku and Mazzola (1997) defined the tabu length to be $NT/2$ for their TS heuristic where N is the number of departments and T is the number of periods. After performing experiments, the tabu length is defined as $NT/2$ and $NT/4$ for data sets 1 and 2, respectively, for the proposed basic TS and PTS heuristics. Also, the tabu length defined here is used as the midpoint between the LB and UB in the TS heuristic with (intensification/diversification) strategies.

2. Penalty value (a)

Chiang and Kouvelis (1996) used a penalty function for their TS heuristic and their penalty value was 7. After performing experiments, the penalty value of 3 and 15 were obtained for data sets 1 and 2, respectively, for the proposed TS heuristic with strategies. Note: a larger penalty is used for data set 2, since the objective function values (OFVs) are relatively larger.

3. Upper bound (UB) and lower bound (LB) for the dynamic tabu list size

Chiang and Kouvelis (1996) used a dynamic tabu list size strategy for their TS heuristic, and their settings of lower and upper bounds where (LB, UB) is defined as $(N/3, 3N/4)$

where N is the number of departments. After performing experiments, the upper bound (UB) and lower bound (LB) for the dynamic tabu list size for the proposed TS heuristic with strategies are set by using the tabu length (TL) as the mid-point or median value of the range from the LB to the UB (as mentioned in (1) above). Recall, the tabu length (TL) is defined as $NT/2$ and $NT/4$ for data sets 1 and 2, respectively. Therefore, the setting for (LB, UB) is $(\frac{1}{4}NT, \frac{3}{4}NT)$ and $(\frac{3}{16}NT, \frac{5}{16}NT)$ for data sets 1 and 2, respectively.

4. Significant improvement from the total cost of the current solution ($\alpha\%$) and the significant percentage reduction from the total cost of the best found solution so far ($\gamma\%$)

Chiang and Kouvelis (1996) used diversification and intensification strategies for their TS heuristic and the setting for $\alpha\%$ is 0.05%, which is the same setting in both the diversification and intensification strategies (i.e., the authors used $\alpha\%$ for both percentage reductions which mean $\gamma\% = \alpha\%$). The parameters of significant improvements from the total cost of current solution ($\alpha\%$) and significant percentage reduction from the total cost of the best found solution so far ($\gamma\%$) for the proposed TS heuristic with strategies are set as parameters in the diversification and intensification strategies, respectively. The parameters of $\alpha\%$ and $\gamma\%$ are set based on gathering statistics from performing initial runs of the steepest descent pairwise exchange heuristic. The procedure for setting the parameters $\alpha\%$ and $\gamma\%$ is explained below.

Step 1: Use the steepest descent pairwise exchange heuristic to generate statistics for setting the parameters $\alpha\%$ and $\gamma\%$.

Step 1.0: Input the following data: the distances between locations (D_{jl}), flows between departments in each period (F_{tik}), and the rearrangement costs

for each department (A_{ii}). Also, set the total number of initial solutions generated for each test problem $initial_n$ and initialize the initial solution counter ic (i.e., set $ic = 1$). The total number of initial solutions ($initial_n$) generated for each test problem is given in Tables 5.1 and 5.2 for data sets 1 and 2, respectively.

Problem size		$initial_n$
N	T	
6	3	50
	5	50
12	3	100
	5	100
20	3	200
	5	400
30	3	600
	5	800

Table 5.1 The total number of initial solutions for data set 1.

Problem size		$initial_n$
N	T	
6	5	100
	10	100
15	5	200
	10	400
30	5	800
	10	1600

Table 5.2 The total number of initial solutions for data set 2.

Step 1.1: Obtain an initial solution by randomly assigning the departments to locations for period 1 and use the same layout for each period. Hence, an initial solution $\pi_{initial}$ is obtained. Also, determine the total cost of the initial solution which is denoted as $TC(\pi_{initial})$. Next, set the current solution $\pi = \pi_{initial}$ and $TC(\pi) = TC(\pi_{initial})$.

Step 1.2: Run the steepest descent pairwise exchange heuristic by evaluating all the candidate moves in the neighborhood of the current solution π . The best move is selected, and its corresponding solution is the best solution found so far which is denoted as π'_{best} . Also, obtain the total cost of the best solution so far $TC(\pi'_{best})$.

Step 1.3: Calculate the percentage reduction ($is(\pi)$) of the total cost of the best solution so far $TC(\pi'_{best})$ from the current solution $TC(\pi)$. The percentage reduction ($is(\pi)$) is calculated as follow:

$$is(\pi) = [TC(\pi) - TC(\pi'_{best})] * 100\% / TC(\pi)$$

If $is(\pi) > 0$, then record the percentage reduction $is(\pi)$ in a list, set the best solution as the current solution (i.e., $\pi = \pi'_{best}$), and go to Step 1.2.

Else, record the percentage reduction $ds(\pi_{initial})$ from the initial solution to the current solution by using the formula

$$ds(\pi_{initial}) = [TC(\pi_{initial}) - TC(\pi)] * 100\% / TC(\pi_{initial}),$$

and if $ic < initial_n$, then set $ic = ic + 1$ and go to Step 1.1.

Step 2: Repeat the above steps for each test problem in the data set.

Step 3: Obtain the minimum (min), average (avg), and maximum (max) values for $is(\pi)$, and $ds(\pi_{initial})$ from the lists. The values are given in Tables 5.3 and 5.4.

problem size		Problem No.	Diversification			Intensification		
N	T		$\min(ds(\pi_{initial}))$	$\text{avg}(ds(\pi_{initial}))$	$\max(ds(\pi_{initial}))$	$\min(is(\pi))$	$\text{avg}(is(\pi))$	$\max(is(\pi))$
6	3	P01	0.0100	0.0500	0.1000	0.0010	0.0050	0.0100
		P02	0.0100	0.0500	0.1000	0.0010	0.0050	0.0100
		P03	0.0100	0.0500	0.1000	0.0010	0.0050	0.0100
		P04	0.0100	0.0500	0.1000	0.0010	0.0050	0.0100
	5	P05	0.0100	0.0100	0.1693	0.0010	0.0028	0.0301
		P06	0.0100	0.0085	0.0509	0.0010	0.0027	0.0104
		P07	0.0100	0.0500	0.1000	0.0010	0.0050	0.0100
		P08	0.0100	0.0500	0.1000	0.0010	0.0050	0.0100
12	3	P09	0.0325	0.1773	0.2663	0.0075	0.0163	0.0290
		P10	0.0235	0.0998	0.1669	0.0052	0.0118	0.0191
		P11	0.0100	0.1066	0.2105	0.0010	0.0151	0.0330
		P12	0.0100	0.0427	0.1180	0.0010	0.0075	0.0161
	5	P13	0.0166	0.0385	0.0523	0.0032	0.0071	0.0113
		P14	0.0313	0.0772	0.1224	0.0022	0.0045	0.0071
		P15	0.0100	0.0528	0.1819	0.0010	0.0075	0.0210
		P16	0.0100	0.0119	0.1074	0.0010	0.0020	0.0121
20	3	P17	0.0568	0.1904	0.2998	0.0101	0.0173	0.0305
		P18	0.0380	0.1381	0.2166	0.0065	0.0115	0.0169
		P19	0.0048	0.1653	0.4027	0.0008	0.0103	0.0231
		P20	0.0103	0.0675	0.1426	0.0010	0.0067	0.0146
	5	P21	0.0979	0.2733	0.4361	0.0041	0.0069	0.0123
		P22	0.0884	0.1779	0.2746	0.0021	0.0037	0.0060
		P23	0.0182	0.1150	0.4222	0.0010	0.0092	0.0243
		P24	0.0177	0.1065	0.3070	0.0010	0.0071	0.0148
30	3	P25	0.0340	0.0655	0.0926	0.0043	0.0085	0.0136
		P26	0.0200	0.0392	0.0577	0.0029	0.0052	0.0090
		P27	0.0032	0.0193	0.0335	0.0035	0.0068	0.0147
		P28	0.0376	0.1547	0.2671	0.0020	0.0042	0.0079
	5	P29	0.1265	0.2387	0.3223	0.0020	0.0031	0.0048
		P30	0.0075	0.0134	0.0175	0.0011	0.0018	0.0027
		P31	0.0097	0.1306	0.2612	0.0010	0.0073	0.0125
		P32	0.0262	0.1363	0.2501	0.0023	0.0046	0.0075

Table 5.3 Minimum, average, and maximum values for data set 1.

Problem size		Problem No.	Diversification			Intensification		
N	T		$\min(ds(\pi_{initial}))$	$avg(ds(\pi_{initial}))$	$\max(ds(\pi_{initial}))$	$\min(is(\pi))$	$avg(is(\pi))$	$\max(is(\pi))$
6	5	P01	0.0003	0.0200	0.0398	0.0003	0.0042	0.0081
		P02	0.0085	0.0526	0.0968	0.0085	0.0098	0.0112
		P03	0.0090	0.0199	0.0314	0.0090	0.0120	0.0140
		P04	0.0084	0.0546	0.1319	0.0006	0.0093	0.0156
		P05	0.0087	0.0235	0.0268	0.0085	0.0135	0.0146
		P06	0.0417	0.0774	0.0836	0.0042	0.0089	0.0100
		P07	0.0281	0.0435	0.0821	0.0091	0.0146	0.0234
		P08	0.0410	0.0582	0.0795	0.0095	0.0133	0.0161
	10	P09	0.0087	0.0431	0.0734	0.0029	0.0049	0.0070
		P10	0.0455	0.0943	0.0960	0.0036	0.0062	0.0073
		P11	0.0613	0.0768	0.0853	0.0054	0.0078	0.0088
		P12	0.0108	0.0535	0.1211	0.0027	0.0051	0.0083
		P13	0.0686	0.0934	0.1059	0.0084	0.0109	0.0148
		P14	0.0248	0.0545	0.0792	0.0033	0.0046	0.0065
		P15	0.0127	0.0480	0.0796	0.0051	0.0093	0.0114
		P16	0.0693	0.0735	0.0851	0.0038	0.0056	0.0073
15	5	P17	0.0582	0.1176	0.1620	0.0023	0.0040	0.0065
		P18	0.0357	0.0788	0.1431	0.0017	0.0035	0.0067
		P19	0.0526	0.0936	0.1303	0.0021	0.0041	0.0068
		P20	0.0546	0.0923	0.1280	0.0022	0.0040	0.0081
		P21	0.0419	0.0976	0.1489	0.0021	0.0040	0.0067
		P22	0.0531	0.1057	0.1611	0.0028	0.0054	0.0112
		P23	0.0703	0.1241	0.1556	0.0027	0.0042	0.0069
		P24	0.0620	0.1060	0.1534	0.0021	0.0041	0.0070
	10	P25	0.0702	0.0975	0.1241	0.0011	0.0019	0.0030
		P26	0.0464	0.0977	0.1550	0.0011	0.0020	0.0030
		P27	0.0662	0.0917	0.1139	0.0012	0.0020	0.0032
		P28	0.0756	0.1043	0.1364	0.0015	0.0023	0.0034
		P29	0.0684	0.1007	0.1304	0.0013	0.0022	0.0041
		P30	0.0696	0.1052	0.1340	0.0026	0.0042	0.0061
		P31	0.0550	0.0947	0.1390	0.0010	0.0019	0.0033
		P32	0.0600	0.0980	0.1369	0.0011	0.0020	0.0035
30	5	P33	0.0218	0.0544	0.0901	0.0008	0.0017	0.0033
		P34	0.0215	0.0610	0.1132	0.0017	0.0033	0.0054
		P35	0.0345	0.0602	0.1008	0.0010	0.0017	0.0033
		P36	0.0393	0.0730	0.1089	0.0012	0.0022	0.0037
		P37	0.0430	0.0842	0.1232	0.0016	0.0029	0.0052
		P38	0.0313	0.0715	0.1073	0.0014	0.0027	0.0053
		P39	0.0122	0.0643	0.1122	0.0011	0.0022	0.0037
		P40	0.0321	0.0575	0.0801	0.0022	0.0038	0.0067
	10	P41	0.0192	0.0489	0.0794	0.0005	0.0009	0.0013
		P42	0.0196	0.0480	0.0755	0.0009	0.0016	0.0026
		P43	0.0196	0.0401	0.0603	0.0011	0.0018	0.0028
		P44	0.0375	0.0652	0.0922	0.0007	0.0011	0.0017
		P45	0.0491	0.0786	0.1039	0.0008	0.0014	0.0022
		P46	0.0412	0.0665	0.0940	0.0008	0.0013	0.0021
		P47	0.0372	0.0587	0.0777	0.0013	0.0022	0.0036
		P48	0.0321	0.0490	0.0670	0.0007	0.0010	0.0016

Table 5.4 Minimum, average, and maximum values for data set 2.

Step 4: Generate parameters $\alpha\%$ and $\gamma\%$ by using formulas related to the statistics.

Step 4.1: After experimenting with the values $\min(ds(\pi_{initial}))$, $avg(ds(\pi_{initial}))$, and $\max(ds(\pi_{initial}))$, the following formulas were obtained for setting the heuristic parameter $\alpha\%$:

$$\alpha\% = 0.4 \min(ds(\pi_{initial})) \text{ for data set 1 and}$$

$$\alpha\% = 0.6 \min(ds(\pi_{initial})) \text{ for data set 2.}$$

Step 4.2: After experimenting with the values $\min(is(\pi))$, $avg(is(\pi))$, and $\max(is(\pi))$, the following formulas were obtained for setting the heuristic parameter $\gamma\%$:

$$\gamma\% = avg(is(\pi)) \text{ for data set 1 and}$$

$$\gamma\% = \max(is(\pi)) \text{ for data set 2.}$$

5. Relatively large improvement from the total cost of current solution ($\beta\%$) and dynamic tabu length of $\beta\%$ reduction (TL_β)

Chiang and Kouvelis (1996) used a dynamic tabu list size strategy for their TS heuristic and their settings of $\beta\%$ is much larger than $\alpha\%$ (the authors did not mention the specific value or formula for $\beta\%$) and TL_β is set as $2N$ where N is the number of departments. Since a relatively large improvement from the total cost of the current solution ($\beta\%$) should be much larger than $\alpha\%$, the value of $\beta\%$ is set as $2\alpha\%$, which is $\beta\% = 0.8 * \min(ds(\pi_{initial}))$ and $1.2 * \min(ds(\pi_{initial}))$ for data sets 1 and 2, respectively. Dynamic tabu length (TL_β) of $\beta\%$ reduction is set to a large value which is $TL_\beta = 2NT$ for both data sets.

6. The iteration number after which the intensification strategy is invoked (η)

Chiang and Kouvelis (1996) used an intensification strategy in their TS heuristic, and the authors mentioned that the iteration number (η) cannot be too small, since intensification would be performed with “poor” solutions; thus, increasing computational time. After performing experiments, the iteration number after which the intensification strategy is

invoked (η) is set using a formula based on the number of departments (N) and number of periods (T). The formulas for η are NT and $20NT$ for data sets 1 and 2, respectively.

7. The total number of moves in the candidate list (M) and the probability (p) to accept a move

Chiang and Chiang (1998) used a PTS heuristic, and their settings for M and p are 10 and 0.33, respectively. After performing experiments, the proposed PTS heuristic performs better when M is set as 15 and 5 for data sets 1 and 2, respectively. Also, the probability p is set as 0.33 for both data sets.

8. Computational time (*CPU_time*)

Computational time for each problem instance is defined according to the problem size. Furthermore, the TS heuristic with strategies were used to determine the setting of the stopping criterion, since this heuristic requires more computation time. More specifically, first the number of consecutive iterations without improvement was used as the stopping criterion which was defined as $2NT$ for the TS heuristic with strategies. Using this stopping criterion, the run times were obtained for each test problem in each data set. The maximum time for each set of test problems with the same number of departments (N) was determined and was rounded to the nearest preferred integer. These times were used as the computational times for all three of the proposed heuristics so that the performances of these heuristics can be compared. This process was performed for each data set. However, since it was more difficult to solve the test problems in data set 1 with 30 departments, the computational times for these problems were increased. The computational time for each problem instance in data sets 1 and 2 is given in Tables 5.5 and 5.6, respectively.

Problem size		Computational time (min)
N	T	
6	3	0.33333
	5	0.5
12	3	1.5
	5	4
20	3	8
	5	12
30	3	15
	5	18

Table 5.5 The computational time for data set 1.

Problem size		Computational time (min)
N	T	
6	5	0.01667
	10	0.01667
15	5	0.16667
	10	0.58333
30	5	3
	10	12

Table 5.6 The computational time for data set 2.

The detailed settings for each parameter of the proposed TS heuristics are given in Tables 5.7 and 5.8, respectively.

Heuristic/Strategy		Parameter	Setting
Basic TS		TL	$NT/2$
TS with Strategies	Frequency-based Memory	a	3
	Dynamic tabu size list	LB	$1/4*NT$
		UB	$3/4*NT$
		$\alpha\%$	$0.4 \min(ds(\pi_{initial}))$
		$\beta\%$	$0.8 \min(ds(\pi_{initial}))$
		TL_{β}	$2NT$
	Intensification	η	NT
		$\gamma\%$	$avg(is(\pi))$
PTS		M	15
		p	0.33

Table 5.7 Parameter settings for data set 1.

Heuristic/strategy		Parameter	Setting
Basic TS		TL	$NT/4$
TS with Strategies	Frequency-based Memory	a	15
	Dynamic tabu size list	LB	$3/16*NT$
		UB	$5/16*NT$
		$\alpha\%$	$0.6 \min(ds(\pi_{initial}))$
		$\beta\%$	$1.2 \min(ds(\pi_{initial}))$
		TL_{β}	$2NT$
	Intensification	η	$20NT$
		$\gamma\%$	$max(is(\pi))$
PTS		M	5
		p	0.33

Table 5.8 Parameter settings for data set 2.

5.3 Results

The proposed TS heuristics were programmed using the C++ programming language, and the test problems from Lacksonen and Enscore (1993) as well as Balakrishnan and Cheng (2000) data sets were used to test the performances of the proposed heuristics. The test problems were solved on an AMD Athlon 2600+ 1.92 GHz PC with 1 G of memory. As mentioned in the previous section, the stopping criterion was computation time. In other words, the computation time for each test problem for each of the proposed heuristics was the same. For the computation times and the heuristic parameter settings, see Tables 5.5 - 5.8. Since the PTS heuristic is a stochastic heuristic, while the other two heuristics are deterministic, each test problem was solved three times using the PTS heuristic. The results for both data sets are given below.

5.3.1 Data Set from Lacksonen and Enscore (1993)

First, the test problems were solved using the basic TS heuristic, and then each strategy and combination of strategies were added to this heuristic to measure their impact. Table 5.9 summarizes the results obtained by the basic TS heuristic (TSbasic) as well as the basic TS heuristic with frequency based memory (TSfre), dynamic tabu list size (TSdiver), intensification strategy (TSinten), frequency based memory and dynamic tabu list size (TSfrediver), frequency based memory and intensification (TSfreinten), dynamic tabu list size and intensification (TSdiverinten), and all strategies combined (TSall). In the last column, the best solution obtained is given, and the bold numbers indicate the best objective function value (OFV) obtained from the heuristic for each test problem in this data set. In the last row, the number of the best solutions obtained from the TS heuristic with each

combination of strategies is given. The TSbasic, TSfre, TSdiver, TSinten, TSfrediver, TSfreinten, TSdiverinten, and TSall heuristics obtained the best solution for 21, 17, 22, 17, 24, 16, 18 and 26 of the 32 test problems, respectively. Therefore, the proposed TS heuristic with all strategies is the preferred TS heuristic for the data set taken from Lacksonen and Enscore (1993).

Table 5.10 summarizes the results obtained by the PTS heuristic. The result for each run and the average of the three runs for each of the test problems are given. In the last column, the best solution obtained from the PTS heuristic for the data set taken from Lacksonen and Enscore (1993) is given. The bold numbers indicate the best objective function value (OFV) obtained from the PTS heuristic for each test problem.

Tables 5.11 - 5.14 summarize the results obtained by the proposed heuristics (basic TS (TSbasic), the TS heuristic with all strategies (TSall), and the PTS heuristic (PTS)) as well as the heuristics presented by Lacksonen and Enscore (1993) using a cutting plane algorithm (CP), Kaku and Mazzola (1997) using a tabu search heuristic (TS-KM), McKendall and Shang (2005) using hybrid ant systems (HAS), and McKendall *et al.* (2005) using SA heuristics (SA). The proposed TS heuristics are compared to the cutting plane algorithm CP since this is the best exact method available in the literature. Also, the proposed TS heuristics are compared to the tabu search heuristic TS-KM, since this is the only paper which presents a TS heuristic for DFLP. Furthermore, the results presented using HAS and SA heuristics are also used in the analysis since these techniques perform well on this data set.

Pro. size		Problem No.	TSbasic	TSfre	TSdiver	TSinten	TSfrediver	TSfreinten	TSdiverinten	TSall	Best solution
<i>N</i>	<i>T</i>										
6	3	P01	267	267	267	267	267	267	267	267	267
		P02	260	260	260	260	260	260	260	260	260
		P03	363	363	363	368	363	363	363	363	363
		P04	299	299	299	299	299	299	299	299	299
	5	P05	442	442	442	442	442	442	442	442	442
		P06	586	586	586	586	586	586	586	586	586
		P07	424	424	424	424	424	424	424	424	424
		P08	428	428	428	428	428	428	428	428	428
12	3	P09	1624	1624	1624	1624	1624	1624	1624	1624	1624
		P10	1973	1973	1973	1973	1973	1973	1973	1973	1973
		P11	1661	1665	1661	1665	1661	1665	1661	1661	1661
		P12	2102	2097	2097	2102	2097	2097	2097	2097	2097
	5	P13	2930	2930	2943	2930	2930	2930	2943	2930	2930
		P14	3701	3703	3701	3701	3709	3703	3701	3701	3701
		P15	2779	2756	2756	2765	2756	2756	2756	2756	2756
		P16	3364	3364	3364	3366	3364	3382	3387	3364	3364
20	3	P17	2758	2758	2822	2758	2758	2758	2822	2758	2758
		P18	5318	5318	5318	5318	5318	5318	5318	5318	5318
		P19	3034	3054	3034	3034	3085	3102	3038	3034	3034
		P20	5873	5873	5873	5873	5869	5904	5873	5869	5869
	5	P21	4554	4588	4554	4554	4573	4588	4554	4573	4554
		P22	9734	9754	9745	9734	9724	9754	9745	9724	9724
		P23	4654	4677	4740	4654	4675	4667	4740	4675	4654
		P24	8979	8989	8979	8979	8979	8997	8989	8979	8979
30	3	P25	7131	7142	7131	7131	7130	7142	7131	7130	7130
		P26	14528	14478	14563	14528	14487	14478	14563	14487	14478
		P27	8098	8136	8047	8059	8074	8120	8054	8049	8047
		P28	14933	14973	14901	14933	14945	14906	14901	14908	14901
	5	P29	13396	13378	13463	13396	13374	13489	13463	13374	13374
		P30	25515	25536	25448	25515	25410	25536	25448	25428	25410
		P31	12163	12203	12163	12229	12204	12279	12163	12163	12163
		P32	24307	24369	24389	24334	24283	24391	24389	24283	24283
Best solution			21	17	22	17	24	16	18	26	32

Table 5.9 Solution results for TS heuristics for data set 1.

problem size		Problem No.	1	2	3	Average	Best solution
N	T						
6	3	P01	267	267	267	267	267
		P02	260	260	260	260	260
		P03	363	363	385	370	363
		P04	299	299	309	302	299
	5	P05	442	442	442	442	442
		P06	586	586	586	586	586
		P07	424	424	424	424	424
		P08	428	428	428	428	428
12	3	P09	1624	1624	1624	1624	1624
		P10	1973	1973	1973	1973	1973
		P11	1661	1661	1661	1661	1661
		P12	2097	2097	2097	2097	2097
	5	P13	2930	2930	2930	2930	2930
		P14	3701	3701	3701	3701	3701
		P15	2756	2756	2756	2756	2756
		P16	3364	3364	3364	3364	3364
20	3	P17	2758	2758	2758	2758	2758
		P18	5318	5318	5318	5318	5318
		P19	3038	3034	3034	3035	3034
		P20	5878	5869	5873	5873	5869
	5	P21	4586	4581	4554	4574	4554
		P22	9746	9736	9743	9742	9736
		P23	4654	4654	4654	4654	4654
		P24	8979	8979	8979	8979	8979
30	3	P25	7130	7131	7131	7131	7130
		P26	14528	14478	14478	14495	14478
		P27	8075	8087	8076	8079	8075
		P28	14939	14936	14913	14929	14913
	5	P29	13514	13383	13379	13425	13379
		P30	25523	25509	25484	25505	25484
		P31	12148	12148	12151	12149	12148
		P32	24200	24200	24232	24211	24200

Table 5.10 Solution results for the PTS heuristic for data set 1.

In Tables 5.11 - 5.14, the cutting plane algorithm solutions presented in Lacksonen and Ensore (1993) and the best tabu search heuristic solutions presented in Kaku and Mazzola (1997) are given under the “CP” and “TS-KM” columns, respectively. Also, the best HAS solutions presented in McKendall and Shang (2005) are given under the “HAS” column, and

the best SA heuristic solutions presented in McKendall *et al.* (2005) are given under the “SA” column. The best solutions obtained from either CP, TS-KM, HAS, or SA heuristic are given under the “Best found solution” column. In the last column, the percent deviation that the best solution obtained from the proposed heuristics is below the best found solution obtained from the heuristics presented in the literature is given under “% Dev” for each problem. In the last row, the number of the best solutions obtained from each heuristic is given. The bold numbers indicate the best objective function value (OFV) obtained for each test problem.

In Table 5.11, the results are shown for the test problems where $N = 6$. For test problems with $T = 3$ and $T = 5$ (P01-08), the TSbasic, TSall, and PTS heuristics obtained the best solution for all 8 test problems. Therefore, all the proposed heuristics performed equally well. Since the proposed TS heuristics, TS-KM, HAS, and SA heuristics obtained the best solutions for all 8 test problems, these heuristics are the preferred choice for this set of 8 test problems.

problem size		Problem No.	TSbasic	TSall	PTS	CP	TS-KM	HAS	SA	Best found solution	% Dev	
N	T											
6	3	P01	267	267	267	267	267	267	267	267	0	
		P02	260	260	260	260	260	260	260	260	0	
		P03	363	363	363	363	363	363	363	363	0	
		P04	299	299	299	299	299	299	299	299	0	
	5	P05	442	442	442	442	442	442	442	442	442	0
		P06	586	586	586	589	586	586	586	586	586	0
		P07	424	424	424	424	424	424	424	424	424	0
		P08	428	428	428	428	428	428	428	428	428	0
Best solution			8	8	8	7	8	8	8	8	8	

Table 5.11 Solution results for problems with $N = 6$ in data set 1.

In Table 5.12, the results are shown for the test problems where $N = 12$. For test problems with $T = 3$ and 5 (P09-16), both the TSall and PTS heuristics obtained the best solution for all 8 test problems. Therefore, both TSall and PTS performed equally well. Since TSall, PTS, TS-KM, HAS, and SA heuristics obtained the best solutions for all 8 test problems, these heuristics are the preferred choice for this set of 8 problems.

Pro. size		Problem No.	TSbasic	TSall	PTS	CP	TS-KM	HAS	SA	Best found solution	% Dev
N	T										
12	3	P09	1624	1624	1624	1624	1624	1624	1624	1624	0
		P10	1973	1973	1973	1973	1973	1973	1973	1973	0
		P11	1661	1661	1661	1661	1661	1661	1661	1661	0
		P12	2102	2097	2097	2097	2097	2097	2097	2097	0
	5	P13	2930	2930	2930	2930	2930	2930	2930	2930	0
		P14	3701	3701	3701	3726	3701	3701	3701	3701	0
		P15	2779	2756	2756	2756	2756	2756	2756	2756	0
		P16	3364	3364	3364	3364	3364	3364	3364	3364	0
Best solution			6	8	8	7	8	8	8	8	8

Table 5.12 Solution results for problems with $N = 12$ in data set 1.

In Table 5.13, the results are shown for the test problems where $N = 20$. For test problems with $T = 3$ and 5 (P17-24), TSbasic, TSall and PTS heuristics obtained the best solution for 6, 6 and 7 of the 8 test problems, respectively. Therefore, the PTS heuristic outperformed the other proposed heuristics (TSbasic and TSall). However, CP, TS-KM, HAS, and SA heuristics obtained the best solutions for 2, 4, 6, and 6 problems, respectively. Thus, the PTS heuristic outperformed all other heuristics.

problem size		Problem No.	TSbasic	TSall	PTS	CP	TS-KM	HAS	SA	Best found solution	% Dev
<i>N</i>	<i>T</i>										
20	3	P17	2758	2758	2758	2763	2758	2758	2758	2758	0
		P18	5318	5318	5318	5318	5318	5318	5318	5318	0
		P19	3034	3034	3034	3048	3056	3034	3034	3034	0
		P20	5873	5869	5869	5873	5903	5881	5873	5873	-0.068
	5	P21	4554	4573	4554	4581	4605	4575	4554	4554	0
		P22	9734	9724	9736	9825	9746	9724	9724	9724	0
		P23	4654	4675	4654	4654	4654	4654	4660	4654	0
		P24	8979	8979	8979	8985	8979	8979	8979	8979	0
Best solution			6	6	7	2	4	6	6	7	

Table 5.13 Solution results for problems with $N = 20$ in data set 1.

In Table 5.14, the results are shown for the test problems where $N = 30$. For test problems with $T = 3$ and 5 (P25-32), both the TSall and PTS heuristics obtained the best solution for 4 of the 8 test problems. Thus, both proposed heuristics performed equally well. Also, HAS and SA obtained the best solutions for 4 of the 8 test problems. Therefore, TSall, PTS, HAS, and SA heuristics are the preferred heuristics for these test problems.

problem size		Problem No.	TSbasic	TSall	PTS	CP	TS-KM	HAS	SA	Best found solution	% Dev
<i>N</i>	<i>T</i>										
30	3	P25	7131	7130	7130	7163	7130	7130	7130	7130	0
		P26	14528	14487	14478	14583	14478	14478	14478	14478	0
		P27	8098	8049	8075	8066	8115	8066	8070	8066	-0.211
		P28	14933	14908	14913	14940	14925	14925	14901	14901	0.047
	5	P29	13396	13374	13379	13719	13606	13374	13374	13374	0
		P30	25515	25428	25484	26027	25583	25521	25472	25472	-0.173
		P31	12163	12163	12148	12351	12163	12163	12170	12163	-0.123
		P32	24307	24283	24200	24409	24200	24200	24300	24200	0
Best solution			0	4	4	0	3	4	4	5	

Table 5.14 Solution results for problems with $N = 30$ in data set 1.

In summary, the TSall, PTS, HAS, and SA heuristics obtained the best solutions for 26 and 27, 26, and 26 of the 32 problems, respectively. Therefore, the PTS heuristic slightly out-performed the TSall, HAS, and SA heuristics with respect to solution quality. However, the PTS heuristic required multiple runs (3 runs) for each test problem, and the TSall heuristic performed only one run, since it is a deterministic heuristic. Therefore, the total run time for the PTS heuristic is 3 times the total run time of the TSall heuristic. Also, the results given under the HAS heuristic are the results obtained from running 3 different HASs with 3 runs each, and the results given under the SA heuristic are the results obtained from running 2 different SA heuristics with 5 runs each. Kaku and Mazzola (1997) gave the average computation times for their TS heuristic. The average computation time for the TS heuristic for the larger size problems (problems 29 – 32) was approximately 2 hours and 47 minutes on a Pentium 200 MHz PC. For the HAS heuristics, the average computation time was 20 minutes for each run (3 runs) on a Pentium IV 2.4 GHz PC. For the SA heuristics, the average computation time was 8.5 minutes for each run (5 runs) on a Pentium IV 2.4 GHz PC. However, the computation time for the TSall heuristic was 18 minutes on an AMD Athlon 2600+ 1.92 GHz PC, and the average computation time for the PTS heuristic was 18 minutes for each run (3 runs). Hence, the TSall heuristic may be the preferred heuristic with respect to solution quality and computational time.

5.3.2 Data Set from Balakrishnan and Cheng (2000)

First, the test problems were solved using the basic TS heuristic (TSbasic), and then each strategy and combination of strategies were added to this heuristic to measure their

impact as discussed in Section 5.3.1. Table 5.15 summarizes the results obtained by TSbasic, TSfre, TSdiver, TSinten, TSfrediver, TSfreinten, TSdiverinten, TSall. In the last column, the best solution obtained is given, and the bold numbers indicate the best objective function value (OFV) obtained from the heuristic for each test problem in this data set. In the last row, the number of the best solutions obtained from the TS heuristic with each combination of strategies is given. The TSbasic, TSfre, TSdiver, TSinten, TSfrediver, TSfreinten, TSdiverinten, and TSall heuristics obtained the best solution for 17, 22, 21, 22, 27, 26, 30 and 33 of the 48 test problems, respectively. Therefore, the proposed TS heuristic with all strategies is the preferred TS heuristic for the data set taken from Balakrishnan and Cheng (2000).

Table 5.16 summarizes the results obtained by the PTS heuristic. The result for each run and the average of the three runs for each of the test problems are given. In the last column, the best solution obtained from the PTS heuristic for the data set taken from Balakrishnan and Cheng (2000) is given. The bold numbers indicate the best objective function value (OFV) obtained from the PTS heuristic for each test problem.

Problem size		Problem No.	TSbasic	TSfre	TSdiver	TSinten	TSfrediver	TSfreinten	TSdiverinten	TS all	Best solution
N	T										
6	5	P01	106,419	106,419	106,419	106,419	106,419	106,419	106,419	106,419	106,419
		P02	104,834	104,834	104,834	104,834	104,834	104,834	104,834	104,834	104,834
		P03	104,520	104,320	104,320	104,320	104,320	104,320	104,320	104,320	104,320
		P04	106,399	106,399	106,399	106,399	106,399	106,399	106,399	106,399	106,399
		P05	105,737	105,628	105,628	105,628	105,628	105,628	105,628	105,628	105,628
		P06	103,985	103,985	103,985	103,985	103,985	103,985	103,985	103,985	103,985
		P07	106,447	106,439	106,439	106,439	106,439	106,439	106,439	106,439	106,439
		P08	106,152	103,771	103,771	103,771	103,771	103,771	103,771	103,771	103,771
	10	P09	214,313	214,313	214,313	214,313	214,313	214,313	214,313	214,313	214,313
		P10	212,134	212,134	212,134	212,134	212,134	212,134	212,134	212,134	212,134
		P11	207,987	208,060	208,673	207,987	208,060	207,987	207,987	207,987	207,987
		P12	212,530	212,530	212,530	212,530	212,530	212,530	212,530	212,530	212,530
		P13	210,906	210,906	210,906	210,906	210,906	210,906	210,906	210,906	210,906
		P14	209,932	209,932	210,176	209,932	209,932	209,932	209,932	209,932	209,932
		P15	214,252	214,252	214,252	214,252	214,252	214,252	214,252	214,252	214,252
P16	212,588	212,588	212,588	212,588	212,588	212,588	212,588	212,588	212,588		
15	5	P17	480,453	480,453	480,453	480,453	480,453	480,453	480,453	480,453	480,453
		P18	484,761	484,761	484,761	484,761	484,761	484,761	484,761	484,761	484,761
		P19	489,335	489,058	489,058	489,335	490,174	489,058	489,126	489,058	489,058
		P20	484,621	484,876	484,446	484,621	484,446	484,621	484,446	484,446	484,446
		P21	487,822	487,989	488,687	487,822	487,753	487,822	487,753	487,822	487,753
		P22	486,493	486,689	487,275	486,493	486,493	486,493	486,493	486,493	486,493
		P23	486,268	486,268	487,385	486,268	486,268	486,268	486,819	486,268	486,268
		P24	490,551	491,016	491,035	490,551	490,551	490,551	490,812	490,551	490,551
	10	P25	983,061	982,344	981,335	983,061	981,335	982,344	980,546	980,399	980,399
		P26	978,874	979,081	977,399	978,874	977,338	978,874	977,338	977,399	977,338
		P27	982,944	983,273	983,354	983,658	982,889	983,273	981,280	981,172	981,172
		P28	972,325	972,963	972,019	972,325	972,019	972,325	971,720	972,019	971,720
		P29	978,033	978,563	978,439	978,033	978,439	978,033	976,784	977,657	976,784
		P30	969,124	970,085	970,208	969,124	970,456	969,124	967,617	970,085	967,617
		P31	979,881	979,991	978,681	979,881	978,681	979,881	978,851	978,681	978,681
P32	985,105	985,370	984,177	985,105	983,882	985,105	983,076	984,177	983,076		
30	5	P33	576,269	575,429	574,876	576,269	576,057	575,429	573,941	574,876	573,941
		P34	569,119	567,969	571,215	567,995	572,234	567,969	569,592	567,969	567,969
		P35	573,930	571,639	572,895	572,556	573,647	571,639	572,292	571,639	571,639
		P36	565,637	565,831	567,246	565,637	567,412	565,637	565,859	564,725	564,725
		P37	556,946	556,243	556,230	556,946	556,719	556,243	555,807	556,230	555,807
		P38	565,559	567,009	565,731	565,559	564,867	565,559	565,300	565,559	564,867
		P39	574,278	568,115	576,468	574,278	574,747	568,115	570,813	568,376	568,115
		P40	573,873	574,046	574,387	573,873	572,658	573,194	573,474	573,194	572,658
	10	P41	1,160,941	1,161,455	1,158,836	1,159,896	1,158,777	1,160,941	1,162,490	1,158,836	1,158,777
		P42	1,160,273	1,161,056	1,159,281	1,158,432	1,159,680	1,158,432	1,160,644	1,159,281	1,158,432
		P43	1,158,212	1,155,632	1,156,360	1,158,212	1,154,210	1,154,490	1,156,744	1,152,362	1,152,362
		P44	1,149,047	1,144,948	1,143,078	1,145,943	1,145,468	1,149,047	1,146,639	1,143,078	1,143,078
		P45	1,127,721	1,127,606	1,124,831	1,126,362	1,125,238	1,126,362	1,122,947	1,124,091	1,122,947
		P46	1,143,559	1,143,181	1,140,450	1,143,211	1,142,611	1,143,559	1,143,688	1,140,450	1,140,450
		P47	1,150,130	1,148,017	1,146,150	1,148,732	1,146,921	1,148,732	1,144,980	1,146,150	1,144,980
		P48	1,166,646	1,165,096	1,165,803	1,166,327	1,164,303	1,165,413	1,161,914	1,161,426	1,161,426
Best solution			17	22	21	22	27	26	30	33	48

Table 5.15 Solution results for TS heuristics for data set 2.

Problem size		Problem No.	1	2	3	Average	Best solution
<i>N</i>	<i>T</i>						
6	5	P01	106,419	106,419	106,419	106,419	106,419
		P02	104,834	104,834	104,834	104,834	104,834
		P03	104,320	104,520	104,320	104,387	104,320
		P04	106,399	106,399	106,399	106,399	106,399
		P05	105,628	105,737	105,737	105,701	105,628
		P06	103,985	103,985	103,985	103,985	103,985
		P07	106,447	106,447	106,447	106,447	106,447
		P08	106,152	106,152	106,152	106,152	106,152
	10	P09	214,313	218,656	214,313	215,761	214,313
		P10	212,134	213,828	212,134	212,699	212,134
		P11	207,987	209,031	207,987	208,335	207,987
		P12	212,530	213,974	212,530	213,011	212,530
		P13	210,906	213,216	210,906	211,676	210,906
		P14	209,932	210,417	209,932	210,094	209,932
		P15	214,252	215,054	214,252	214,519	214,252
		P16	212,588	214,120	212,588	213,099	212,588
15	5	P17	480,453	480,497	480,453	480,468	480,453
		P18	484,799	484,761	484,799	484,786	484,761
		P19	489,265	489,335	489,335	489,312	489,265
		P20	484,621	485,436	484,621	484,893	484,621
		P21	488,128	487,822	487,753	487,901	487,753
		P22	487,426	487,619	486,493	487,179	486,493
		P23	486,268	487,578	487,414	487,087	486,268
		P24	492,151	490,551	492,015	491,572	490,551
	10	P25	981,077	980,906	982,140	981,374	980,906
		P26	979,102	980,276	978,815	979,398	978,815
		P27	983,898	983,988	985,007	984,298	983,898
		P28	973,970	972,755	972,019	972,915	972,019
10	P29	978,479	978,297	977,534	978,103	977,534	
	P30	968,077	970,472	967,617	968,722	967,617	
	P31	979,928	980,613	979,513	980,018	979,513	
	P32	985,649	985,947	985,105	985,567	985,105	
30	5	P33	576,145	574,577	576,323	575,682	574,577
		P34	571,051	567,691	569,918	569,553	567,691
		P35	573,732	573,307	574,075	573,705	573,307
		P36	565,849	567,592	566,991	566,811	565,849
		P37	557,098	557,256	557,788	557,381	557,098
		P38	566,335	565,670	566,594	566,200	565,670
		P39	573,918	572,701	571,085	572,568	571,085
		P40	574,854	575,176	575,274	575,101	574,854
	10	P41	1,161,089	1,160,196	1,163,852	1,161,712	1,160,196
		P42	1,159,088	1,160,113	1,165,069	1,161,423	1,159,088
		P43	1,159,029	1,157,350	1,155,280	1,157,220	1,155,280
		P44	1,148,298	1,147,598	1,146,881	1,147,592	1,146,881
		P45	1,125,845	1,128,677	1,125,429	1,126,650	1,125,429
		P46	1,145,100	1,145,245	1,144,625	1,144,990	1,144,625
		P47	1,148,081	1,146,200	1,147,234	1,147,172	1,146,200
		P48	1,165,610	1,163,528	1,167,172	1,165,437	1,163,528

Table 5.16 Solution results for the PTS heuristic for data set 2.

Tables 5.17 - 5.19 summarize the results obtained by the proposed heuristics (basic TS (TSbasic), the TS heuristic with all strategies (TSall), and the PTS heuristic (PTS)) as well as the heuristics presented by Baykasoglu and Gindy (2001) using a SA heuristic, Balakrishnan *et al.* (2003) using GA algorithm (GA), Erel *et al.* (2003) using dynamic programming approaches (DP), McKendall and Shang (2005) using hybrid ant systems (HAS), and McKendall *et al.* (2005) using SA heuristics (SA). The proposed TS heuristics are compared to the SA heuristic presented by Baykasoglu and Gindy (2001) since the results (SA_EG) presented in Erel *et al.* (2003) by using Baykasoglu and Gindy (2001) SA heuristic are competitive in test problems with $N = 15$. However, the solutions have been corrected and are available at their website (Erel, 2005). Also, the proposed TS heuristics are compared to the GA heuristic presented in Balakrishnan *et al.* (2003), since this heuristic gives the best GA results for this set of test problems. Furthermore, the proposed heuristics are also compared to DP approaches, HAS and SA heuristics, since these techniques perform well on this data set.

In Tables 5.17 - 5.19, the best DP solutions and the best SA heuristic solutions presented in Erel *et al.* (2003) are given under the “DP” and “SA_EG” columns, respectively. It is important to note that the corrections obtained for SA_EG from Erel (2005) are given in the parentheses. Also, the best GA heuristic solutions presented in Balakrishnan *et al.* (2003), the best HAS solutions presented in McKendall and Shang (2005), and the best SA heuristic solutions presented in McKendall *et al.* (2005) are given under “GA”, “HAS” and “SA” columns, respectively. The best solutions obtained from either DP, SA_EG, GA, HAS, or SA heuristic are given under the “Best found solution” column. In the last column, the percent deviation that the best solution obtained from the proposed heuristics is below the best found solution obtained from the heuristics presented in the literature is given under “% Dev” for

each problem. In the last row, the number of the best solutions obtained from each heuristic is given. The bold numbers indicate the best objective function value (OFV) obtained for each test problem.

In Table 5.17, the results are shown for the test problems where $N = 6$. For test problems with $T = 5$ and 10 (P01-16), TSbasic, TSall, and PTS heuristics obtained the best solution for 12, 16, and 14 of the 16 test problems, respectively. Therefore, the TSall heuristic outperformed the other proposed heuristics (TSbasic and PTS). Since TSall, HAS, and SA heuristics obtained the best solutions for all 16 test problems, these heuristics are the preferred choice for this set of 16 problems.

Pro. size		Problem No.	TSbasic	TS all	PTS	DP	SA_EG	GA	HAS	SA	Best found solution	%Dev
N	T											
6	5	P01	106,419	106,419	106,419	106,419	106,419	106,419	106,419	106,419	106,419	0
		P02	104,834	104,834	104,834	104,834	104,834	104,834	104,834	104,834	104,834	0
		P03	104,520	104,320	104,320	104,320	104,320	104,320	104,320	104,320	104,320	0
		P04	106,399	106,399	106,399	106,399	106,399	106,515	106,399	106,399	106,399	0
		P05	105,737	105,628	105,628	105,628	105,628	105,628	105,628	105,628	105,628	0
		P06	103,985	103,985	103,985	103,985	103,985	104,053	103,985	103,985	103,985	0
		P07	106,447	106,439	106,447	106,447	106,439	106,439	106,439	106,439	106,439	0
		P08	106,152	103,771	106,152	103,771	103,771	103,771	103,771	103,771	103,771	0
	10	P09	214,313	214,313	214,313	214,313	214,313	214,313	214,313	214,313	214,313	0
		P10	212,134	212,134	212,134	212,134	212,134	212,134	212,134	212,134	212,134	0
		P11	207,987	207,987	207,987	207,987	207,987	207,987	207,987	207,987	207,987	0
		P12	212,530	212,530	212,530	212,741	212,747	212,741	212,530	212,530	212,530	0
		P13	210,906	210,906	210,906	211,022	211,072	210,944	210,906	210,906	210,906	0
		P14	209,932	209,932	209,932	209,932	209,932	210,000	209,932	209,932	209,932	0
		P15	214,252	214,252	214,252	214,252	214,438	215,452	214,252	214,252	214,252	0
		P16	212,588	212,588	212,588	212,588	212,588	212,588	212,588	212,588	212,588	0
Best solution			12	16	14	13	13	10	16	16	16	

Table 5.17 Solution results for problems with $N = 6$ in data set 2.

In Table 5.18, the results are shown for the test problems where $N = 15$. For test problems with $T = 5$ and 10 (P17-32), TSbasic, TSall and PTS heuristics obtained the best solutions for 5, 8, and 7 of the 16 test problems, respectively. Therefore, the TSall heuristic performed better than the other proposed heuristics (TSbasic and PTS). Also, HAS and SA heuristics obtained the best solutions for 5 and 8 of the 16 test problems, respectively. Therefore, TSall and SA heuristics are the preferred choice for this set of 16 problems.

Pro. size		Pro. No.	TSbasic	TS all	PTS	DP	SA_EG	GA	HAS	SA	Best found solution	%Dev
N	T											
15	5	P17	480,453	480,453	480,453	482,123	481,378 (481,738)	484,090	480,453	480,453	480,453	0
		P18	484,761	484,761	484,761	485,702	478,816 (485,167)	485,352	484,761	484,761	484,761	0
		P19	489,335	489,058	489,265	491,310	487886 (*)	489,898	488,748	488,748	488,748	0.063
		P20	484,621	484,446	484,621	486,851	481,628 (485,862)	484,625	484,446	484,405	484,405	0.008
		P21	487,822	487,822	487,753	491,178	484,177 (489,304)	489,885	487,722	487,882	487,722	0.006
		P22	486,493	486,493	486,493	489,847	482321 (488,452)	488,640	486,685	487,147	486,685	-0.039
		P23	486,268	486,268	486,268	489,155	485,384 (487,576)	489,378	486,853	486,779	486,779	-0.105
		P24	490,551	490,551	490,551	497,577	489,072 (493,030)	500,779	491,016	490,812	490,812	-0.053
	10	P25	983,061	980,399	980,906	983,070	982298 (*)	987,887	980,351	979,468	979,468	0.095
		P26	978,874	977,399	978,815	983,826	973,179 (982,714)	980,638	978,271	978,065	978,065	-0.068
		P27	982,944	981,172	983,898	988,635	985,364 (988,465)	985,886	978,027	982,396	978,027	0.322
		P28	972,325	972,019	972,019	976,456	974,994 (976,456)	976,025	974,694	972,797	972,797	-0.080
		P29	978,033	977,657	977,534	982,893	975,498 (982,191)	982,778	979,196	977,188	977,188	0.035
		P30	969,124	970,085	967,617	974,436	968,323 (973,199)	973,912	971,548	967,617	967,617	0
		P31	979,881	978,681	979,513	982,790	977,410 (*)	982,872	980,752	979,114	979,114	-0.044
		P32	985,105	984,177	985,105	988,584	985,041 (988,304)	987,789	985,707	983,672	983,672	0.051
Best solution		5	8	7	0	0	0	5	8	10		

(*) The corrected solution for SA_EG was worse than the solution obtained using DP.

Table 5.18 Solution results for problems with $N = 15$ in data set 2.

In Table 5.19, the results are shown for the test problems where $N = 30$. For test problems with $T = 5$ and 10 (P33-48), TSbasic, TSall and PTS heuristics obtained the best solutions for 1, and 11 and 3 of the 16 test problems, respectively. Therefore, the TSall heuristic outperformed the other proposed heuristics (TSbasic and PTS). Also, HAS and SA heuristic obtained the best solution for 1 and 2 of the 16 test problems, respectively, and DP, SA_EG and GA obtained the best solution for 0 of the 16 test problems. Clearly, TSall heuristic outperformed all of the other heuristics for this set of 16 problems.

Pro. size		Problem No.	TSbasic	TS all	PTS	DP	SA_EG	GA	HAS	SA	Best found solution	%Dev
N	T											
30	5	P33	576,269	574,876	574,577	579,741	583,081	578,689	576,886	576,039	576,039	-0.254
		P34	569,119	567,969	567,691	570,906	573,965	572,232	570,349	568,095	568,095	-0.071
		P35	573,930	571,639	573,307	577,402	577,787	578,527	576,053	573,739	573,739	-0.366
		P36	565,637	564,725	565,849	569,596	572,139	572,057	566,777	566,248	566,248	-0.269
		P37	556,946	556,230	557,098	561,078	563,503	559,777	558,353	558,460	558,353	-0.380
		P38	565,559	565,559	565,670	567,154	570,905	566,792	566,792	566,077	566,077	-0.092
		P39	574,278	568,376	571,085	568,196	571,499	567,873	567,131	567,131	567,131	0.220
		P40	573,873	573,194	574,854	575,273	581,614	575,720	575,280	573,755	573,755	-0.098
	10	P41	1,160,941	1,158,836	1,160,196	1,171,178	1,174,815	1,169,474	1,166,164	1,163,222	1,163,222	-0.377
		P42	1,160,273	1,159,281	1,159,088	1,169,138	1,173,015	1,168,878	1,168,878	1,161,521	1,161,521	-0.209
		P43	1,158,212	1,152,362	1,155,280	1,165,525	1,166,295	1,166,366	1,166,366	1,156,918	1,156,918	-0.394
		P44	1,149,047	1,143,078	1,146,881	1,152,684	1,154,196	1,154,192	1,148,202	1,145,918	1,145,918	-0.248
		P45	1,127,721	1,124,091	1,125,429	1,128,136	1,140,116	1,133,561	1,128,855	1,126,432	1,126,432	-0.208
		P46	1,143,559	1,140,450	1,144,625	1,143,824	1,158,227	1,145,000	1,141,344	1,145,146	1,141,344	-0.078
		P47	1,150,130	1,146,150	1,146,200	1,142,494	1,157,505	1,145,927	1,140,773	1,140,744	1,140,744	0.474
		P48	1,166,646	1,161,426	1,163,528	1,167,163	1,177,565	1,168,657	1,166,157	1,161,437	1,161,437	-0.001
Best solution			1	11	3	0	0	0	1	2	2	

Table 5.19 Solution results for problems with $N = 30$ in data set 2.

In summary, the TSbasic, TSall, PTS, DP, SA_EG, GA, HAS, and SA heuristics obtained the best solutions for 18, 35, 24, 13, 13, 10, 22 and 26 of the 48 problems, respectively. Therefore, the TSall heuristic clearly outperformed all the other heuristics for this data set with respect to solution quality. In addition, the PTS heuristic required multiple runs (3 runs) for each test problem, the results given under the DP heuristic are the best results obtained from running 8 heuristics, the results given under the SA_EG heuristic are the best results obtained from running 2 different SA settings with 5 runs each, the results given under the GA heuristic are the best results obtained from 2 different initial solutions, the results given under the HAS heuristic are the best results obtained from running 3 different HASs with 3 runs each, and the results given under the SA heuristic are the best results obtained from running 2 different SA heuristics with 5 runs each. Furthermore, the 30 department, 10 period problems were solved in 12 minutes for TSall and in an average of 12 minutes for PTS on an AMD Athlon 2600+ 1.92 GHz PC. For the DP approaches, the average computational time for the best solutions was between 30 minutes and 2 hours on an Ultra Enterprise sever operating under Solaris 7 at 250 MHz. For the SA_EG heuristic, the average computational time was 18.5 hours on an Ultra Enterprise sever operating under Solaris 7 at 250 MHz, and the average computational time was 16.7 minutes on DEC Alpha machines for the GA heuristic. For the HAS heuristic, the average computational time was 45 minutes for each run on a Pentium IV 2.4 GHz PC. Last, for the SA heuristic, the average computational time was 7.8 minutes for each run on a Pentium IV 2.4 GHz PC. Hence, the TSall heuristic is the preferred heuristic with respect to solution quality and computational time.

CHAPTER 6

CONCLUSION

6.1 Summary of Research

In this research, the dynamic facility layout problem is considered and tabu search heuristics are presented to solve the DFLP. The proposed tabu search heuristics are: a simple tabu search heuristic which is a straightforward implementation of the tabu search heuristic (TSbasic), a tabu search heuristic with frequency-based memory as well as diversification and intensification strategies (TSall), and a probabilistic tabu search heuristic (PTS).

The proposed tabu search heuristics were tested using two data sets. The first data set was taken from Lacksonen and Enscore (1993) with 32 test problems, and the second data set was taken from Balakrishnan and Cheng (2000) with 48 test problems. The probabilistic tabu search (PTS) heuristic obtained the best solutions for 27 of the 32 test problems presented by Lacksonen and Enscore (1993) and performed slightly better than the tabu search heuristic with strategies (TSall). Also, PTS slightly performed better than two of the best heuristics presented in the literature for this data set. The tabu search heuristic with strategies (TSall) obtained the best solutions for 35 of the 48 test problems and out-performed all the other heuristics for the data set presented by Balakrishnan and Cheng (2000). Therefore, the proposed TS heuristics (more specifically TSall and PTS) performed better than all of the other heuristics with respect to solution quality for the two data sets presented in the literature.

6.2 Recommendations for Future Research

The following recommendations are given for future research:

1. Consider rearrangement costs with respect to periods and distances between the locations of exchanged departments.
2. Use other criteria to free the departments in the intensification strategy to search the neighborhoods of good solutions more in depth.
3. Consider solving the DFLP by using a hybrid technique that combines deterministic and stochastic heuristics (e.g. TS and GA or TS and SA) to obtain better results.
4. Consider developing construction algorithms to construct good initial solutions.

REFERENCES

- Ahuja, R.K., Orlin, J.B., and Tiwari, A., 2000, "A Greedy Genetic Algorithm for the Quadratic Assignment Problem," *Computers and Operations Research*, Vol. 27, pp. 917-934.
- Armour, G.C. and Buffa, E.S., 1963, "A Heuristic Algorithm and Simulation Approach to Relative Location of Facilities," *Management Science*, Vol.9, pp.294-309.
- Balakrishnan, J. and Cheng, C.H., 1998, "Dynamic Layout Algorithm: A State-of-the-art Survey," *Omega*, Vol. 26, No. 4, pp. 507-521.
- Balakrishnan, J., and Cheng, C.H., 2000, "Genetic Search and the Dynamic Layout Problem," *Computers & Operations Research*, Vol.27, pp. 587-593.
- Balakrishnan, J., Cheng, C.H., and Conway, D.G., 2000, "An Improved Pair-wise Exchange Heuristic for the Dynamic Plant Layout Problem," *International Journal of Production Research*, Vol. 38, No.13 pp.3067-3077.
- Balakrishnan, J., Cheng, C.H., Conway, D.G. and Lau, C. M., 2003, "A Hybrid Genetic Algorithm for the Dynamic Plant Layout Problem," *International Journal of Production Economics*, Vol. 86, pp. 107-120.
- Balakrishnan, J., Jacobs, F.R., and Venkataramanan, M.A., 1992, "Solution for the Constrained Dynamic Facility Layout Problem," *European Journal of Operational Research*, Vol. 57, pp. 280-286.
- Ballou, R., 1968, "Dynamic Warehouse Location Analysis," *Journal of Marketing Research*, Vol. 5, No. 3, pp.271-276.
- Baykasoglu, A. and Gindy, N.N.Z., 2001, "A Simulated Annealing Algorithm for Dynamic Layout Problem," *Computers & Operations Research*, Vol.28, pp. 1403-1426.
- Bazarra, M.S. and Sherali, M.D., 1980, "Bender's Partitioning Scheme Applied to A New Formulation of the Quadratic Assignment Problem," *Naval Research Logistics Quarterly*, Vol. 27, No. 1, pp. 29-41.

Bozer, Y.A., Meller, R.D., and Erlebacher, S. J., 1994, "An Improvement-type Layout Algorithm for Single and Multiple Floor Facilities." *Management Science*, Vol. 40, No.7, pp. 918-932.

Buffa, A.R., Armour, G.C., and Vollman, T.E., 1964, "Allocating Facilities with CRAFT," *Harvard Business Review*, Vol. 42, pp. 136-158.

Burkard, R.E. and Bonniger, T., 1983, "A Heuristic for Quadratic Boolean Problems with Applications to Quadratic Assignment Problems," *European Journal of Operational Research*, Vol. 13, pp.374-386.

Burkard, R.E. and Rendl, F., 1984, "A Thermodynamically motivated Simulation Procedure for Combinatorial Optimization Problems," *European Journal of Operational Research*, Vol. 17, pp. 169-174.

Chiang, W. and Chiang, C., 1998, "Intelligent Local Search Strategies for Solving Facility Layout Problems with the Quadratic Assignment Problem Formulation," *European Journal of Operational Research*, Vol. 106, pp. 457-488.

Chiang, W. and Kouvelis, P., 1996, "An Improved Tabu Search Heuristic for Solving Facility Layout Design Problems," *International Journal of Production Research*, Vol. 34, No. 9. pp. 2565-2585.

Conway, D.G. and Venkataramanan, M.A., 1994, "Genetic Search and the Dynamic Facility Layout Problem," *Computers & Operations Research*, Vol. 21, No. 8, pp.955-960.

Erel, E., 2005, "Dynamic Layout Problem," <http://www.bikent.edu.tr/~erel/dynlayout.html>.

Erel, E., Ghosh, J.B., and Simon, J.T., 2003, "New Heuristic for the Dynamic Layout Problem," *Journal of the Operational Research Society*, Vol. 54, pp. 1275-1282.

Fleurent, C. and Ferland, J.A., 1994, "Genetic Hybrids for the Quadratic Assignment Problem," *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, Vol. 16, pp. 173-188.

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

Gambardella, L.M., Taillard, E.D., and Dorigo, M., 1999, "Ant Colonies for the Quadratic Assignment Problem," *Journal of Operational Research Society*, Vol. 50, pp. 167-176.

Gilmore, P.C., 1962, "Optimal and Suboptimal Algorithms for the Quadratic Assignment Problem," *Journal of the Society for the Industrial and Applied Mathematics*, Vol. 10, pp. 305-313.

Glover, F., 1986, "Future Paths for Inter Programming and Links to Artificial Intelligence," *Computers and Operations Research*, Vol.1, No. 3, pp. 533-549.

Glover, F., 1989, "Tabu Search – Part I.," *ORSA Journal on Computing*, Vol. 1, No. 3, pp. 190-206.

Glover, F., 1990, "Tabu Search – Part II.," *ORSA Journal on Computing*, Vol. 2, No. 1, pp. 4-32.

Golany, B. and Rosenblatt, M.J., 1989, "A heuristic Algorithm for the Quadratic Assignment Formulation to the Plant Layout Problem," *International Journal of Production Research*, Vol. 27, No. 2, pp. 293-308.

Gomory, R.E. and Hu, T.C., 1961, "Multi-Terminal Network Flows," *SIAM Journal*, Vol. 9, pp. 551-570.

Heragu, S.S., 1992, "Experimental Analysis of Simulated Annealing Based Algorithms for the Layout Problem," *European Journal of Operational Research*, Vol. 57, pp. 190-202.

Heragu, S.S. and Kusiak, A., 1991, "Efficient Models for the Facility Layout Problem," *European Journal of Operational Research*, Vol. 53, pp. 1-13.

Kaku, B.K. and Mazzola, J.B., 1997, "A Tabu-Search Heuristic for the Dynamic Plant Layout Problem," *Inform Journal on Computing*, Vol. 9, No. 4, pp. 374-384.

Kaku, B.K. and Thompson, G.L. 1986, "An Exact Algorithm for the General Quadratic Assignment Problem," *European Journal of Operational Research*, Vol. 106, pp. 457-488.

Kaku, B.K., Thompson, G.L., and Morton, E.T., 1991, "A Hybrid Heuristic for the Facilities Layout Problem," *Computers & Operations Research*, Vol. 21, No. 8, pp. 885-893.

Kelly, J.P., Laguna, M., and Glover, F. 1994, "A Study of Diversification Strategies for the Quadratic Assignment Problem." *Computers & Operations Research*, Vol. 21, No. 8, pp. 885-893.

Koopmans, T.C. and Beckman, M., 1957, "Assignment Problems and the Location of Economic Activities," *Econometrical*, Vol. 25, No. 1, pp. 53-76.

Kouvelilis, P., Kurawarwala, A. A., and Gutierrez, G.J., 1992, "Algorithms for Single and Multiple Period Layout Planning for Manufacturing Systems," *European Journal of Operational Research*, Vol. 63, pp.287-303.

Kuppusamy, S., 2001, "Simulated Annealing Heuristics for the Dynamic Facility Layout Problem", *Master Thesis*, West Virginia University.

Kusiak, A. and Heragu, S.S., 1987, "The Facility Layout Problem," *European Journal of Operational Research*, Vol. 29, pp.229-251.

Lacksonen, T.A. and Enscore, E.E., 1993, "Quadratic Assignment Algorithms for the Dynamic Layout Problem," *International Journal of Production Research*, Vol. 31, No. 3, pp. 503-517.

Lawler, E.L., 1963, "The Quadratic Assignment Problem," *Management Science*, Vol. 9, No. 4, pp. 568-599.

Lim, A., Rodrigues, B., Xiao, F., and Zhu, Y., 2004, "Crane Scheduling with Spatial Constraints," *Naval Research Logistics*, Vol. 51, pp. 386-406.

McKendall, A.R., Noble, J.S., and Klein, C.M., 1999, "Facility Layout of Irregular-shaped Departments Using a Nested Approach," *International Journal of Production Research*, Vol. 37, No. 13, pp. 2895-2914.

McKendall, A.R. and Shang, J., 2005, "Hybrid Ant Systems for the Dynamic Facility Layout Problem," *Computers and Operations Research*, to appear.

McKendall, A.R., Shang, J. and Kuppusamy, S., 2005, "Simulated Annealing Heuristics for the Dynamic Facility Layout Problem," *Computers and Operations Research*, to appear.

Meller, R.D. and Gau, K.Y., 1996, "The Facility Layout Problem: Recent and emerging trends and Perspectives," *Journal of Manufacturing Systems*, Vol. 15, No. 5, pp. 351-366.

Nugent, C.E., Vollman, T.E. and Ruml, J. 1968, "An Experimental Comparison of Techniques for the Assignment of Facilities to Locations," *Operations Research*, Vol. 16, pp. 229-256.

Pardalos, P.H. and Crouse, J.V., 1989, "A Parallel Algorithm for the QAP," *Proceedings of the 1989 Supercomputer Conference*, ACM Press, New York, pp.351-360.

Rosenblatt, M.J., 1986, "The Dynamics of Plant Layout," *Management Science*, Vol. 32, No. 1, pp. 76-86.

Shang, J., 2002, "Ant Colony Heuristics for the Dynamic Facility Layout Problem," *Master Thesis*, West Virginia University.

Simchi-Levi, D., Kaminsky, P., and Simchi-Levi, E., 2003, "*Designing & Managing The Supply Chain Concepts, Strategies & Case Studies*," NY: McGraw-Hill/Irwin.

Skorin-Kapov, J., 1990, "Tabu Search Applied to the Quadratic Assignment Problem", *ORSA Journal on Computing*, Vol. 2, No. 1, pp. 33-45.

Skorin-Kapov, J., 1994, "Extensions of a Tabu Search Adaptation to the Quadratic Assignment Problem", *Computers and Operations Research*, Vol. 21, No. 8, pp.855-865.

Suresh, G., Vinod, V.V., and Sahu, S., 1995, "A Genetic Algorithm for Facility Layout," *International Journal of Production Research*, Vol. 33, No. 12, pp. 3411-3423.

Sweeney, D.S. and Tatham, R. L., 1976, "An Improved Long-Run Model for Multiple Warehouse Location," *Management Science*, Vol. 22, No. 7, pp. 748-758.

Taillard, E., 1991, "Robust Tabu Search for Quadratic Assignment," *Parallel Computing*, Vol. 17, pp. 443-455.

Tate, D.E. and Smith, A.E., 1995, "A Genetic Approach to the Quadratic Assignment Problem," *Computers and Operations Research*, Vol. 22, pp. 73-83.

Tompkins, J.A. and Reed, R. JR., 1976, "An Applied Model for the Facilities Design Problem," *International Journal of Production Research*, Vol. 14, No. 5, pp. 583-595.

Tompkins, J.A., White, J.A., Bozer, Y.A., Frazelle, E.H., Tanchoco, J.M.A., and Trevino, J., 2003, "*Facilities Planning*," NY: John Wiley & Sons, Inc.

Urban, T.L., 1993, "A Heuristic for the Dynamic Facility Layout Problem," *IIE Transactions*, Vol. 25, No. 4, pp. 57-63.

Wilhelm, M.R. and Ward, T.L., 1987, "Solving the Quadratic Assignment Problem by Simulated Annealing," *IIE Transactions*, Vol. 19, pp. 107-119.