MODELLING AND ANALYSIS OF MULTI-ROW LAYOUT PROBLEM

SUOAD Y. ALI EL MAGSSABI

FEBRUARY 2018

MODELLING AND ANALYSIS OF MULTI-ROW LAYOUT PROBLEM

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OFÇANKAYA UNIVERSITY

BY

SUOAD Y. ALI EL MAGSSABI

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
INDUSTRIAL ENGINEERING

FEBRUARY 2018

Title of the Thesis: **MODELLING AND ANALYSIS OF MULTI-ROW LAYOUT PROBLEM**

Submitted by **SUOAD Y. ALI EL MAGSSABI**

Approval of the Graduate School of Natural and Applied Sciences, Çankaya University

Prof. Dr. Can ÇOĞUN
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

Assoc. Prof. Dr. Ferda Can ÇETİNKAYA
Head of Department

This is to certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.

Asst. Prof. Dr. S. Hakan ÖZAKTAŞ
Co-Supervisor

Asst. Prof. Dr. Nureddin KIRKAVAK
Supervisor

Examination Date : 07 February, 2018

Examining Committee Members

Assoc. Prof. Dr. Mustafa Alp ERTEM (Çankaya University)

Assoc. Prof. Dr. Uğur BAÇ (Atılım University)

Asst. Prof. Dr. Nureddin KIRKAVAK (Çankaya University)

STATEMENT OF NON-PLAGIARISM

I confirm that all information provided in this thesis is taken and presented according to the academic and ethical conduct which requires the referencing and citation of any material that are not original of this thesis.

Name, Last Name	: Suoad Y. Ali El MAGSSABI
Signature	: سعاد
Date	: 07.02.2018

ABSTRACT

MODELLINGAND ANALYSIS OF MULTI-ROW LAYOUT PROBLEM

SUOAD Y. ALI EL MAGSSABI

M.S., Department of Industrial Engineering

Supervisor: Asst. Prof. Dr. Nureddin KIRKAVAK

February 2018, 81 pages

In this study, we considered multi-row,cost-distance objective, rectangular non-equal departmental areasto formulate a layout problem.According to the structural property of the factory, there are two types of facility layout problems, one of them is single-row and the other one is multi-row layout problem. If the total departmental area requirement is large, then it becomes necessary to formulate the layout problem asa multi-row model.In this thesis, we formulate a multi-row layout problem in order to analyzeand compare alternative solution techniques over various layout problems in several experiments.This study is applicable for problems where either product or process type layout exist in the factory. In the formulation, there are a number of rows in the layout and positions within a row, so that the mathematical model will assign the departments to one of these positions. For this reason, the assignment process shows both discrete and continuous characteristics, to behybrid.The solutions obtained using an optimal seeking solution technique(GAMS software) is to be compared with a permutation based enumeration technique and a limited sampling of random permutations. Although, the best solutions are obtained using time-limited GAMS software, the solutions obtained from permutation based and random sampling heuristic techniques are not statistically worse since the objective behaves so flat around the optimal point as we conclude that it is robust.

Keywords:GAMS software, facility layout problem, permutation based methods, random sampling methods, t-test.

ÖZET

ÇOK HOLLÜYERLEŞİM DÜZENLEMESİPROBLEMİNİN MODELLENMESİ VE ANALİZİ

SUOAD Y. ALI EL MAGSSABI

M.S., Endüstri Mühendisliği Bölümü

Danışman: Yrd. Doç. Dr. Nureddin KIRKAVAK

Şubat 2018, 81 sayfa

Bu çalışmada, maliyet-mesafe tabanlı hedef fonksiyonlu, dikdörtgen şeklinde eşit olmayanalan gereksinimlerini birden çok hole yerleştirecek bir tesis yerleşim probleminiformüle etmeyi göz önüne aldık. Fabrikaların yapısal özelliklerine göre iki tür yerleşim düzenlemesi problemi vardır; bunlardan biri tek diğeri çok hollü yerleşim düzenlemesi problemidir. Toplam yerleşim alanı gereksinimi büyükse, bu tip problemleri çok hollü olarak formüle etmek gerekir. Bu tezde, çeşitli yerleşim problemleri üzerinde tasarlanan deneyler ile alternatif çözüm tekniklerinin analizini yapmak ve karşılaştırmak için çok hollü bir yerleşim problemi modelledik. Bu çalışma, ürün veya işlem tipi yerleşim düzeninin bulunduğu fabrikalardaki problemler için uygulanır. Bu formülasyonda, yerleşim alanında holler,ve her hol içerisinde de konumlar bulunmaktadır, böylece matematiksel model bölümleri bu konumlardan birine atayacaktır. Bu nedenle, konumlara atama süreci hem ayrık ve hem de sürekli özellikler göstermekte olduğundan melez bir yapıdadır. Optimal çözüm arayış tekniği (GAMS yazılımı) kullanılarak elde edilen çözümlerle, permütasyon tabanlı sıralama ve rasgele oluşturulan permütasyonlardan örnekleme yapan tekniklerin çözümleri karşılaştırılacaktır. En iyi çözümlerin, zamanı sınırlı GAMS yazılımı kullanılarak elde edilmiş olmasına rağmen, permütasyon tabanlı ve rasgele örnekleme yapan yaklaşık çözüm teknikleride istatistiksel açıdan kötü sonuçlar vermemiştir. Çünkü, hedef fonksiyonuen iyi çözümün etrafında o kadar düzdür ki, sağlam (robust) olduğu sonucuna varırız.

Anahtar Kelimeler: GAMS yazılımı, tesis yerleşim problemi, permütasyon esaslı yöntemler, rassal örnekleme yöntemleri, t-testi.

ACKNOWLEDGMENTS

I would like to express my special thanks of gratitude to Asst. Prof. Dr. Nureddin KIRKAVAK for his supervision, and helping in doing a lot of research to do this wonderful thesis on the topic “MODELLING AND ANALYSIS OF MULTI-ROW LAYOUT PROBLEM”.

And special thanks to Asst. Prof. Dr. Hakan ÖZAKTAŞ for his support to me all the time. I would also like to thank my mother and my brother Salem for their support to complete my study in Turkey.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS

FLP	Facility Layout Problem
MRFLP	Multi-Row Facility Layout Problem
PERM	Enumeration of Permutations Based Method
QAP	Quadratic Assignment Problem
RAND	Random Sampling of Permutations Based Method
SRFLP	Single-Row Facility Layout Problem

CHAPTER

1. INTRODUCTION

Arranging the layout of a facility is a task of which all the related units, departments, facilities, workstations or machines in manufacturing, production or service sectors are organized in a manner that optimizes some objective. Moreover, there are many factorsto beconsidered when formulating and solving a facility layout problem (FLP). However, it is quite challenging to define the FLP due to its various types and solutions(Drira, Pierreval, & Hajri-Gabouj, 2007). Furthermore, there are three main factors to consider when encountering a facility layout problem which are:

1. The features of the facility, which includes the production system, the material handling and the shape of the departments.
2. The type of the FLP in terms of “formulation, objectives and constraints”.
3. The solution technique.

Drira, et al. (2007) provides a tree which shows the many factors that can affect the FLP as shown in Figure 1.

However, the main objectives of the FLP in various models are:

1. Configuring the departments in order to reduce transportation of material between them to the minimum as possible, Koopmans and Beckmann (1957).
2. Finding a non-overlapping arrangement of a certain number of rectangular shaped departments within a rectangular shaped facility by minimizing distance as possible, Meller, Narayanan and Vance(1999).
3. Allocating the available space of the facility to the required number and areas of the departments as possible, Azadivar and Wang (2000).

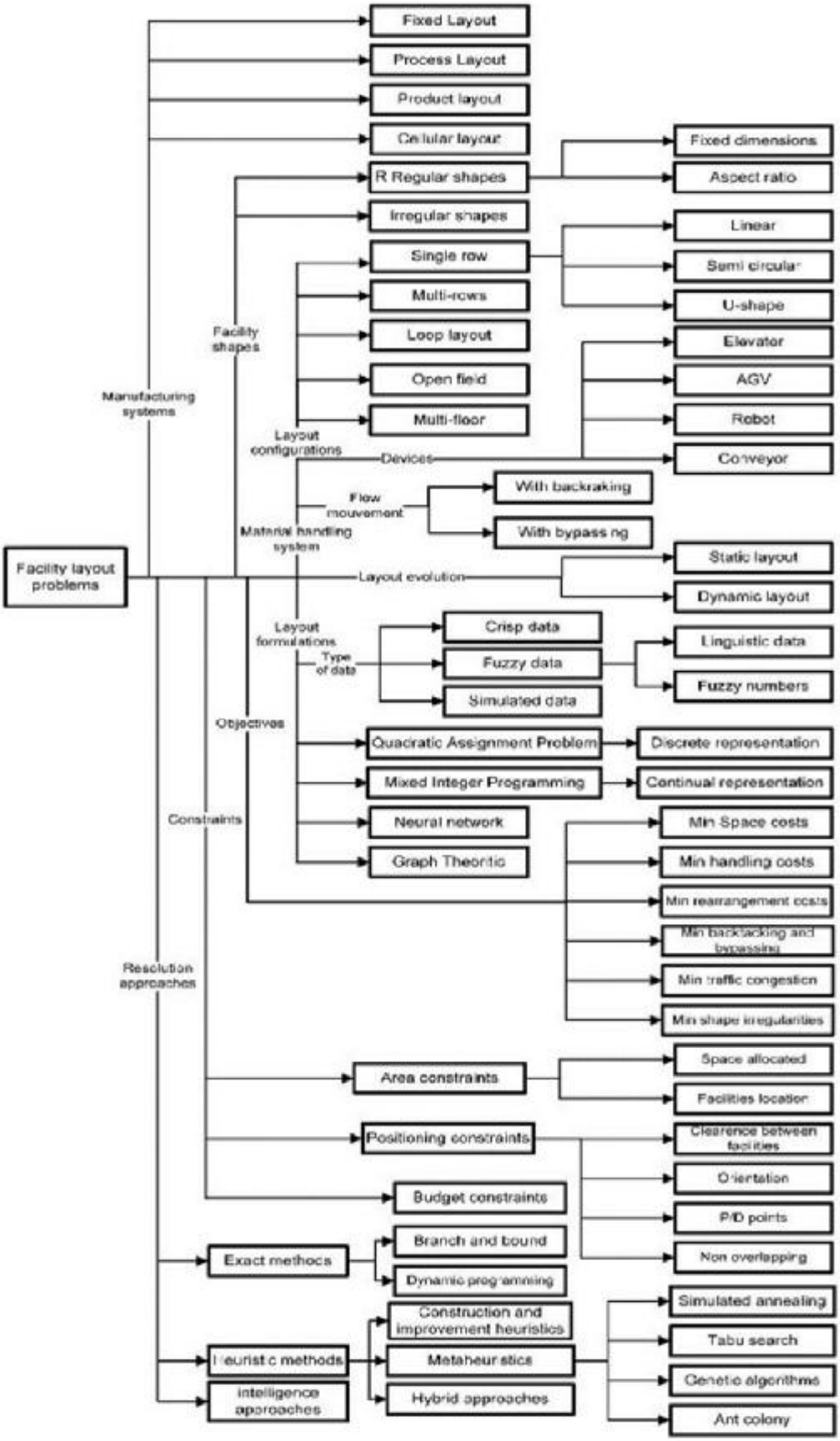


Figure 1: Representation of facility layout problems

4. Arranging the departments in unequal areas and sizes within the limits of the length and the width of the facility as possible Lee and Lee (2002).
5. Optimizing the facility layout during the design by considering the interfaces between the departments and the material movement systems as possible Shayan and Chittilappilly(2004).

Furthermore, after reviewing the several definitions illustrated by the FLP's objectives provided by the industry's experts throughout its development, it is also important to understand the several factors and inputs that affect the FLP and its approach (Drira, Pierreval, & Hajri-Gabouj, 2007):

1. Products' variety and volumes: according to the product movement, variety and volume, the type of the FLP may vary as shown in Figure 2.

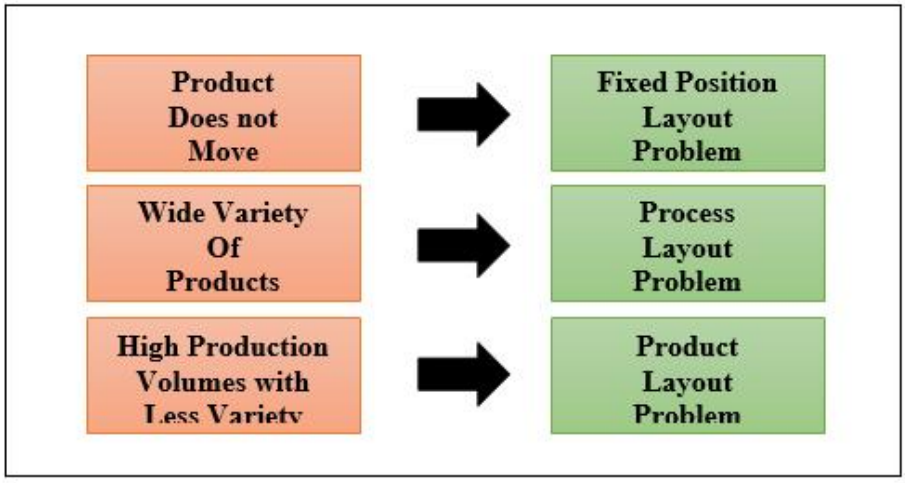


Figure 2: FLP according to movement, variety and volume of products

2. Facility shapes and dimensions: a given facility would have a fixed length (L_i) and a fixed width (W_i) and based on that an aspect ratio and bounds (a_{iu} which is the upper bound and a_{il} which is the lower bound) are developed as shown below.

$$Aspect\ Ratio\ (a_i) = \frac{L_i}{W_i}$$

Such as $a_{il} \leq a_i \leq a_{iu}$

If $a_{il} = a_i = a_{iu}$ the facility would be a fixed shape block case.

i:refers to the facility number

3. Material handling systems: this factor determines the arrangement of the departments according to the material handling path which subsequently affects the choice of the handling device. Figures 3 and 4 illustrate the single-row and the multi-rows layout as examples of the material handling systems that could be used in a facility which are related to the subject of this thesis.

Moreover, the Single Row Facility Layout Problem (SRFLP) is a type of FLP that is used commonly in arranging a certain number of hospital rooms or market departments within a facility in a single path or line. Such problems are considered to be a Non-deterministic Polynomial hard (NP-hard: the time of computation increases exponentially with the size of the problem), where the major concern is to minimize the distance between the departments that rectangular but not similar that forms the facility (Kothari & Ghosh, 2011).

Furthermore, as the flexibility of any facility might be one of the requirements due to the changing client demands or the variety of products or both, a Multi-Row Facility Layout problem (MRFLP) may be introduced. The MRFLP is usually used for a facility that has a medium variety and production volume which makes it neither a mass-production nor a customized-production facility (Soimart & Pongcharoen, 2011).

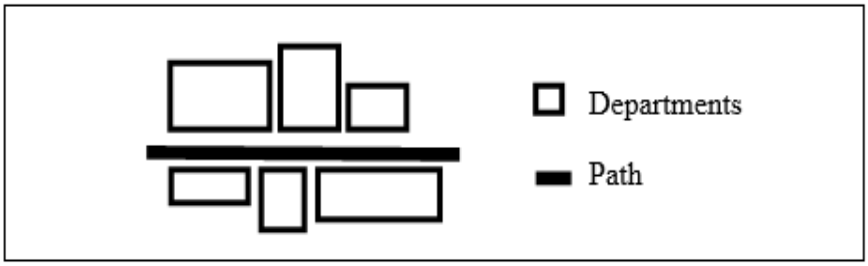


Figure 3: Single-row facility layout

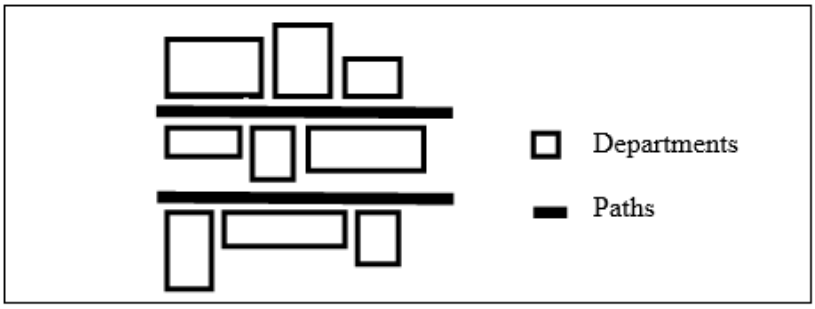


Figure 4: Multi-rows facility layout

In this study, first literature on MRFLP will be reviewed and a problem classification will be developed in order to define the problem in the next chapter. The problem formulation will be given in Chapter 3, explaining the objective and the constraints over an example layout problem. The analysis of the formulated layout problem over optimal seeking and heuristic solution techniques will be summarized in the following chapter together with some concluding results.Finally, there will be a conclusion chapter at the end.

CHAPTER 2

2. LITERATURE REVIEW

There are lots of small to medium sized enterprises around in the organized industrial regions. The structural type of most of these facilities shows us multi-row patterns either there are separation walls between rows or not. Most of these facilities start their operation as a workshop, and as time passed with increasing demand level of their customers, they extend their closed area with such modular multi-row structures. In those extension operations, generally they do not consider the optimization of material handling requirements in order to promote the flow of work-in-process inventories through the shop-floors. So, modelling and analysis of such structures, for better organization of units, departments on the shop-floor forces some level of layout optimization studies to be performed.

2.1. Facility Layout Problems Classification

The classification of the type of the facility layout problem depends on the departments' shapes, locations and relations. Moreover, the classification can dependon the solving technique, the constraints and objectives of the problem (Drira, Pierreval, & Hajri-Gabouj, 2007). Therefore, the simplest way to classify the facility layout problems is according to the following criteria:

1. Static versus dynamic: The main concern of a static FLP is that the departmentsshould not change any parameter value during the optimization process(Arikaran, Jayabalan, & Senthilkumar, 2010). However, thelocations of the departments are determined in the dynamic FLP but they change in different time periods (Afrazeh, Keivani, & Farahani, 2010).

2. Single-row versus multi-row: The single-row FLP arranges the departments on a single flow line while the multi-row FLP puts this arrangement into multiple flow lines as explained in chapter one of this manuscript(Drira, Pierreval, & Hajri-Gabouj, 2007).
3. Adjacency-based (qualitative data) versuscost-distance-based (quantitative data)objectives: the qualitative data takes into consideration the subjective evaluation of some relational factors within the departments. This method uses a rating method which classifies the ratings into A (absolutely necessary), E (especially important), I (important), O (ordinary closeness is sufficient), U (unimportant) and X (undesirable closeness), wherethe importance of these ratings ordered as: "A>E>I>O>U". Nevertheless, the values used for each rating varied between different studies as shown in Table 1 below:

Table 1: Qualitative data rating values as universally used

Rating	Value A	Value B
A	4	64
E	3	16
I	2	4
O	1	1
U	0	0
X	-1	-1024

Furthermore, the quantitative data, also known as cost-distance-based, has different distance metrics such as Rectilinear, Euclidean, Squared Euclidean or flow path. In such quantitative models, it becomes a simple objective of reducing the cost of material handling using a Quadratic Assignment Problem (QAP)(Sahoo, Shekhar, & Sahu, 2002).

4. Discrete versus continuous model representation: The discrete model uses zones with fixed locations and dimension which are pre-specified in order to locate the centers of the departments in only one zone at a time and avoid overlapping. However, using this model may impose many restrictions on choosing the solving technique. Moreover, the continuous model positions each department on X-axis and Y-axis, and by fixing the bottom-left and

upper right corners of the department, the overlap is avoided by forcing the departments to be located to the right or left of each other (Montreuil, Brotherton, & Marcotte, 2002).

5. Equal versus unequalarea requirements of departments: This classification simply means the departments can either have equal or unequal area requirements compared to each other. Therefore, the inequality of the department area is often presented as a constraint of the objective function. While considering equal department areas leads to a simpler problem, solving the unequal area problem has the advantage of better representing the real-life problems and it further reducesthe costs (Jadid & Firouz, 2016).
6. Rectangular versus non-rectangular department areas: While the rectangular department areas are simply rectangles, the non-rectangular departments may take many forms such as L-shape, U-shape, T-shape, etc.. The rectangular assumption of a FLP provides a constraint to the shape of the departments which simplifies the problem, but reduces the flexibility on the shapes that could be used in the facilities. Using a non-rectangular shape is closer to real-life problems and facilitates further cost reduction that cannot be achieved using a rectangular-shaped department approach (Bukchin & Tzur, 2014).
7. Rectilinear versus Euclidean distance metric between departments: This classification solely depends on the type of the material handling system that is used between departments. Thus, a material handling systems such as fork lifts, pallet jacks and AGV imposes a rectilinear distance using the orthogonal aisles allowed for material handling. Nevertheless, the Euclidean distance is mostly used when conveyors and monorails are used (Ozdemir, Smith, & Norman, 2002). Figure 5 shows an illustration of the rectilinear and Euclidean distance metrics between the centroids of the departments A and B.

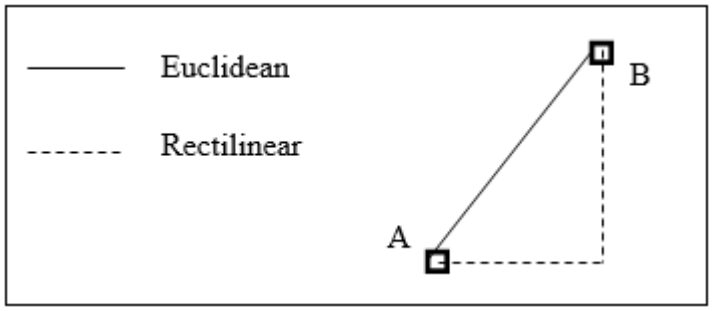


Figure 5: The rectilinear and Euclidean distances between departments A and B.

8. Single-floor versus multi-floor: the objective function may change depending on the number of floors exist in thefacility(Singh & Sharma, 2006). If the facility layout is constructed on one floor, then it is considered a single floor FLP. However, if the facility is constructed ontwo or more floors, which material is transported between the different floors, then it is considered as a multi-floorFLP (Krishnan, Jaafari, Abolhasanpour, Hojabri, & Hosein, 2009).

2.2.Literature Review

Production and industrial facilities have had a need to organize their facilities' layouts in a way that achieves the most optimum placement of their departments and accomplishes the most efficient work flow around the facilities. The necessity to solve such problem started in the 1960's and continued to develop during the 1970's and the 1980's.

The first unequal area Facility Layout Problem (FLP) was developed by Armour and Buffain (1963) with the goal to divide the main departments into smaller sub-departments to reduce the material handling and the overall cost of the facility (Shebanie, 2004). The material handling and movement costs may reach up to 50% of the total production costs, which is an enough motivation to develop a solution(Niroomand, 2013). In their study, Armour and Buffa developed a simple equation that considered two main parameters which are f_{ij} (the flow between departments $i$and$j$) and d_{ij} (the distance between two departments i and$j$) which were put in the following objective function to calculate the lowest total cost evaluated per unit time (such as 1 week , 1 month) of the facility:

$$Total\ Cost\ = \sum_{i=1}^{n} \sum_{\substack{j=1 \\ i \neq j}}^{n} f_{ij}\, d_{ij}$$

where the parameter $n$is the number of departments in the facility that are needed to be organized.

However, this permutation needed more development in order to consider all the exchange possibilities between the departments. Therefore, further development was made to this algorithm by Bazaraa in 1975, who used the same previous layout by Armour and Buffa but with similar shaped sub-departments, assembled each department by itself, and then repeated the same operation to complete the full facility layout(Shebanie, 2004).

Since then many types of facility layouts and approaches were developed in order to find the optimum facility layout and to accommodate the manufacturing changes that may appear in the same facility. Subsequently, many facility layout approaches were adopted in order to account for the different variations and types of the facility layout problems.

According toHeragu(1989), in previous studies of Facility Layout Problems (FLP) the shape and dimensions of the containing building were not taken into considerationregardless of the problem pattern. However, reading into the modeling literature, there were few models that necessitate defining the building dimensions including Quadratic Assignment Problem (QAP), nonconvex mathematical programming. Therefore, in his study, Heragu (1989) performs an objective function definition in order to eliminate the requirements of having the building dimensions.

As Heragu (1989)'s study, in a single layout problem as illustrated in Figure 6, certain assumptions have to be made:

1. The arrangement of the facility has to be referenced to a benchmark which is line (bm).
2. The orientation of the facilityhas to be in one direction.
3. The facilities' shape hasto be set prior solving the SRFLP.
4. No building boundaries are assumed.

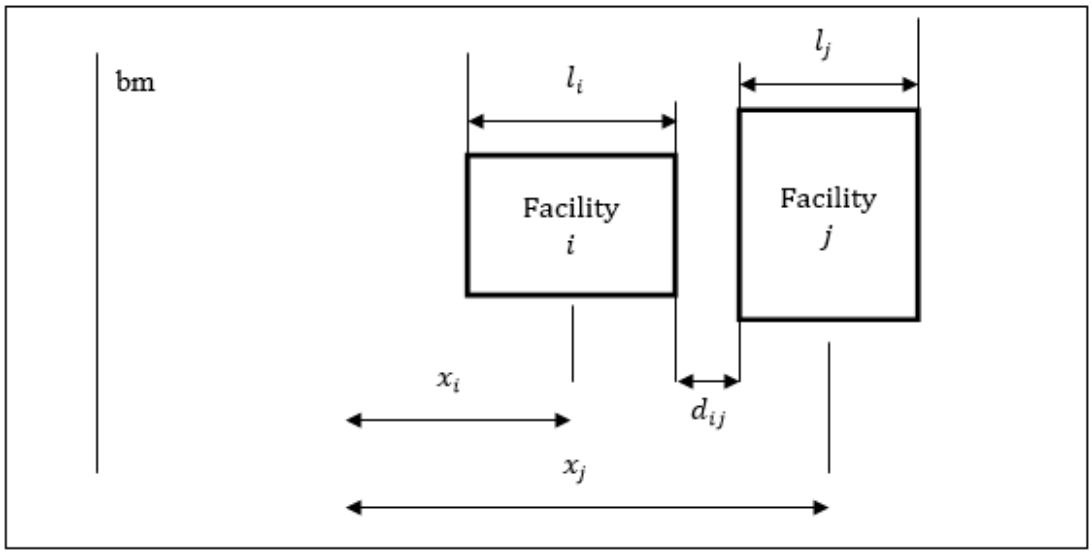


Figure 6: SRFLP model diagram

From Figure 6 and the proceeding equations, the definition of the notations are as follows:

f_{ij} = Number of trips between facilities i and j (in unit time,such as 1 week)

c_{ij} =Cost per unit of the distance travelled between facilities i and j

l_i = Length of facility i

l_j = Length of facility j

d_{ij} = The minimum distance separating facilities i and j

x_i = The distance between the benchmark line and the center of facility i

x_j = The distance between the benchmark line and the center of facility j

Therefore, the objective function that minimizes the cost while considering the number of trips between the facilities is as the following:

$$\sum_{i=1}^{n-1} \sum_{j=i+1}^{n} c_{ij} f_{ij} \left| x_i - x_j \right| \qquad (1)$$

Subject to

$$\left| x_i - x_j \right| \geq \tfrac{1}{2}(l_i - l_j) + d_{ij}, \; i = 1, \ldots, n-1, \quad j = i+1, \ldots, n \qquad (2)$$

Considering constraint (2) eliminates the errors resulting from facilities overlapping results. Moreover, since functions (1) and (2) include absolute values, the standard linear programming code cannot be used. Therefore, the layout problem model has to be transformed to a mixed-integer programming model by defining limits (3), (4) and (5):

$$x_{ij}^{+} = \begin{cases} (x_i - x_j) if x_i - x_j > 0, \\ 0 \qquad if x_i - x_j \leq 0; \end{cases} \tag{3}$$

$$x_{ij}^{-} = \begin{cases} (x_i - x_j) if x_i - x_j \leq 0, \\ 0 \qquad if x_i - x_j > 0; \end{cases} \tag{4}$$

$$z_{ij} = \begin{cases} 1 \qquad if x_i < x_j, \\ 0 \qquad if x_i \geq x_j; \end{cases} \tag{5}$$

Based on the model, the results of the limit definition is as follows:

$$\left| x_i - x_j \right| = x_{ij}^{+} + x_{ij}^{-}, \tag{6}$$

$$(x_i - x_j) = x_{ij}^{+} - x_{ij}^{-} \tag{7}$$

By setting a benchmark line and defining the aforementioned results, the model can be solved using the mixed-integer programming model without defining the building's dimensions and shape.

According to Drira et. al. (2007), there are many factors which affect the FLP as mentioned earlier in the thesis introduction which are:

1. The variety and volume of the production.
2. The shape and dimensions of the facilities.
3. The used handling system.
4. The floors' number of the manufacturing facility.
5. The movement between the departments.
6. The locations of pick-up and drop-off points.

Therefore, the SRFLP objective function was formulated as the following in order to minimize the handling costs:

$$\sum_{i=1}^{N}\sum_{j=1}^{N}\sum_{k=1}^{N}\sum_{l=1}^{N} f_{ik}d_{jl}\ X_{ij}X_{kl} \quad (8)$$

Subject to

$$\sum_{i=1}^{N} X_{ij} = 1\,, \quad j = 1 \ldots\ldots, N \quad (9)$$

$$\sum_{j=1}^{N} X_{ij} = 1\,, \quad i = 1, \ldots\ldots, N \quad (10)$$

where,

N : Number of departments in the facility

f_{ik} :The flow cost fromdepartment i to department k

d_{jl} :The distance from location j to location l

X_{ij} :A variable of 1 or 0 if department i at location$j$

From the aforementioned, the objective function (1) represents the sum of the flow costs between two certain departments. Furthermore, constraints (2) and (3) are for the purpose of having each location with only one department and the placement of each department is only in one location, respectively.

Drira et. al. (2007), suggests also using a discrete formulation to minimize congestion resulting from backtrack by using either one of the two measures which are min-sum, in order to minimize the overall total congestion, or min-max, in order to minimize the maximum congestion amongst the groups of departments.

The following equation (11) is used to calculate the distance between two given facilities in a SRFLP either by considering the centroid of each facility or the coordinates of the bottom-left corner of each facility.

$$d_{ij}((x_i, y_i), (x_j, y_j)) = \left| x_i - x_j \right| + \left| y_i + y_j \right| \quad (11)$$

Moreover, the constraint of the distance between the pick-up of department i to the drop-off the department$j$ is defined by (12):

$$d_{ij} = \left| x_i^O - x_j^{\mathrm{I}} \right| + \left| y_i^O - y_j^{\mathrm{I}} \right| \quad (12)$$

Another important constraint is to ensure no overlapping in the X-projection and the Y-projection as presented by functions (13) and (14), respectively.

$$(x_{jt} - x_{ib})(x_{jb} - x_{it}) \geq 0 \quad (13)$$

$$(y_{jt} - y_{ib})(y_{jb} - y_{it}) \geq 0 \quad (14)$$

Wong (1976) develops the facility layout problem considering one dimension of each department which are located on a single line. Therefore, the departments are identified from the longest to the shortest by 1, 2, etc. where the job of each department is well-known and the objective of the problem is to reduce the distance the product travels between each related pair of departments. Assuming that the relation of the departments is considered between their centroids and using a Branch & Bound constraint, the formulation of the problem is as the following:

$$\text{Minimize } \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} w_{ij}(R_{ij} + L_{ij}) \quad (15)$$

Subject to

$$R_{ij} - L_{ij} = x_i - x_j + \frac{1}{2}(h_j - h_i) \quad (16)$$

$$x_i - x_j + M(\alpha_{ij}) \geq h_i \quad (17)$$

$$-x_i + x_j + M(1 - \alpha_{ij}) \geq h_j \quad (18)$$

$h_i \leq x_i \leq N$ (19)

$\alpha_{ij} = 0, 1,$

$R_{ij}, L_{ij}, x_1, x_2, \ldots\ldots\ldots, x_n \geq 0,$

where,

n : Number of departments in the facility

N : The total length of all departments in the facility

w_{ij} :The weight of the product moving from department i to location j(Always positive)

R_{ij} :If department i is at the right of department j, this is the distance between thecentroid of these departments. Otherwise the value is equal to zero.

L_{ij} :If department i is at the left of department j, this is the distance between thecentroid of these departments. Otherwise the value is equal to zero.

x_i :The end point location of department i, on the interval [0, N] farthest from the line origin.

h_i :The length of department i

M : A randomly large number

$$\alpha_{ij}: \begin{cases} 1 \text{when department } i \text{ is to the left of department } j \\ 0 \text{ when department } i \text{ is to the right of department } j \end{cases}$$

The variety of the solutions, which are binary, is determined by $\frac{1}{2}n(n-1)$ and the variety of constraints are determined by $3[\frac{1}{2}n(n-1)]$, which do not include the all positive constraints and the upper and lower bounds on x_i.

The constraints are as the following:

(2) is a function that converts the distance between the end point x_i and x_j to R_{ij}or L_{ij}, which are the distance between the centroids of department i and j.(3) and (4)

ensurethat the distance which relates each two departments is not violated.(5) ensures that all the departments lie within the interval.

With using this formulationwe can use for solve a one-dimensional space allocation problem has $\frac{1}{2}n(n-1)$binary variables and the variety of $3[\frac{1}{2}n(n-1)]$constraints. This integer programmingapproach is a possibility for getting optimal solutions to smallproblems when no specific codes are readily available. If the departments have the same length, then the problem becomes the module placement which is the problem of locating n facilities in n locations.

2.3. Problem Definition

In this study, we found the following FLP is a good point to start with modeling, formulation and analysis of multi-row FLPs that can be classified as the following:

1. Static,
2. Multi-row,
3. Cost-distance based (quantitative data),
4. The representation of continuous model is somewhere betweencontinuous and discrete (more close to discrete assignment or better to call hybrid),
5. Non-equal space between departments,
6. Rectangular department areas,
7. Rectilinear distance between departments,
8. Single floor.

CHAPTER 3

3. PROBLEM FORMULATION

We try to develop the selected multi-row FLP as a mixed-integer linear programming formulation in this chapter. Then, the mathematical model will be verified and validated by interpreting the solution to be obtained from optimal seeking GAMS software. For experimental comparison of larger FLPs several heuristic methods will be described.

3.1. Mathematical Model

In our problem we have many departments, rows, and positions. In order to solve such a problem, firstly the departments are assigned to rows, then within each row the assigned departments are sequenced or ordered, so that the total cost of material handling between departments is to be minimized.

$$Minimize \ \sum_{i\in N}\sum_{j\in N} C_{ij} * f_{ij} * FD_{ij} \quad (20)$$

$$Subject\ to$$

$$\sum_{i=1}^{\mathcal{P}-1} A_{(i)} \leq (L + \varepsilon) * W \leq \sum_{i=1}^{\mathcal{P}} A_{(i)} \quad (21)$$

$$\sum_{i\in N} A_i \leq R * L * W \quad (22)$$

$$\sum_{r\in R}\sum_{p\in P} Q_{irp} = 1 \ \ for \ \forall \ i \in N \quad (23)$$

$$\sum_{i\in N} Q_{irp} \leq 1 \ \ for \ \forall \ r \in R \ and \ p \in P \quad (24)$$

$$L - \varepsilon \leq \sum_{i \in N} \sum_{p \in P} \left(\frac{A_i}{W} * Q_{irp}\right) \leq L + \varepsilon \quad for \forall\, r \in R \qquad (25)$$

$$x_{rp} = W * (r - 0.5) \quad for \forall\, r \in R \; and \; p \in P \qquad (26)$$

$$z_{rp} = \sum_{i \in N} \left(\frac{A_i}{W} * Q_{irp}\right) for \forall\, r \in R \; and \; p \in P \qquad (27)$$

$$y_{r,1} = 0.5 * z_{r,1} \quad for \forall\, r \in R \qquad (28)$$

$$y_{rp} = y_{r,p-1} + 0.5 * (z_{r,p-1} + z_{r,p}) \quad for \forall\, r \in R \; and \; p \in P \qquad (29)$$

$$x_{r_1,p_1} - x_{r_2,p_2} = XP_{r_1,p_1,r_2,p_2} - XM_{r_1,p_1,r_2,p_2} \; for \forall\, r_1, r_2 \in R \; and \; p_1, p_2 \in P \qquad (30)$$

$$y_{r_1,p_1} - y_{r_2,p_2} = YP_{r_1,p_1,r_2,p_2} - YM_{r_1,p_1,r_2,p_2} \; for \forall\, r_1, r_2 \in R \; and \; p_1, p_2 \in P \qquad (31)$$

$$XP_{r_1,p_1,r_2,p_2} + XM_{r_1,p_1,r_2,p_2} + YP_{r_1,p_1,r_2,p_2} + YM_{r_1,p_1,r_2,p_2} \leq FD_{ij} + M * \left(2 - Q_{i,r_1,p_1} - Q_{i,r_2,p_2}\right) for \forall\, r_1, r_2 \in R \; and \; p_1, p_2 \in P \; and \; i, j \in N \qquad (32)$$

$$Q_{irp} \in \{0,1\} \; for \; i \in N, r \in R, and \; p \in P \qquad (33)$$

$$x_{rp}, y_{rp}, and \; z_{rp} \geq 0 \; for \; r \in R, and \; p \in P \qquad (34)$$

$$XP_{r_1,p_1,r_2,p_2}, XM_{r_1,p_1,r_2,p_2}, YP_{r_1,p_1,r_2,p_2}, YM_{r_1,p_1,r_2,p_2} \geq 0 \; for \forall\, r_1, r_2 \in R \; and \; p_1, p_2 \in P \qquad (35)$$

$$FD_{ij} \geq 0 \; for \forall\, i \; and \; j \in N \qquad (36)$$

where parameters are defined as**:**

N : $Set\ of\ all\ departments$

R : $Set\ of\ rows$

P : $Set\ of\ positions\ in\ a\ row$

L : $Length\ of\ a\ hall$

W : $Width\ of\ a\ hall$

A_i : $Area\ requirement\ for\ department\ i \in N$

$A_{(i)}$: $ordered\ Ai\ in\ increasing\ order$

C_{ij}: $unit\ transportation\ cost\ from\ i\ th\ department\ to\ the\ j\ th\ department$

f_{ij}: $Frequency\ of\ flow\ from\ ith\ department\ to\ the\ jth\ department$

ε : $A\ sufficiently\ small\ number$

M : $A\ sufficiently\ large\ number$

where the decision variables are defined as**:**

$$Q_{irp} = \begin{cases} 1, & if\ department\ i\ is\ assigned\ to\ the\ position\ p\ at\ row\ r \\ 0, & otherwise \end{cases}$$

x_{rp} : $x - coordinate\ of\ centroid\ location\ of\ pth\ position\ at\ rth\ row$

y_{rp} : $y - coordinate\ of\ centroid\ location\ of\ pth\ position\ at\ rth\ row$

z_{rp} : $length\ of\ department\ assigned\ to\ the\ pth\ position\ at\ rth\ row$

$XP_{r_1,p_1,r_2,p_2}, XM_{r_1,p_1,r_2,p_2}, YP_{r_1,p_1,r_2,p_2}, YM_{r_1,p_1,r_2,p_2}$: $Dummy\ variables\ for\ linearization$

FD_{ij} : $Distance\ between\ departmets\ i\ and\ j$

In the objective function total material handling costs are minimized which depends on the total flow between each unit, unit transportation cost, and the distance between the departments.

Constraint 21 identifies the maximum number of positions that can be used in a row. To do that the departments are ordered according to A_i values in increasing order. Then we check the cumulative sum of the areas from 1 to $\mathcal{P} - 1$ until this cumulative sum exceeds the available area in a row. $\mathcal{P}$gives the maximum number of positions that can be used in a row.

Constraint 22 checks the feasibility of the problem that the total areas of the departments cannot exceed available area of the facility.

Constraint 23 ensures that each department is assigned to exactly one of the positions in one of the rows.

Constraint 24 guarantees that at most one department is assigned to each position in each row.

Constraint 25 checks the feasibility of assignments in each row, total length of the departments assigned to a row should be in the allowable length limits of the rows.

Constraints 26 – 29 define the centroid locations of each positions in each row.

Constraint 30 – 32 estimates the distance between the departments using the centroid locations of the positions in each row and assigned departments to these positions.

Finally,constraints 33 – 36 define the types and ranges of the decision variables.

3.2.GAMS Model

To solve the defined problem, we used GAMS software and corresponding GAMS model is given in Appendix A, Figure 10(however to preserve the feasibility of the problem we calculate the $\mathcal{P}$ value outside the model and use it as a parameter in the model).See Appendix B, Figure 11 for the developed visual basic code to generate problem test instances.

Based on the numerical experience gained over randomly generated FLPs, we show that the optimal seeking exact solution technique, GAMS software, finds an acceptable feasible solution soon, but it takes too much computation time to prove its optimality.

3.3. Alternative Solution Methods

In order to avoid computation time problem in analyzing the behavior of objective function, we have developed several heuristic solution methodologies especially for solving larger sized real-life FLPs.

3.3.1. Permutation Based Solution Approach

A permutation is any order of all departments in sequence as if the problem is single-row FLP. Here, in this solution approach, all possible permutations of departments are generated. Then, each permutation is partitioned into the number of rows (R) in the problem, based on R-median sum of area requirements. Since all possible permutations are examined in order to select the best solution, it can be considered as a total enumerative procedure.Because of the approximation made in R-median partitioning of all departments from one sequence into R sub-sequences, it behaves like an approximating heuristic. Because of generating all possible permutations, the computation time requirement significantly increases when the number of

departments is more than ten. An FPL with more than fifteen departments, the solution could not be completed within several days. See Appendix C, Figure 12 for the developed visual basic code to implement permutation based heuristic solution method.

3.3.2. Random Sampling Based Solution Approach

In this solution approach, based on the experience gained in the previous technique instead of generating all possible permutations, a random sampling can be done.It is a general method for generating random permutations. Then, these permutations are used in the same manner in the previous technique in order to obtain feasible solutions. Since the objective function is flat around the optimal point, it is robust. This way, acceptable feasible solutions can be obtained in reasonable times.See Appendix C, Figure 12 for the developed visual basic code to implement permutation based heuristic solution method. See Appendix D, Figure 13 for the developed visual basic code to implement random sampling based heuristic solution method.

CHAPTER 4

4. EXPERIMENTAL ANALYSIS OF THE SOLUTION METHODS

In the first section of this chapter, a solution of a randomly generated multi-row FLP is to be interpreted from the solution obtained from GAMS software. In the other sections, a series of experiments is to be summarized in order to show the performance of alternative solution methods.

4.1. Solving a Multi-Row Layout Problem

A sample problem instance is generated using the visual basic code. The generated departmental area requirements, cost and flow data of the sample problem instance are supplied in Table 2.

The solution of a multi-row FLP is given by the values of the assignment variables. From the results, we have tried to verify that the model produces correct results and the solution satisfies all of the constraints. See Tables 3 and 4.

Using the values of the decision variables and other parameters the assignment of departments to the rows are illustrated in the Figure 6. As can be seen from the figure,two or three departments are assigned to each row since we can assign at most three departments to each row. The length of the departments in row 2 and row 4 exactly equal to the length of the row. On theother hand, in row 1 the length of the departments is slightly smaller than the length of the row, while inrow 3 the length of the departmentsis slightly larger than the length of the row. However, thedifference is within the allowable limits.

Table 2: Generation parameters of the sample problem

Departments	Area Requirements (sq. meter)
1	250
2	850
3	400
4	900
5	850
6	800
7	200
8	350
9	500
10	900

Unit Cost ($)	1	2	3	4	5	6	7	8	9	10
1	2.20	1.88	1.10	1.19	0.98	1.28	1.92	2.45	1.35	0.09
2	0.68	1.08	1.12	2.87	1.23	2.42	0.44	1.41	2.34	0.92
3	0.20	2.63	0.84	2.64	1.19	2.16	1.25	2.17	1.05	2.21
4	0.09	0.71	0.65	1.03	2.76	1.84	2.54	0.62	0.56	2.07
5	1.35	0.58	0.66	2.56	1.07	1.43	1.68	1.14	1.15	2.84
6	1.86	2.52	0.70	2.62	2.22	0.20	0.52	2.14	1.69	2.63
7	0.08	0.98	1.71	2.09	1.02	0.96	2.80	2.25	2.63	2.20
8	1.90	0.12	1.02	2.78	2.05	0.57	0.57	2.06	2.41	2.66
9	0.02	1.84	0.72	1.03	2.49	2.36	2.18	2.74	0.10	2.05
10	0.14	1.11	2.87	0.79	0.94	2.68	1.57	1.04	0.22	2.52

Handling Frequency	1	2	3	4	5	6	7	8	9	10
1	33	35	20	35	53	8	32	34	77	93
2	33	93	38	22	28	81	82	92	40	7
3	38	44	48	21	15	84	38	38	21	21
4	94	94	6	29	0	17	41	36	73	35
5	0	42	24	97	94	54	40	26	5	50
6	88	98	64	71	53	41	43	34	93	63
7	75	68	22	2	1	84	19	5	39	58
8	59	26	80	48	72	56	7	36	26	74
9	44	84	33	20	20	31	86	3	72	3
10	12	20	58	54	57	16	7	28	95	51

Table 3: Assignment of departments to rows and positons

Row	***Position***	***Department Assigned***
1	1	1
1	2	-
1	3	-
1	4	6
2	1	7
2	2	4
2	3	-
2	4	3
3	1	2
3	2	-
3	3	9
3	4	5
4	1	10
4	2	-
4	3	-
4	4	8

Table 4: Coordinate locations of the centroids of the positions in each row

Row	Position	X-coordinate	Y-coordinate	Department Assigned
1	**1**	15.00	4.17	1
1	**2**	15.00	8.33	-
1	**3**	15.00	8.33	-
1	**4**	15.00	28.33	6
2	**1**	45.00	4.17	7
2	**2**	45.00	20.00	4
2	**3**	45.00	31.67	-
2	**4**	45.00	40.83	3
3	**1**	75.00	5.83	2
3	**2**	75.00	11.67	-
3	**3**	75.00	20.00	9
3	**4**	75.00	40.00	5
4	**1**	105.00	4.17	10
4	**2**	105.00	8.33	-
4	**3**	105.00	8.33	-
4	**4**	105.00	29.17	8

Using the centroid locations of the departments, the rectilinear distances between the departments are illustrated in Table 5. Using the values total material handling cost between the departments is estimated as $\mathbf{272,860}$based on the rectilinear distances between the centroids of the departments that is given in Table 6.

$$\text{Objective Function Value} = \sum_{i \in N} \sum_{j \in N} C_{ij} * f_{ij} * FD_{ij} = \mathbf{272,860}$$

Table 5: Coordinate locations of the centroids of the departments

Department	X-coordinate	Y-coordinate
1	15.00	4.17
2	75.00	5.83
3	45.00	40.83
4	45.00	20.00
5	75.00	40.00
6	15.00	28.33
7	45.00	4.67
8	105.00	29.17
9	75.00	20.00
10	105.00	4.17

Table 6: Rectilinear distance between the centroids of the departments

From Department	To Department									
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
1	0.00	61.67	66.67	45.83	95.83	24.17	30.00	115.00	75.83	90.00
2	61.67	0.00	65.00	44.17	34.17	82.50	31.67	53.33	14.17	31.67
3	66.67	65.00	0.00	20.83	30.83	42.50	36.67	71.67	50.83	96.67
4	45.83	44.17	20.83	0.00	50.00	38.33	15.83	69.17	30.00	75.83
5	95.83	34.17	30.83	50.00	0.00	71.67	65.83	40.83	20.00	65.83
6	24.17	82.50	42.50	38.33	71.67	0.00	54.17	90.83	68.33	114.17
7	30.00	31.67	36.67	15.83	65.83	54.17	0.00	85.00	45.83	60.00
8	115.00	53.33	71.67	69.17	40.83	90.83	85.00	0.00	39.17	25.00
9	75.83	14.17	50.83	30.00	20.00	68.33	45.83	39.17	0.00	45.83
10	90.00	31.67	96.67	75.83	65.83	114.17	60.00	25.00	45.83	0.00

The block layout to be obtained based on the solution obtained from GAMS software is illustrated in Figure 7.

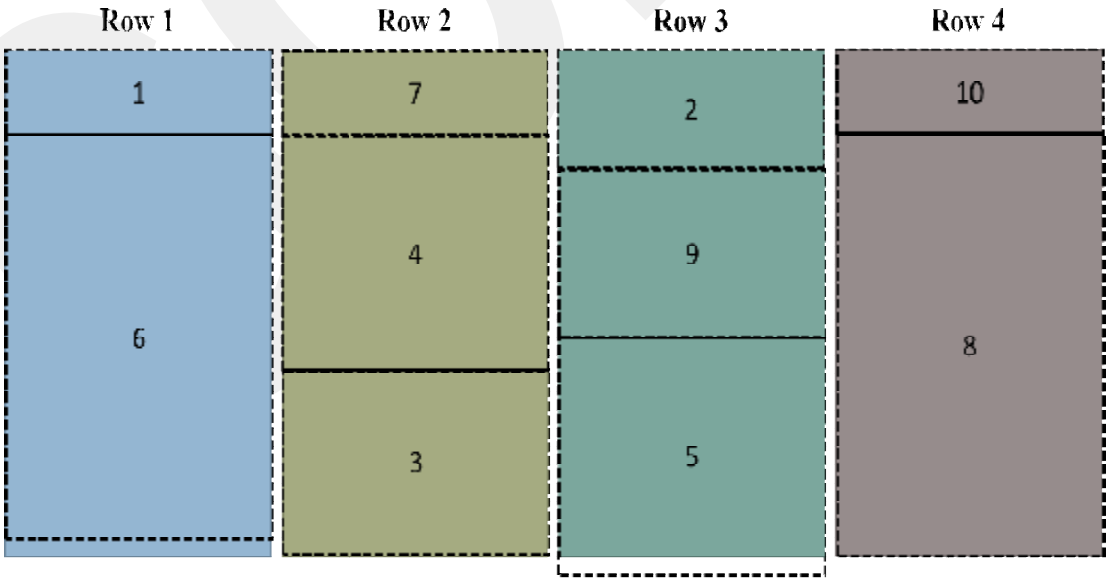


Figure 7: Assignment of departments to the positions in each row

4.2. Experiment #1: Model Validation

In the multi-row FLP we have three factors, namely Area [A], Number of Rows [R], and Number of Departments [N]. Each factor has three levels that can be classified as low, medium, and high. We try to understand the effect of each factor (areas, rows and departments) and to check the complexity of the problem depends on which factor.The factors and their levels are presented in the Table7:

Table 7: Factors and their levels

FACTOR LEVELS	FACTORS		
	Area (m^2)[A]	**Number of Rows[R]**	**Number of Departments[N]**
Low	8000	2	10
Medium	16000	4	15
High	32000	8	20

In the experimental design, while holding two factors constant, we consider different levels of other factorand try to understand effect of each factor. The constant factors are held constant at low, medium, and high levels sequentially and the other factor is considered for each factor level. Area depends on number of rows, length and width. We held width constant for all of the experiments and we modify length and by multiplying L*W*R we obtain different area levels.

To illustrate, in the first three experiments, A and R are held constant at low level and different levels of N are considered. In experiments 4-6, A and R are held constant at medium level and different levels of N are considered. Finally, in experiments 7-9, A and R are held constant at high level and different levels of N are considered. Then the independent factor is changed and the same procedure is applied for the other experiments. In experiments 1-9, A and R are held constant, in experiments 10-18, A and N are held constant, and in experiments 19-27, R and N are held constant.

The defined problem is a combinatorial optimization problem and hence it is in the NP-Hard Problem class. For this reason, even for the small test instances, the problem cannot be solved optimally in reasonable amount of time. For that purpose, weimpose a timelimitation of3600seconds for the test problems and as this resource level is exceeded, the solution procedure is terminated and the latest available result is displayed.

The test problem instances are generated by using problem instance generation tool that is prepared in Microsoft Excel. All runs are conducted using GAMS/Cplex IBM ILOG CPLEX 24.1.2 ona computer having Intel Quad Core i7 2.6 GHz processor, with 16 GB of RAM and running on 64-bit Windows 10 operating system.

A sample GAMS model taken from the first experimentis given in Appendix A. The solutions are compared in terms of number of blocks of equations, single equations, blocks of variables, single variables, non-zero elements, discrete variables, Cplextimes, optimality gaps, and MIP solutions as shown in the Table 8.

“Blocks of Equations” corresponds to the different type of equations in the model and this number is constant in each experiment. That is because, the same code is used in each experiment and hence equation types are the same.

“Single Equations” correspond to the total number of equations in all of the equation blocks. As the model size increases, then number of single equations also increases. This entry gives information about the size of the problem.

“Block of Variables” corresponds to the different number of variables used in the model independent of the indices. Hence this number is also the same in all of the experiments as the same code is used in each experiment.

“Single Variables” correspondto the total number of variables used in all of the variable blocks. As the model size increases, then number of single variables also increases. This entry gives information about the size of the problem.

The “Non-Zero Elements” entry refers to the number of non-zero coefficients in the problem matrix and this entry also given information about the complexity and size of the problem.

Table 8: Details of solutions obtained from GAMS software in the first experiment

Experiment Number	Blocks of Equations	Single Equations	Blocks of Variables	Single Variables	Non-Zero Elements	Discrete Variables	Cplex Times	Iteration Count	Optimality Gaps	MIP Solutions
1	13	20,065	10	1,067	139,395	140	3,600	38,074,896	0.981791	63,461
2	13	73,642	10	1,846	514,067	270	620	753,106	Out of Memory	
3	13	271,883	10	3,703	1,900,424	520	3,600	1,376,125	1	104,019
4	13	14,759	10	833	102,555	120	3,600	2,260,777	0.5891	159,742
5	13	178,108	10	3,866	1,243,033	420	3,600	466,285	1	376,838
6	13	411,809	10	5,233	2,878,413	640	3,600	42,304	1	949,671
7	13	58,883	10	2,717	408,855	240	3,600	15,539,734	1	51,793
8	13	132,616	10	4,898	1,623,035	480	3,600	2,995,904	1	102,595
9	13	643,405	10	7,721	4,496,164	800	3,600	402,044	1	127,210
10	13	20,065	10	1,067	139,395	140	3,600	38,635,497	0.973963	60,158
11	13	26,199	10	1,333	181,959	160	3,600	31,860,602	1	26,190
12	13	26,211	10	1,333	181,959	160	1,422	13821	Integer Infeasible	
13	13	73,642	10	1,846	514,067	270	3,600	11,553,472	1	170,002
14	13	130,876	10	2,962	913,355	360	3,600	7,910,934	1	85,733
15	13	232,616	10	4,898	1,623,035	480	NA	459,973	Out of Memory	
16	13	271,883	10	3,703	1,900,424	520	3,600	1,125,001	1	374,212
17	13	411,809	10	5,233	2,878,100	640	3,600	954,831	1	210,193
18	13	643,405	10	7,721	4,496,164	800	3,600	160,390	1	137,883
19	13	14,753	10	833	102,495	120	3,600	33,184,190	0.887376	62,106
20	13	14,753	10	833	102,495	120	3,600	39,195,888	0.893892	119,101
21	13	14,753	10	833	102,495	120	3,600	1,631,825	Out of Memory	
22	13	130,876	10	2,962	913,355	360	3,600	7,137,343	1	42,278
23	13	130,876	10	2,962	913,355	360	3,600	7,724,370	1	85,066
24	13	90,908	10	2,186	634,499	300	3,600	8,018,209	1	170,565
25	13	643,405	10	7,721	4,496,164	800	3,600	195,981	1	31,915
26	13	643,405	10	7,721	4,496,164	800	3,600	427,364	1	61,070
27	13	411,821	10	5,233	2,878,100	640	3,600	1,719,444	1	91,821

“Discrete Variables” entry shows the number of discrete variables used in the model. As number of discrete variables increases, then the problem becomes more complex and solution times increase.

“Cplex Times” shows the total execution time of the model. As we limit execution time to 3600 seconds, convergence is not guaranteed.

The entry “Iteration Count”provides the number of iterations used by the solver. Branch and Bound algorithms (as they are used for MIP) maintain two very important numbers: “best estimate” and “best integer”. The “best integer” is the best solution that satisfies all integer requirements found so far. The “best estimate” provides a bound for the optimal integer solution.Having those two numbers we can calculate the “quality” of the best integer. The quality of a solution can be measured as the distance from the optimal solution. Unfortunately, we don't have the optimal solution, but we have a bound for the optimal solution (“best estimate”). Hence an upper bound for the distance between best integer and optimal solution is “best estimate” - “best integer”. Hence, optimality gap is a measure of the quality of the obtained solutions.

Finally, “MIP Solutions” entry gives the number of integer solutions found up to program termination. This number can be used as a measure of problem complexity.

During the analysis, we consider the values of the entries explained above and try to observe the relation with the problem size and the problem complexity. As explained above we have three main factors that determine the size of the problems, these parameters are total area, number of rows, and number of departments. Each departmenthas three levels, and in the experimental design we held two factors constant while measuring the effect of the other factors.

As a result of experiments we observe that the solution time of the problem strictly depends on the number of discrete variables. As number of discrete variables increases, then the number of solutions found in the determined time frame decreases. Especially increasing number of rows and number of departments increases the number of discrete variables and hence problem complexity strictly depends on these two factors. Therefore,in order to solve larger problems in reasonable time, heuristic solution approaches should be used.

4.3. Experiment #2: GAMS vs. Random Sampling Methods

In our experimental design we have six different number of departments (7,9,11, 13,15,and 17), with four rows each row has specified area with a maximum and minimum levels and the flexibilities(0%, 10%, and 25%). In this experimental setting we considered just one row which its area is 3000 with all the several departments options and 0% flexibility for all the experiments.

The experiment parameter settings are given in Table9. For each setting, we have generated different random parameters for the problem and then run GAMS model and Random Sampling based solution (RAND).

Table 9: The design parameters for experiments #2

Number of Halls in the Facility		Number of Departments											
		7		9		11		13		15		17	
1	3000	214 .. 643		167 .. 500		136 .. 409		115 .. 346		100 .. 300		88 .. 265	
		±0%		±0%		±0%		±0%		±0%		±0%	
		GAMS ..	RAND.	GAMS	RAND.	GAMS ..	RAND.	GAMS ..	RAND.	GAMS ..	RAND.	GAMS ..	RAND.

As can be seen from Table 10 and Figure 8, GAMS model produced better results compared to the RAND for all of the experiments. In these experiments, we use a time limit resource limit in GAMS model, and for some experiments, the model cannot be solved optimally within the determined time interval.

Hence, not all of the results are optimal. If we increase the time resource limit then we can obtain better results compared to RAND. On the other hand, RAND produces feasible results within seconds. In this respect, it is far better than the mathematical model.

Table 10: Results of the experiment #2 for GAMS and Random sampling

EXPERIMENT # 1		GAMS SOLUTION		RANDOM SAMPLING BASED SOLUTION				% Difference
No	Problem ID	GAMS:Iteration#	GAMS:Obj.	RAND: Number of Iterations	RAND: Objective	RAND: Max. Objective	RAND: Avg. Objective	
1	01070001	84,253	**297,779.49**	280,001	**335,593.87**	594,789.00	447,642.21	**-12.7**
2	01070002	94,465	**311,589.18**	280,001	**311,589.18**	469,285.74	396,542.95	**0.0**
3	01070003	84,253	**297,779.49**	280,001	**297,779.49**	529,045.75	397,143.14	**0.0**
4	01090001	3,560,676	**556,778.35**	360,001	**557,439.61**	860,149.37	706,525.83	**-0.1**
5	01090002	3,302,174	**598,519.35**	360,001	**598,533.79**	958,291.12	772,947.40	**-0.0**
6	01070003	4,818,670	**413,125.09**	360,001	**413,125.09**	714,260.81	552,112.08	**0.0**
7	01110001	9,409,270	**656,070.64**	440,001	**670,234.40**	1,241,625.41	929,463.64	**-2.2**
8	01110002	10,686,075	**760,496.96**	440,001	**769,118.24**	1,191,176.41	974,543.91	**-1.1**
9	01110003	1,397,488	**687,475.39**	440,001	**703,804.93**	1,160,461.92	910,599.19	**-2.4**
10	01130001	1,095,839	**1,000,085.00**	520,001	**1,025,457.93**	1,668,622.59	1,310,359.80	**-2.5**
11	01130002	423,106	**1,090,697.80**	520,001	**1,093,873.88**	1,702,999.38	1,384,655.65	**-0.3**
12	01130003	727,655	**1,101,111.35**	520,001	**1,144,495.52**	1,767,652.43	1,469,983.00	**-3.9**
13	01150001	760,966	**1,363,005.87**	600,001	**1,399,326.44**	2,188,605.02	1,790,319.95	**-2.7**
14	01150002	188,373	**1,321,061.07**	600,001	**1,363,802.31**	2,247,304.74	1,771,368.09	**-3.2**
15	01150003	439,659	**1,474,520.47**	600,001	**1,495,446.54**	2,300,588.10	1,875,698.52	**-1.4**
16	01170001	660,882	**1,816,886.84**	680,001	**1,822,920.17**	2,803,639.03	2,250,907.56	**-0.3**
17	01170002	776,092	**1,845,981.37**	680,001	**1,876,292.35**	3,004,472.06	2,414,596.69	**-1.6**
18	01170003	306,605	**1,816,886.84**	680,001	**1,822,920.17**	2,803,639.03	2,250,907.56	**-0.3**

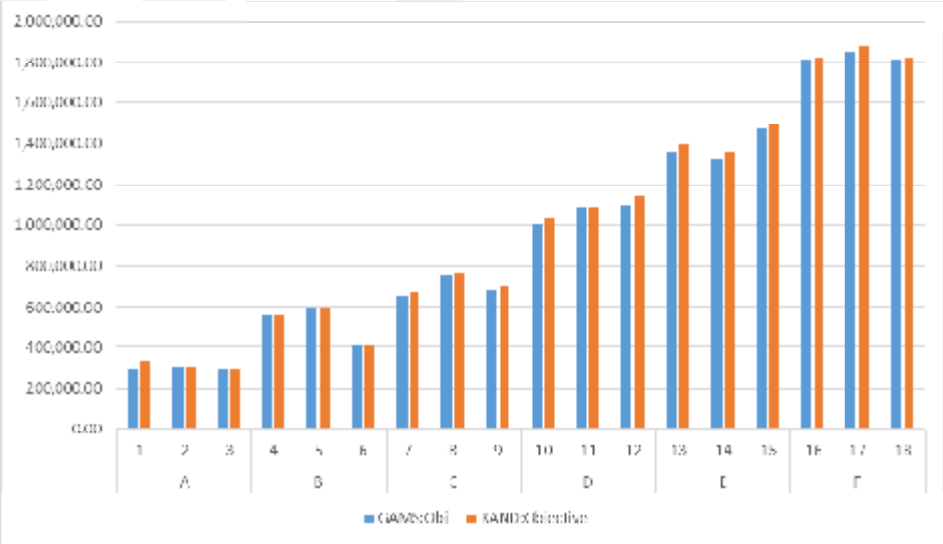


Figure 8: Percentage of the difference between GAMS and Random sampling objectives

4.4.Experiment #3: GAMS vs. Permutation vs. RandomSampling Methods

In this experimental setting, we considered three levels of row numbers namely 2, 3 and 4 their areas 6000,9000 and 12000 and we used all the flexibility levels for eachexperiment. In addition, we have three department number options, namely 7, 9, 11.The parameter settings of the problem instances related to experiment #3 are given in Table 11. For each setting, we have generated differentrandomparameters for the problem instances and then run the GAMS model, Permutation Based Heuristic Solution, andRandomSampling Based Heuristic Solution. The results are summarized in Table 12.

Table 11: The design parameters for experiments #3

		Number of Halls in the Facility								
		2			3			4		
		6000			9000			12000		
Number of Departments	7	429	..	1286	643	..	1929	857	..	2571
		± 0%	± 10%	± 25%	± 0%	± 10%	± 25%	± 0%	± 10%	± 25%
		GAMS	PERM.	RAND.	GAMS	PERM.	RAND.	GAMS	PERM.	RAND.
	9	333	..	1000	500	..	1500	667	..	2000
		± 0%	± 10%	± 25%	± 0%	± 10%	± 25%	± 0%	± 10%	± 25%
		GAMS	PERM.	RAND.	GAMS	PERM.	RAND.	GAMS	PERM.	RAND.
	11	273	..	818	409	..	1227	545	..	1636
		± 0%	± 10%	± 25%	± 0%	± 10%	± 25%	± 0%	± 10%	± 25%
		GAMS	PERM.	RAND.	GAMS	PERM.	RAND.	GAMS	PERM.	RAND.

For some instances, we cannot obtain feasible solutions, such as experiments 19-21, 37-39, 46-48, and 73-75. In all of these problems, since the flexibility % is zero, no integer feasible solution could be found in which all departments assigned to each row has the same total hall area.

Table 12: Results of the experiment #3 for GAMS, Permutation, and Random sampling

Observation No	Number of Departments	Number of Halls	Flexibility %	GAMS Objective	Permutation Objective	Random Sampling Objective
1				491,428.57	491,428.57	491,428.57
2			0	476,288.01	476,288.01	476,288.01
3				420,198.79	420,198.79	420,198.79
4				477,536.03	477,536.03	477,536.03
5		2	10	453,318.38	453,318.38	453,318.38
6				426,181.15	426,181.15	426,181.15
7				466,211.22	466,211.22	466,211.22
8			25	440,269.93	440,269.93	440,269.93
9				450,120.37	450,120.37	450,120.37
10				648,322.58	648,322.59	648,322.59
11			0	583,473.60	583,473.60	583,473.60
12				996,249.71	996,249.72	996,249.72
13				538,109.82	538,109.82	538,109.82
14	7	3	10	556,965.31	556,965.31	556,965.31
15				543,888.11	543,888.11	543,888.11
16				642,977.03	642,977.03	642,977.03
17			25	570,509.18	570,509.18	570,509.18
18				668,064.25	668,064.25	668,064.26
19				Infeasible	Infeasible	Infeasible
20			0	Infeasible	Infeasible	Infeasible
21				Infeasible	Infeasible	Infeasible
22				719,421.93	719,421.94	719,421.94
23		4	10	583,808.04	583,808.05	583,808.05
24				492,017.58	492,017.58	492,017.58
25				551,363.50	551,363.50	551,363.50
26			25	669,939.05	672,400.31	672,400.31
27				610,017.55	610,017.55	610,017.55
28				811,006.78	811,007.42	811,007.42
29			0	853,194.66	853,194.66	853,194.66
30				810,081.01	810,082.02	810,082.02
31				978,858.63	976,644.55	976,644.56
32		2	10	697,975.42	697,975.41	697,975.41
33				671,285.69	671,285.69	671,285.69
34				897,687.49	897,687.50	897,687.50
35			25	718,093.62	718,093.91	718,093.91
36				858,260.78	858,260.77	858,260.77
37				Infeasible	Infeasible	Infeasible
38			0	Infeasible	Infeasible	Infeasible
39				Infeasible	Infeasible	Infeasible
40				885,846.75	896,522.82	896,522.82
41	9	3	10	843,140.05	843,140.04	843,140.04
42				824,729.33	824,729.32	824,729.32
43				900,396.33	891,499.32	891,499.32
44			25	880,882.05	880,882.05	880,882.05
45				998,451.47	998,451.47	998,451.47
46				Infeasible	Infeasible	Infeasible
47			0	Infeasible	Infeasible	Infeasible
48				Infeasible	Infeasible	Infeasible
49				1,018,084.15	1,018,084.16	1,018,084.16
50		4	10	1,022,301.18	1,022,301.19	1,022,301.19
51				1,161,111.73	1,161,111.74	1,161,111.74
52				1,093,037.18	1,093,037.18	1,093,037.18
53			25	1,158,092.91	1,158,092.91	1,158,092.91
54				1,235,402.99	1,235,402.99	1,235,402.99
55				1,178,560.65	1,178,563.33	1,188,952.53
56			0	1,155,467.53	1,122,858.75	1,194,688.25
57				1,235,187.94	1,235,187.95	1,290,508.56
58				1,060,962.74	1,004,684.35	1,047,191.44
59		2	10	1,174,129.25	1,144,968.37	1,189,937.75
60				1,301,917.07	1,280,971.95	1,305,670.61
61				1,075,885.70	1,021,681.35	1,028,036.19
62			25	1,228,965.39	1,213,170.31	1,255,156.25
63				1,100,578.21	1,060,244.23	1,086,323.43
64				1,680,070.78	1,689,010.77	1,705,315.59
65			0	1,581,837.07	1,545,923.41	1,582,076.56
66				1,538,019.71	1,545,799.41	1,582,757.66
67				1,542,271.08	1,542,271.08	1,560,570.85
68	11	3	10	1,498,948.62	1,478,675.32	1,502,674.25
69				1,422,812.48	1,439,906.95	1,427,281.71
70				1,388,201.17	1,366,212.15	1,406,038.19
71			25	1,341,054.88	1,315,681.91	1,323,692.83
72				1,697,518.67	1,677,216.98	1,700,084.40
73				Infeasible	Infeasible	Infeasible
74			0	Infeasible	Infeasible	Infeasible
75				Infeasible	Infeasible	Infeasible
76				1,847,600.04	1,825,094.78	1,872,590.76
77		4	10	1,834,803.06	1,822,188.38	1,870,494.47
78				1,595,134.08	1,522,260.17	1,583,610.44
79				1,622,161.13	1,562,253.40	1,607,432.56
80			25	1,553,085.06	1,492,446.81	1,538,676.41
81				1,570,869.73	1,557,105.95	1,588,387.15
			Average:	966455.01	958884.85	969673.28
			St.Dev:	409148.54	400766.57	413693.36
			Maximum:	1847600.04	1825094.78	1872590.76
			Minimum:	479198.79	479198.79	479198.79

Permutation Based and Random Sampling Based Methods produced the very similar objective function values. On the other hand, GAMS produced better results in the smaller-sized problem instances, and worse results in the other experiments. If the time limit for the GAMS model is increased, much better results can be obtained. However, in terms of solution time requirements, GAMS and Permutation based heuristic technique haveworse performance than Random Sampling based heuristic technique.

Especially as problem size increases, solution times and optimality gaps in GAMS solutions increases extensively.

4.5. ConcludingResults

Recall that, in experiment # 2, we have 6 different levels of the number of departments (7, 9, 11, 13, 15, and 17) with single-row layout structures, with four rows each row has specified area with a maximum and minimum levels and the flexibilities (0%, 10%, and 25%), for each of thesea single-row FLP instances (a total of 18 problem instances) generated and solved with limited computation time for GAMS software and permutation based heuristic method.

See Figure 9, for a summary of T-test for paired samples of objectives obtained fromtime-limited GAMS software and Random sampling heuristic technique.Based on 18 observations in this experiment,it is found that the sample means are967,213.919 and 983,430.773, respectively for GAMS and Random sampling heuristic objectives.The sample mean of difference between (GAMS – RANDOM) with a value −16,216.853 showed that GAMS objectives are less than the objectives obtained from Random sampling heuristic technique. The 95% confidence interval for the differences is $-24{,}171.38 < \mu_D < -8{,}262.327$.

It is statistically concluded that the null hypothesis H_0 ($H_0 : \mu_D \geq 0$) is rejected. Therefore, there is enough evidence to claim that population mean μ_{GAMS} is less than μ_{RANDOM}, at the 0.05 significance level.

Summary of T-test for paired samples: GAMS versus Random Sampling

From the sample data, it is found that the corresponding sample means are:

$$\overline{X_1} = 967{,}213.919$$
$$\overline{X_2} = 983{,}430.773$$

Also, the provided sample standard deviations are:

$$s_1 = 539{,}331.206$$
$$s_2 = 544{,}742.843$$

and the sample size is n = 18. For the score differences we have

$$\overline{D} = -16{,}216.853$$
$$s_D = 15{,}995.805$$

Observation No	Number of Departments	GAMS Objective	Random Sampling Objective	Difference (%)
1	7	297,779.49	335,593.87	-0.13
2		311,589.18	311,589.18	0.00
3		297,779.49	297,779.49	0.00
4	9	556,778.35	557,439.61	0.00
5		598,519.35	598,533.79	0.00
6		413,125.09	413,125.09	0.00
7	11	656,070.64	670,234.40	-0.02
8		760,496.96	769,118.24	-0.01
9		687,475.39	703,804.93	-0.02
10	13	1,000,085.00	1,025,457.93	-0.03
11		1,090,697.80	1,093,873.88	0.00
12		1,101,111.35	1,144,495.52	-0.04
13	15	1,363,005.87	1,399,326.44	-0.03
14		1,321,061.07	1,363,802.31	-0.03
15		1,474,520.47	1,495,446.54	-0.01
16	17	1,816,886.84	1,822,920.17	0.00
17		1,845,981.37	1,876,292.35	-0.02
18		1,816,886.84	1,822,920.17	0.00
			Average:	-0.02
			St.Dev:	0.03
			Maximum:	0.00
			Minimum:	-0.13

(1) *Null and Alternative Hypotheses*

The following null and alternative hypotheses need to be tested:

$$H_0 : \mu_D \geq 0$$
$$H_1 : \mu_D < 0$$

This corresponds to a left-tailed test, for which a t-test for two paired samples be used.

(2) *Rejection Region*

Based on the information provided, the significance level is α = 0.05, and the degrees of freedom are d_f = 17. Hence, it is found that the critical value for this left-tailed test is $t_c = -1.74$, for α = 0.05 and d_f = 17. The rejection region for this left-tailed test is R = {t :< −1.74}.

(3) *Test Statistics*

The t-statistic is computed as shown in the following formula:

$$T = \frac{\overline{D}}{s_D/\sqrt{n}} = \frac{-16{,}216.853}{15995.805/\sqrt{18}} = -4.301$$

(4) *Decision about the null hypothesis*

Since it is observed that t = −4.301 $< t_c = -1.74$, it is then concluded that *the null hypothesis is rejected.* Using the P-value approach: The p-value is p = 0.0002, and since p = 0.0002 < 0.05, it is concluded that the null hypothesis is rejected.

(5) *Conclusion*

It is concluded that the null hypothesis H_0 is rejected. Therefore, ***there is enough evidence to claim that population mean μ_1 is less than μ_2,*** at the 0.05 significance level.

(6) Confidence Interval

The 95% confidence interval is −24,171.38 <μ_D< −8,262.327

Figure 9: Summary of T-test for paired samples: GAMS and Random sampling objectives

See Figure 10, for a summary of T-test for paired samples of objectives obtained fromtime-limited GAMS software andPermutation based heuristic technique.Based on 69 observations (since out of 81 problem instances, 12 result with infeasible solution) in this experiment,it is found that the sample means are966,455.012 and 958,884.856, respectively for GAMS and Permutation based heuristic objectives.The sample mean of difference between (GAMS – PERMUTATION) with a value 7,570.156 showed that GAMS objectives are greater than the objectives obtained from Permutation based heuristic technique. The 95% confidence interval for the differences is $3{,}236.913<\mu_D<11{,}903.399$. It is statistically concluded that the null hypothesis H_0($H_0 : \mu_D \geq 0$) is rejected. Therefore, there is enough evidence to claim that population mean μ_{GAMS} is greater than $\mu_{PERMUTATION}$, at the 0.05 significance level.

This shows that, since permutation based heuristic technique is a total enumerative procedure, it can give better solutions than the time-limited GAMS software as the problem size increases.

See Figure 11, for a summary of T-test for paired samples of objectives obtained fromtime-limited GAMS software and Random sampling based heuristic technique.Based on 69 observations (since out of 81 problem instances, 12 result with infeasible solution) in this experiment,it is found that the sample means are966,455.012 and 969,673.283, respectively for GAMS and Random sampling based heuristic objectives.The sample mean of difference between (GAMS – RANDOM) with a value −3,218.272 showed that GAMS objectives are slightly less than the objectives obtained from Random sampling based heuristic technique. The 95% confidence interval for the differences is $-6{,}680.703<\mu_D<244.16$. It is statistically concluded that the null hypothesis H_0($H_0 : \mu_D \geq 0$) is rejected. Therefore, there is enough evidence to claim that population mean μ_{GAMS} is less than μ_{RANDOM}, at the 0.05 significance level.

This shows that, since Random sampling based heuristic technique is a just considering a small portion of all possible permutations, GAMS can give better solutions than the Random sampling based heuristic. Surprisingly, the 95% confidence interval for the differences includes zero. This highlights to a significant

potential of Random sampling based heuristic technique to reach acceptably good solutions within a reasonable time.

Summary of T-test for paired samples: GAMS versus Permutation

From the sample data, it is found that the corresponding sample means are:

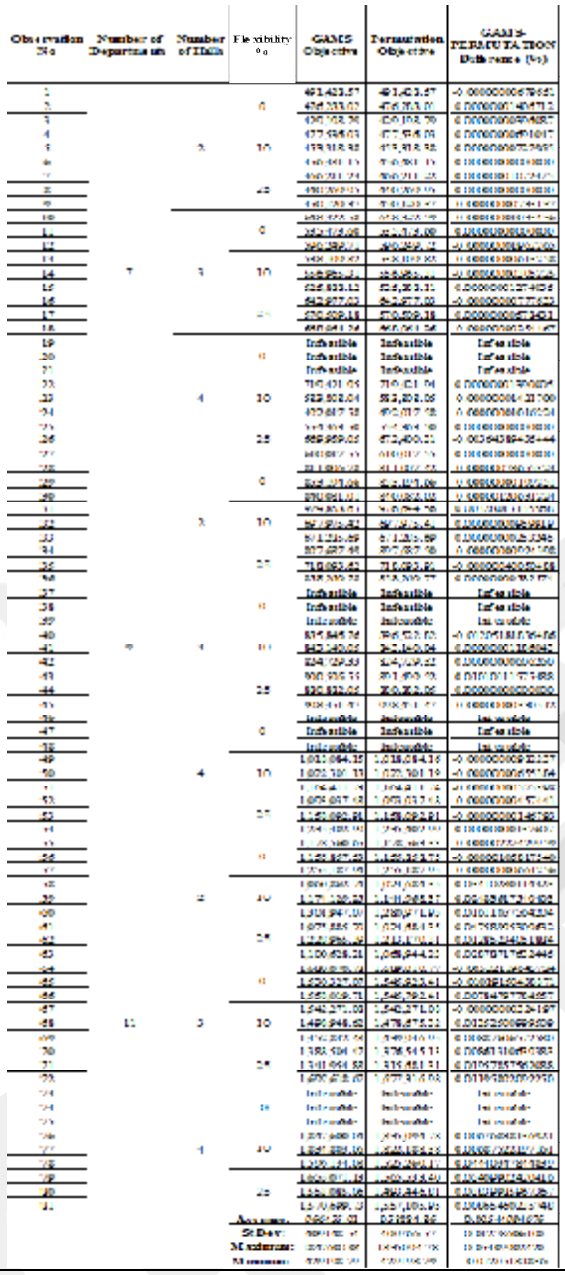
$\overline{X_1}$ = 966,455.012
$\overline{X_2}$= 958,884.856

Also, the provided sample standard deviations are:

s_1 = 409,148.537
s_2= 400,766.567

and the sample size is n = 69. For the score differences we have

$\overline{D}$ = 7,570.156
s_D = 18,038.175



(1) *Null and Alternative Hypotheses*

The following null and alternative hypotheses need to be tested:

$$H_0: \mu_D \leq 0$$
$$H_1: \mu_D > 0$$

This corresponds to a right-tailed test, for which a t-test for two paired samples be used.

(2) *Rejection Region*

Based on the information provided, the significance level is α = 0.05, and the degrees of freedom are d_f = 68. Hence, it is found that the critical value for this right-tailed test is t_c = 1.668, for α = 0.05 and d_f = 68. The rejection region for this right-tailed test is R = {t : t> 1.668}.

(3) *Test Statistics*

The t-statistic is computed as shown in the following formula:

$$T = \frac{\overline{D}}{s_D/\sqrt{n}} = \frac{7,570.1563}{18,038.175/\sqrt{69}} = 3.486$$

(4) *Decision about the null hypothesis*

Since it is observed that t = 3.486 >t_c = 1.668, it is then concluded that *the null hypothesis is rejected.* Using the P-value approach: The p-value is p = 0.0004, and since p = 0.0004 < 0.05, it is concluded that the null hypothesis is rejected.

(5) *Conclusion*

It is concluded that the null hypothesis H_0 is rejected. Therefore, ***there is enough evidence to claim that population mean μ_1 is greater than μ_2,*** at the 0.05 significance level.

(6) Confidence Interval

The 95% confidence interval is 3,236.913 <μ_D< 11,903.399.

Figure 10: Summary of T-test for paired samples: GAMS and Permutation objectives

Summary of T-test for paired samples: GAMS versus Random Sampling

From the sample data, it is found that the corresponding sample means are:

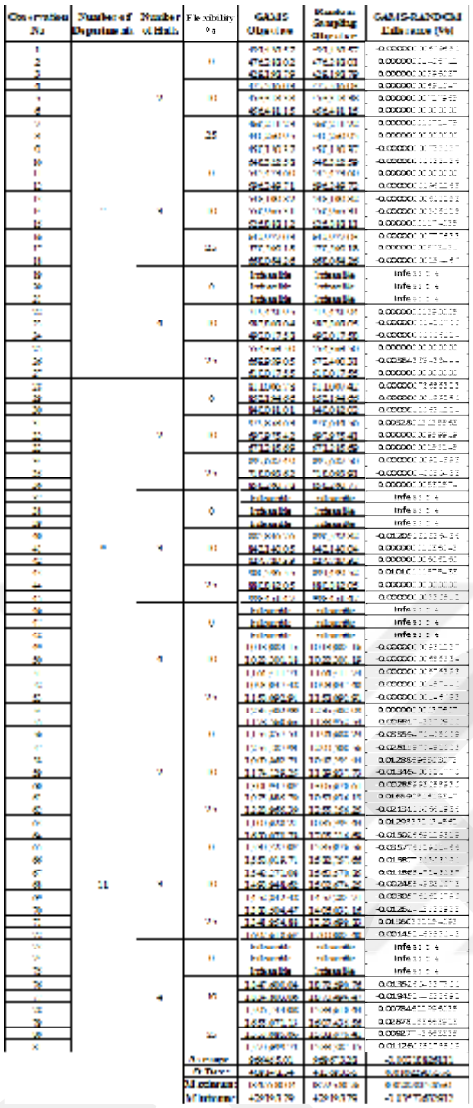
$$\overline{X_1} = 966,455.012$$
$$\overline{X_2} = 969,673.283$$

Also, the provided sample standard deviations are:

$$s_1 = 409,148.537$$
$$s_2 = 413,693.557$$

and the sample size is n = 69. For the score differences we have

$$\overline{D} = -3,218.272$$
$$s_D = 14,413.212$$



(1) *Null and Alternative Hypotheses*

The following null and alternative hypotheses need to be tested:

$$H_0: \mu_D \geq 0$$
$$H_1: \mu_D < 0$$

This corresponds to a left-tailed test, for which a t-test for two paired samples be used.

(2) *Rejection Region*

Based on the information provided, the significance level is α = 0.05, and the degrees of freedom are d_f = 68. Hence, it is found that the critical value for this left-tailed test is t_c = −1.668, for α = 0.05 and d_f = 68. The rejection region for this left-tailed test is R = {t : t< −1.668}.

(3) *Test Statistics*

The t-statistic is computed as shown in the following formula:

$$T = \frac{\overline{D}}{s_D/\sqrt{n}} = \frac{-3,218.272}{14,413.212/\sqrt{69}} = -1.855$$

(4) *Decision about the null hypothesis*

Since it is observed that t = −1.855 >t_c = −1.668, it is then concluded that *the null hypothesis is rejected.* Using the P-value approach: The p-value is p = 0.034, and since p = 0.034 < 0.05, it is concluded that the null hypothesis is rejected.

(5) *Conclusion*

It is concluded that the null hypothesis H_0 is rejected. Therefore, ***there is enough evidence to claim that population mean μ_1 is less than μ_2,*** at the 0.05 significance level.

(6) Confidence Interval

The 95% confidence interval is −6,680.703 <μ_D< 244.16.

Figure 11: Summary of T-test for paired samples: GAMS and Random sampling objectives

See Figure 12, for a summary of T-test for paired samples of objectives obtained fromPermutation based and Random sampling heuristic techniques.Based on 16 observations in this experiment,it is found that the sample means are958,884.856 and 969,673.283, respectively for Permutation based and Random sampling heuristic objectives.

Summary of T-test for paired samples: PERMUTATION versus Random Sampling

From the sample data, it is found that the corresponding sample means are:

$\overline{X_1}$ = 958,884.856

$\overline{X_2}$= 969,673.283

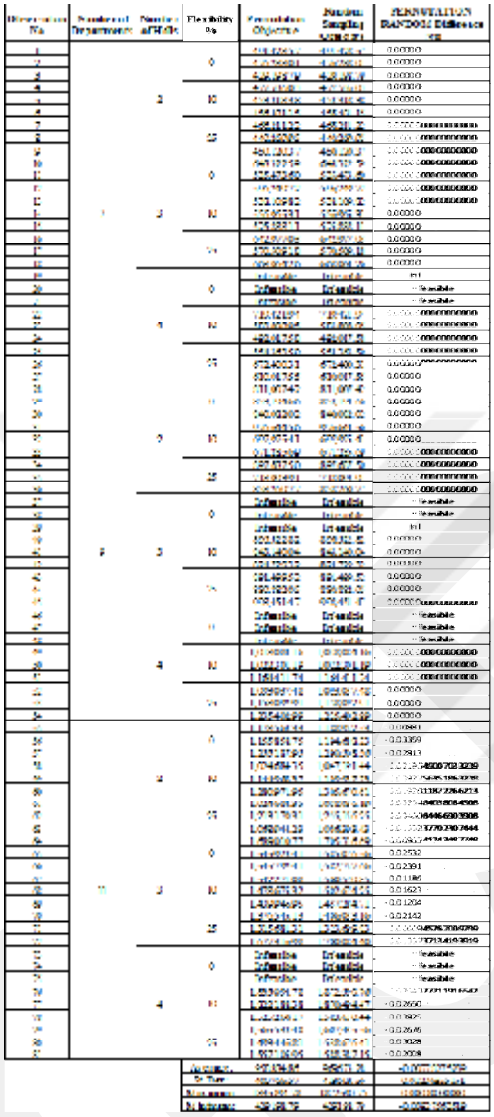
Also, the provided sample standard deviations are:

s_1 = 400,766.567

s_2 = 413,693.557

and the sample size is n = 69. For the score differences we have

$\overline{D}$ = −10,788.427
s_D = 16,673.276



(1) *Null and Alternative Hypotheses*

The following null and alternative hypotheses need to be tested:

$$H_0: \mu_D \geq 0$$
$$H_1: \mu_D < 0$$

This corresponds to a left-tailed test, for which a t-test for two paired samples be used.

(2) *Rejection Region*

Based on the information provided, the significance level is α = 0.05, and the degrees of freedom are d_f = 68. Hence, it is found that the critical value for this left-tailed test is t_c = −1.668, for α = 0.05 and d_f = 68. The rejection region for this left-tailed test is R = {t : t< −1.668}.

(3) *Test Statistics*

The t-statistic is computed as shown in the following formula:

$$T = \frac{\overline{D}}{s_D/\sqrt{n}} = \frac{-10{,}788.427}{16673.276/\sqrt{69}} = -5.375$$

(4) *Decision about the null hypothesis*

Since it is observed that t = −5.375 >t_c = −1.668, it is then concluded that *the null hypothesis is rejected.* Using the P-value approach: The p-value is p = 0.0, and since p = 0.0 < 0.05, it is concluded that the null hypothesis is rejected.

(5) *Conclusion*

It is concluded that the null hypothesis H_0 is rejected. Therefore, ***there is enough evidence to claim that population mean μ_1 is less than μ_2,*** at the 0.05 significance level.

(6) Confidence Interval

The 95% confidence interval is −14,793.785 <μ_D< −6,783.068.

Figure 12: Summary of T-test for paired samples: Permutation and Random sampling objectives

The sample mean of difference between (PERMUTATION – RANDOM) with a value −10,788.427 showed that Permutation based heuristic objectives are less than the objectives obtained from Random sampling heuristic technique. The 95% confidence interval for the differences is −14,793.785<μ_D<−6,783.068. It is statistically concluded that the null hypothesis H_0(H_0 : $\mu_D \geq 0$) is rejected. Therefore, there is enough evidence to claim that population mean $\mu_{PERMUTATION}$ is less than μ_{RANDOM}, at the 0.05 significance level.

CHAPTER V

5. CONCLUSION

In this study, we considered multi-row, cost-distance objective, rectangular non-equal departmental areas to formulate a layout problem. With the formulated model of a multi-row layout problem we tried toanalyzeand compare alternative solution techniques over various facility layout problems in several experiments. This study is applicable for problems where either product or process type layouts exist in the factory.

In the formulation of the model, there are a number of rows in the layout and positions within a row, so that the mathematical model will assign the departments to one of these positions. For this reason, the assignment process shows both discrete and continuous characteristics, to be hybrid.The solutions obtained using an optimal seeking solution technique(GAMS software) is to be compared with permutation based total enumeration technique and a limited sampling of random permutations. Although, the better solutions are obtained using time-limited GAMS software, the solutions obtained from permutation based and random sampling heuristic techniques are not statistically worse since the objective function behaves so flat around the optimal point as we conclude that it is robust.

In the experimental analysis, each factor has three levels that can be classified as low, medium, and high. While holding two factors constant, we consider different levels of other factor and try to understand effect of each factor and the factor that has more effect on the complexity of the problem and time limitation(as we considered 3600 seconds).As a result of experiments, we observed that the solution time of the problem depends on the number of discrete variables. As number of discrete variables increases, then the number of solutions found in the determined time frame decreases. Especially increasing number of rows and number of

departments increases the number of discrete variables and hence problem complexity strictly depends on these two factors.

Additionally, we designed two more experiments. In the first one, we consideredsingle-row layout problem for six different number of departments (7,9,11,13,15 and 17) with 0% (since there is only one row in the layout) flexibility we have generated data for different random problem instancesand then tried to solve these problems usingtime-limited GAMS software and alternatively, random sampling based heuristic solution technique. Based on 18 observations in this experiment, statistically there is enough evidence to claim that population mean μ_{GAMS} is less than μ_{RANDOM}, at the 0.05 significance level. That is,time-limited GAMS software produced better solutions compared to the Random sampling based heuristic solution technique.

In the second experimental design, we considered three different number of departments (namely 7,9 and 11) and threedifferent number of rows (namely 2, 3 and 4), having different area requirements with different flexibilities (corresponding proportional deviation of areas assigned between halls, namely, 0%,10%, 25%)for each experiment. We have generated different random parameters for the problem instances and then run the time-limited GAMS software,Permutation based, and Random sampling based heuristic techniques. Based on 69 observations (since out of 81 problem instances, 12 result with infeasible solution) in this experiment, statistically there is enough evidence to claim that population mean μ_{GAMS} is less than μ_{RANDOM}, population mean μ_{GAMS} is greater than $\mu_{PERMUTATION}$, and population mean $\mu_{PERMUTATION}$ is less than μ_{RANDOM}, at the 0.05 significance level.

Based on our experimental analysis, among the three solution techniques, best solutions are obtained from Permutation based total enumerative heuristic technique. It is sure that, with increasing problem size, the required computation time will also increase, so that it will not be practical to use in larger size FLPs. In addition to this, also limited-time GAMS software produced better solutions that Random sampling based heuristic technique.Surprisingly, the 95% confidence interval for the differences between GAMS and Random sampling based heuristic technique includes zero. Recall that, Random sampling heuristic is just considering a small portion of all possible permutations, the computation time requirement is the least

among the considered three techniques. This highlights to a significant potential of Random sampling based heuristic technique to reach acceptably good solutions within a reasonable time.

REFERENCES

Afrazeh, A., Keivani, A., & Farahani, L. N. (2010). A new model for dynamic multi floor facility layout problem. *Advanced Modeling and Optimization*, 249-256.

Arikaran, P., Jayabalan, V., & Senthilkumar, R. (2010). Analysis of Unequal Areas Facility Layout Problems. *International Journal of Engineering (IJE)*, 44-51.

Bukchin, Y., & Tzur, M. (2014). A new MILP approach for the facility process-layout design problem with rectangular and L/T shape departments. *International Journal of Production Research*, 1-21.

Drira, A., Pierreval, H., & Hajri-Gabouj, S. (2007). Facility layout problems: A survey. *Annual Reviews in Control*, 255-267.

Jadid, A. O., & Firouz, M. (2016). Simulation based approach for solving Unequal Area Facility Layout Problems in Stochastic condition by Genetic Algorithm. *arXiv preprint*, 1608.08321.

Kothari, R., & Ghosh, D. (2011). *The Single Row Facility Layout Problem: State of the Art.* AHMEDABAD: INDIAN INSTITUTE OF MANAGEMENT.

Krishnan, K. K., Jaafari, A. A., Abolhasanpour, M., Hojabri, & Hosein. (2009). A mixed integer programming formulation for multi-floor layout. *African Journal of Business Management*, 616-620.

Montreuil, B., Brotherton, E., & Marcotte, S. (2002). Zone-based facilities layout optimization. *Proceedings of 2002 Industrial Engineering Research Conference*, (pp. 19-22). Orlando.

Niroomand, S. (2013). *Studies on Different Types of Facility Layout Problems (PhD Thesis).* Gazimağusa: Eastern Mediterranean University.

Ozdemir, G., Smith, A. E., & Norman, B. A. (2002). Incorporating Heterogeneous Distance Metrics Within Block Layout Design. *International Journal of Production Research*, 1-18.

Sahoo, N., Shekhar, T., & Sahu, S. (2002). An Approach to multi objective Facility Layout Planning . *INDUSTRIAL ENGINEERING JOURNAL*, 20-25.

Shebanie, C. R. (2004). *AN INTEGRATED, EVOLUTIONARY APPROACH TO FACILITY LAYOUT AND DETAILED DESIGN (Master Thesis).* Pittsburgh: University of Pittsburgh.

Singh, S. P., & Sharma, R. R. (2006). A review of different approaches to the facility layout problems. *International Journal of Advanced Manufacturing Technology*, 425-433.

Soimart, P., & Pongcharoen, P. (2011). Multi-row Machine Layout Design using Aritificial Bee Colony. *International Conference on Economics and Business Information*, V9.

APPENDICES

APPENDIX A – SAMPLE GAMS MODEL OF A PROBLEM TEST INSTANCE

```
Alias(r,r1,r2)

Set N        set of departments  /1*7/

Alias(N,i,j)

Parameter L Length of the halls (meter) /100/

Parameter W Width of the halls (meter)/30/

Set P        set of positions at any row /1*7/

Alias(p,p1,p2)


Parameter A(i) Area requirement for department i /
1       450
2       600
3       200
4       500
5       400
6       550
7       300 /

Table C(i,j) unit transportation cost for ith department to the jth department
1         2         3         4         5         6         7
1       0.00      3.47      2.68      5.16      6.80      6.51      7.87
2       9.15      0.00      8.62      2.20      9.84      8.65      6.43
3       2.31      1.39      0.00      0.81      1.92      2.36      4.10
4       6.61      5.29      2.43      0.00      5.87      7.90      4.12
5       0.39      1.17      6.93      4.77      0.00      7.75      3.64
6       8.37      6.34      7.87      1.29      3.87      0.00      7.01
7       6.81      4.10      9.07      3.01      6.55      4.23      0.00


Table f(i,j) frequency of flow from ith department to the jth department
1        2    3       4       5       6       7
1        0      31      99       8      86       7      68
2       33       0      43      68      23      86      57
3       31      37       0      52      53      84      85
4       16      58      98       0      80      45      56
5       78      40      92      64       0      31      73
6       35      32      63      97      30       0      70
7       96      73      98       4      17      75       0
```

Figure 13: Sample GAMS model of a problem test instance

```
Parameter e a small number;

e=L*0;

Variable Q(i,r,p)   binary variable 1 if ith department is assigned to the pth position at rth row

Variable x(r,p)     x coordinate of centroid location of pth position at rth row

Variable y(r,p)     y coordinate of centroid location of pth position at rth row

Variable z(r,p)     length of department assigned to the pth position at rth row

Variables XP(r1,p1,r2,p2),XM(r1,p1,r2,p2),YP(r1,p1,r2,p2),YM(r1,p1,r2,p2),FD(i,j)

Variable Cost       objective function value

Binary variable Q;

Positive variables x,y,z,XP,XM,YP,YM,FD;

Free variable Cost;

Equations

EQ1
EQ2
EQ3(i)
EQ4(r,p)
EQ6(r)
EQ7(r,p)
EQ8(r,p)
EQ9(r)
EQ10(r,p)
EQ11(r1,r2,p1,p2)
EQ12(r1,r2,p1,p2)
EQ13(r1,r2,p1,p2,i,j)

;

EQ1             .. Cost=e=sum((i,j),c(i,j)*f(i,j)*FD(i,j));

EQ2             ..sum(i,A(i))=l=card(R)*L*W;

EQ3(i)          ..sum((r,p),Q(i,r,p))=e=1;

EQ4(r,p)        ..sum(i,Q(i,r,p))=l=1;

EQ6(r)          ..sum((i,p),A(i)/W*Q(i,r,p))=l=L+e;

EQ7(r,p)        ..x(r,p)=e=W*(card(R)-0.5);

EQ8(r,p)        ..z(r,p)=e=sum(i,A(i)/W*Q(i,r,p));

EQ9(r)          ..y(r,"1")=e=0.5*z(r,"1");
```

Figure 13: Sample GAMS model of a problem test instance (continued)

```
EQ10(r,p)       ..y(r,p)=e=y(r,p-1)+0.5*(z(r,p-1)+z(r,p));

EQ11(r1,r2,p1,p2)..x(r1,p1)-x(r2,p2)=e=XP(r1,p1,r2,p2)-XM(r1,p1,r2,p2);

EQ12(r1,r2,p1,p2)..y(r1,p1)-y(r2,p2)=e=YP(r1,p1,r2,p2)-YM(r1,p1,r2,p2);

EQ13(r1,r2,p1,p2,i,j)..XP(r1,p1,r2,p2)+XM(r1,p1,r2,p2)+YP(r1,p1,r2,p2)+YM(r1,p1,r2,p2)=l=FD(i,j)
+1000000*(2-Q(i,r1,p1)-Q(j,r2,p2));


MODEL MultiRowLayoutProblem /ALL/;

Option Optcr=0.0;

Option Limrow=1000;

OPTION RESLIM = 200;

OPTION ITERLIM = 1000000000;

OPTION work = 16000;

OPTION threads=8;

SOLVE MultiRowLayoutProblem using MIP minimizing Cost;

Display q.l,x.l,y.l,z.l;
```

Figure 13: Sample GAMS model of a problem test instance (continued)

APPENDIX B – VISUAL BASIC CODE TO GENERATE PROBLEM TEST INSTANCES

```
Sub generate()

   Dim R, L, W, TLS, n, Minspa, Maxspa, Mincost, Maxcost, Minfreq, Maxfreq, Epsilon, Unitemp
As Integer
   Dim c, cc As Single
   Dim Area(1 To 25) As Single

   R = Cells(2, 3)
   L = Cells(3, 3)
   W = Cells(4, 3)
   TLS = R * L * W
   n = Cells(6, 3)
   Minspa = Cells(7, 3)
   Maxspa = Cells(8, 3)
   Mincost = Cells(9, 3)
   Maxcost = Cells(10, 3)
   Minfreq = Cells(11, 3)
   Maxfreq = Cells(12, 3)
   Epsilon = Cells(13, 3)
   Unitemp = Cells(14, 3)

   c = 0
   For i = 1 To n
      Area(i) = (Minspa + (Maxspa - Minspa) * Rnd)
      c = c + Area(i)
   Next i

   cc = 0
   For i = 1 To n - 1
      Area(i) = TLS * Area(i) / (c - n * (Unitemp / 2))
      cc = cc + Area(i)
   Next i
   c = 0
   For i = 1 To n - 1
      Area(i) = Int(Area(i) / Unitemp) * Unitemp
      c = c + Area(i)
   Next i

   Area(n) = TLS - c
Range("E2:F26").ClearContents
   For i = 1 To n
Cells(i + 1, 5) = i
Cells(i + 1, 6) = Area(i)
   Next i

Range("I1:AG1").ClearContents
Range("H2:H26").ClearContents
   For i = 1 To n
Cells(1, 8 + i) = i
Cells(1 + i, 8) = i
   Next i
```

Figure 14: Visual basic code to generate test problem instances

```
Range("I2:AG26").ClearContents
   For i = 1 To n
      For j = 1 To n
         If i = j Then
Cells(i + 1, j + 8) = 0
         Else
Cells(i + 1, j + 8) = Int(Mincost + (Maxcost - Mincost) * Rnd * 100) / 100
         End If
      Next j
   Next i

Range("AJ1:BH1").ClearContents
Range("AI2:AI26").ClearContents
   For i = 1 To n
Cells(1, 35 + i) = i
Cells(1 + i, 35) = i
   Next i

Range("AJ2:BH26").ClearContents
   For i = 1 To n
      For j = 1 To n
         If i = j Then
Cells(i + 1, j + 35) = 0
         Else
Cells(i + 1, j + 35) = Int(Minfreq + (Maxfreq - Minfreq) * Rnd)
         End If
      Next j
   Next i
End Sub
```

Figure 14: Visual basic code to generate test problem instances (continued)

APPENDIX C – VISUAL BASIC CODE TO IMPLEMENT PERMUTATION BASED HEURISTIC SOLUTION METHOD

```
Sub solve()
    ' +/- 0 % Gap limit for the deviation between the lengths of the halls.
    Mgap = 200 * Sheets("Generate").Cells(13, 3)
    Sheets("SolveP").Range("A1:Z10000").ClearContents
    Sheets("ReportP").Range("A1:Z10000").ClearContents
    Sheets("Imp_Permute").Range("A1:Z10000").ClearContents

    ' Read the number of departments in the facility.
    ndept = Sheets("Generate").Cells(6, 3)
    Sheets("SolveP").Cells(2, 1) = "Number of Departments"
    Sheets("SolveP").Cells(2, 1).Font.FontStyle = "Bold"
    Sheets("SolveP").Cells(2, 2) = ndept
    Sheets("SolveP").Cells(2, 2).NumberFormat = "#,###;[Red](#,###)"

    ' Read the number of halls in the facility.
    nhall = Sheets("Generate").Cells(2, 3)
    Sheets("SolveP").Cells(3, 1) = "Number of Halls"
    Sheets("SolveP").Cells(3, 1).Font.FontStyle = "Bold"
    Sheets("SolveP").Cells(3, 2) = Format(nhall, "#,###")

    ' Read the length of halls in the facility.
    hallength = Sheets("Generate").Cells(3, 3)
    Sheets("SolveP").Cells(4, 1) = "Length of Halls"
    Sheets("SolveP").Cells(4, 1).Font.FontStyle = "Bold"
    Sheets("SolveP").Cells(4, 2) = Format(hallength, "#,##0.00")


    ' Read the width of halls in the facility.
    hallwidth = Sheets("Generate").Cells(4, 3)
    Sheets("SolveP").Cells(5, 1) = "Width of Halls"
    Sheets("SolveP").Cells(5, 1).Font.FontStyle = "Bold"
    Sheets("SolveP").Cells(5, 2) = Format(hallwidth, "#,##0.00")

    ' Find the total area of the facility.
    totalarea = nhall * hallength * hallwidth
    Sheets("SolveP").Cells(6, 1) = "Total Area"
    Sheets("SolveP").Cells(6, 1).Font.FontStyle = "Bold"
    Sheets("SolveP").Cells(6, 2) = Format(totalarea, "#,##0.00")

    ' Read the felxibility factor
    Sheets("SolveP").Cells(7, 1) = "Flexibility Percentage"
    Sheets("SolveP").Cells(7, 1).Font.FontStyle = "Bold"
    Sheets("SolveP").Cells(7, 2) = Format(Mgap / 2, "#,##0.00")

' Read the area requirements of the departments in the facility.
    ReDim areareq(1 To ndept) As Single
    Sheets("SolveP").Cells(10, 1) = "Department"
    Sheets("SolveP").Cells(10, 1).Font.FontStyle = "Bold"
```

Figure 15: Visual basic code to implement permutation based heuristic solution method

```
Sheets("SolveP").Cells(10, 2) = "Area"
   Sheets("SolveP").Cells(10, 2).Font.FontStyle = "Bold"
   For i = 1 To ndept
      areareq(i) = Sheets("Generate").Cells(i + 1, 6)
      Sheets("SolveP").Cells(10 + i, 1) = i
      Sheets("SolveP").Cells(10 + i, 1).Font.FontStyle = "Bold"
      Sheets("SolveP").Cells(10 + i, 2) = areareq(i)
      Sheets("SolveP").Cells(10 + i, 2).NumberFormat = "#,##0.00_);[Red](#,##0.00)"
   Next i

   ' Read the inter-departmental unit handling cost data.
   ReDim uc(1 To ndept, 1 To ndept) As Single
Sheets("SolveP").Cells(10 + ndept + 5, 1) = "Unit Handling Costs"
   Sheets("SolveP").Cells(10 + ndept + 5, 1).Font.FontStyle = "Bold"
   For j = 1 To ndept
      Sheets("SolveP").Cells(10 + ndept + 5, 1 + j) = j
      Sheets("SolveP").Cells(10 + ndept + 5, 1 + j).NumberFormat = "#,###_);[Red](#,###)"
      Sheets("SolveP").Cells(10 + ndept + 5, 1 + j).Font.FontStyle = "Bold"
   Next j

   For i = 1 To ndept
      Sheets("SolveP").Cells(10 + ndept + 5 + i, 1) = i
      Sheets("SolveP").Cells(10 + ndept + 5 + i, 1).NumberFormat = "#,###_);[Red](#,###)"
      Sheets("SolveP").Cells(10 + ndept + 5 + i, 1).Font.FontStyle = "Bold"
      For j = 1 To ndept
uc(i, j) = Sheets("Generate").Cells(i + 1, 8 + j)
         Sheets("SolveP").Cells(10 + ndept + 5 + i, 1 + j) = uc(i, j)
         Sheets("SolveP").Cells(10 + ndept + 5 + i, 1 + j).NumberFormat =
"#,##0.00_);[Red](#,##0.00)"
      Next j
   Next i

   ' Read the inter-departmental handling frequency data.
   ReDim hf(1 To ndept, 1 To ndept) As Single

Sheets("SolveP").Cells(10 + 2 * ndept + 10, 1) = "Handling Frequencies"
   Sheets("SolveP").Cells(10 + 2 * ndept + 10, 1).Font.FontStyle = "Bold"
   For j = 1 To ndept
      Sheets("SolveP").Cells(10 + 2 * ndept + 10, 1 + j) = j
      Sheets("SolveP").Cells(10 + 2 * ndept + 10, 1 + j).NumberFormat = "#,###_);[Red](#,###)"
      Sheets("SolveP").Cells(10 + 2 * ndept + 10, 1 + j).Font.FontStyle = "Bold"
   Next j

   For i = 1 To ndept
      Sheets("SolveP").Cells(10 + 2 * ndept + 10 + i, 1) = i
      Sheets("SolveP").Cells(10 + 2 * ndept + 10 + i, 1).NumberFormat = "#,###_);[Red](#,###)"
      Sheets("SolveP").Cells(10 + 2 * ndept + 10 + i, 1).Font.FontStyle = "Bold"
      For j = 1 To ndept
hf(i, j) = Sheets("Generate").Cells(i + 1, 35 + j)
         Sheets("SolveP").Cells(10 + 2 * ndept + 10 + i, 1 + j) = hf(i, j)
         Sheets("SolveP").Cells(10 + 2 * ndept + 10 + i, 1 + j).NumberFormat =
"#,##0.00_);[Red](#,##0.00)"
      Next j
   Next i
'-------------------------------------------------
```

Figure 15: Visual basic code to implement permutation based heuristic solution method (continued)

```
' Open file for outout.
   Open "C:\Users\ydurd\Desktop\Permutations.txt" For Output As #1

   ' Set initial permutation {1,2,...,n}
   n = ndept
   W = hallwidth

   ReDim Area(1 To n)
   ReDim L(1 To n)
   ReDim Flow(1 To n, 1 To n)
   ReDim Distance(1 To n, 1 To n)

Total_Area = 0
   Total_Length = 0
   Total_Flow = 0
   For i = 1 To n
      Area(i) = areareq(i)
      Total_Area = Total_Area + Area(i)
      L(i) = Area(i) / W
      Total_Length = Total_Length + L(i)
      For j = 1 To n
Flow(i, j) = uc(i, j) * hf(i, j)
      Next j
   Next i

   ReDim x(n), acculen(n)
   ReDim pmedian(n), min_pmedian(n), Max_pmedian(n)
   ReDim Min_x(n), Max_x(n)
   ReDim xc(n), yc(n)
   ReDim e(n)
   ReDim c(n)
   ReDim hallass(1 To n, 1 To n), min_hallass(1 To n, 1 To n)
   For i = 1 To n
       For j = 1 To n
hallass(i, j) = 0
       Next j
   Next i
   For i = 1 To n
      x(i) = i
   Next i

   pno = 2
   tlen = 0
acculen(0) = 0
   For i = 1 To n
      tlen = tlen + L(x(i))
      acculen(i) = acculen(i - 1) + L(x(i))
   Next i
pmedian(0) = 0
   i = 1
   For j = 1 To n - 1
      limit = (tlen - acculen(pmedian(i - 1))) / (nhall - i + 1)
      If (acculen(j) - acculen(pmedian(i - 1))) <= limit And limit <= (acculen(j + 1) -
acculen(pmedian(i - 1))) Then
```

Figure 15: Visual basic code to implement permutation based heuristic solution method (continued)

```
d1 = Abs(limit - (acculen(j) - acculen(pmedian(i - 1))))
        d2 = Abs(acculen(j + 1) - acculen(pmedian(i - 1)) - limit)
        If d1 < d2 Then
          pmedian(i) = j
          i = i + 1
        Else
          pmedian(i) = j + 1
          i = i + 1
        End If
      End If
    Next j
    If i = n Then pmedian(nhall) = n

e(0) = 0
    hallcolumn = 1
    hallrow = 1
    For i = 1 To n
      If pmedian(hallcolumn - 1) < i And i < pmedian(hallcolumn) Then
hallass(hallrow, hallcolumn) = x(i)
        e(hallrow) = e(hallrow - 1) + L(x(i))
        xc(x(i)) = (hallcolumn - 0.5) * hallwidth
        yc(x(i)) = (e(hallrow) + e(hallrow - 1)) / 2
        hallrow = hallrow + 1
      Else
        If i = pmedian(hallcolumn) Then
hallass(hallrow, hallcolumn) = x(i)
          e(hallrow) = e(hallrow - 1) + L(x(i))
          xc(x(i)) = (hallcolumn - 0.5) * hallwidth
          yc(x(i)) = (e(hallrow) + e(hallrow - 1)) / 2
          hallrow = 1
          hallcolumn = hallcolumn + 1
        End If
      End If
    Next i

    For i = 1 To n
      For j = 1 To n
        Distance(x(i), x(j)) = Abs(xc(x(i)) - xc(x(j))) + Abs(yc(x(i)) - yc(x(j)))
      Next j
    Next i

    Objective = 0
    For i = 1 To n
      For j = 1 To n
        Objective = Objective + Flow(x(i), x(j)) * Distance(x(i), x(j))
      Next j
    Next i

Min_Obj = Objective
    Max_Obj = Objective
    Tot_Obj = 0
    min_gap = 100
    max_gap = 0
    Solution_found = 0
```

Figure 15: Visual basic code to implement permutation based heuristic solution method (continued)

```
  ' Notdun=0 iff current permutation is n, n-1, ..., 1
  notdun = 1
  Satýr = 1
  Do While (notdun)

    tlen = 0
acculen(0) = 0
    For i = 1 To n
      tlen = tlen + L(x(i))
      acculen(i) = acculen(i - 1) + L(x(i))
    Next i

pmedian(0) = 0
    i = 1
    For j = 1 To n - 1
      limit = (tlen - acculen(pmedian(i - 1))) / (nhall - i + 1)
      If (acculen(j) - acculen(pmedian(i - 1))) <= limit And limit <= (acculen(j + 1) -
acculen(pmedian(i - 1))) Then
        d1 = Abs(limit - (acculen(j) - acculen(pmedian(i - 1))))
        d2 = Abs(acculen(j + 1) - acculen(pmedian(i - 1)) - limit)
        If d1 < d2 Then
          pmedian(i) = j
          i = i + 1
        Else
          pmedian(i) = j + 1
          i = i + 1
        End If
      End If
    Next j
    If i = n Then pmedian(nhall) = n

e(0) = 0
    hallcolumn = 1
    hallrow = 1
    For i = 1 To n
      If pmedian(hallcolumn - 1) < i And i < pmedian(hallcolumn) Then
hallass(hallrow, hallcolumn) = x(i)
        e(hallrow) = e(hallrow - 1) + L(x(i))
        xc(x(i)) = (hallcolumn - 0.5) * hallwidth
        yc(x(i)) = (e(hallrow) + e(hallrow - 1)) / 2
        hallrow = hallrow + 1
      Else
        If i = pmedian(hallcolumn) Then
hallass(hallrow, hallcolumn) = x(i)
          e(hallrow) = e(hallrow - 1) + L(x(i))
          xc(x(i)) = (hallcolumn - 0.5) * hallwidth
          yc(x(i)) = (e(hallrow) + e(hallrow - 1)) / 2
          hallrow = 1
          hallcolumn = hallcolumn + 1
        End If
      End If
    Next i
```

Figure 15: Visual basic code to implement permutation based heuristic solution method (continued)

```
    For i = 1 To n
      For j = 1 To n
        Distance(x(i), x(j)) = Abs(xc(x(i)) - xc(x(j))) + Abs(yc(x(i)) - yc(x(j)))
      Next j
    Next i

    Objective = 0
    For i = 1 To n
      For j = 1 To n
        Objective = Objective + Flow(x(i), x(j)) * Distance(x(i), x(j))
      Next j
    Next i

    Tot_Obj = Tot_Obj + Objective
    max_dev = 0
    min_dev = 100
    For i = 1 To nhall
      hall_dev = 100 * (acculen(pmedian(i)) - acculen(pmedian(i - 1))) / hallength
      If hall_dev > max_dev Then max_dev = hall_dev
      If hall_dev < min_dev Then min_dev = hall_dev
    Next i
    Deviation_gap = max_dev - min_dev

    If Objective < Min_Obj And Deviation_gap <= Mgap Then
      Solution_found = 1
      Min_Obj = Objective
      min_pmedian = pmedian
      Min_x = x
      min_devgap = Deviation_gap
      min_hallass = hallass

      ' Print current improved permutation
      Print #1, (pno - 1); Tab(10); ": ";
      For i = 1 To n
        Print #1, x(i); "["; Area(x(i)); "]";
      Next i

      max_dev = 0
      min_dev = 100
      ' Print current pmedian partition
      Print #1, Tab(10); ": ";
      For i = 1 To nhall
        hall_dev = 100 * (acculen(pmedian(i)) - acculen(pmedian(i - 1))) / hallength
        Print #1, "|"; pmedian(i); "["; hall_dev; "]";
      Next i

Print #1, "| Deviation Gap: "; Deviation_gap; "     ";
      ' Print line feed
      Print #1, "Objective:"; Format(Objective, "Standard"); Tab(100); "Minimum Objective:";
Format(Min_Obj, "Standard"); ""
    End If

If Deviation_gap > max_gap Then max_gap = Deviation_gap
    If Deviation_gap < min_gap Then min_gap = Deviation_gap
```

Figure 15: Visual basic code to implement permutation based heuristic solution method (continued)

```
If Objective > Max_Obj Then
        Max_Obj = Objective
      End If

      ' Find next permutation and note whether it is the final one
      notdun = permute(x(), n)
      pno = pno + 1
      t = ((pno / 1000) - Int(pno / 1000))
      If t = 0 Then
        Sheets("SolveP").Cells(1, 2) = Format(pno, "#.###")
        Sheets("Imp_Permute").Cells(Satýr, 1) = pno
        Sheets("Imp_Permute").Cells(Satýr, 2) = Min_Obj
        Sheets("Imp_Permute").Cells(Satýr, 3) = min_devgap / 2
        Sheets("Imp_Permute").Cells(Satýr, 4) = Max_Obj
        Sheets("Imp_Permute").Cells(Satýr, 5) = max_gap / 2
        Sheets("Imp_Permute").Cells(Satýr, 6) = min_gap / 2
        Satýr = Satýr + 1
       End If
    Loop

    Avg_Obj = Tot_Obj / (pno - 1)
    Print #1, "A total of "; pno - 1; " enumeration iterations performed."
    Print #1, ""
    Sheets("ReportP").Cells(1, 1) = "Total Number of Iteration:"
    Sheets("ReportP").Cells(1, 1).Font.FontStyle = "Bold"
    Sheets("ReportP").Cells(1, 2) = Format((pno - 1), "#.###")

    Sheets("ReportP").Cells(3, 1) = "Maximum Objective Value:"
    Sheets("ReportP").Cells(3, 1).Font.FontStyle = "Bold"
    Sheets("ReportP").Cells(3, 2) = Format(Max_Obj, "#,##0.00")

    Sheets("ReportP").Cells(4, 1) = "Maximum Deviation +/- (%):"
    Sheets("ReportP").Cells(4, 1).Font.FontStyle = "Bold"
    Sheets("ReportP").Cells(4, 2) = Format((max_gap / 2), "#,##0.00")

    Sheets("ReportP").Cells(5, 1) = "Minimum Deviation +/- (%):"
    Sheets("ReportP").Cells(5, 1).Font.FontStyle = "Bold"
    Sheets("ReportP").Cells(5, 2) = Format((min_gap / 2), "#,##0.00")

    Sheets("ReportP").Cells(6, 1) = "Average Objective Value:"
    Sheets("ReportP").Cells(6, 1).Font.FontStyle = "Bold"
    Sheets("ReportP").Cells(6, 2) = Format(Avg_Obj, "#,##0.00")

    Sheets("ReportP").Cells(8, 1) = "Minimum Objective Value:"
    Sheets("ReportP").Cells(8, 1).Font.FontStyle = "Bold"
Sheets("ReportP").Cells(8, 2) = Format(Min_Obj, "#,##0.00")

    Sheets("ReportP").Cells(9, 1) = "Minimum Deviation +/- (%):"
    Sheets("ReportP").Cells(9, 1).Font.FontStyle = "Bold"
    Sheets("ReportP").Cells(9, 2) = Format(min_devgap / 2, "#,##0.00")

    If Solution_found = 1 Then
      Sheets("ReportP").Cells(12, 1) = "Minimum Permutation Solution:"
      Sheets("ReportP").Cells(12, 1).Font.FontStyle = "Bold"
```

Figure 15: Visual basic code to implement permutation based heuristic solution method (continued)

```
    For i = 1 To n
       Sheets("ReportP").Cells(11, 1 + i) = i
       Sheets("ReportP").Cells(11, 1 + i).Font.FontStyle = "Bold"
       Sheets("ReportP").Cells(12, 1 + i) = Format(Min_x(i), "#,###")
     Next i

     Sheets("ReportP").Cells(15, 1) = "P-median Partition:"
     Sheets("ReportP").Cells(15, 1).Font.FontStyle = "Bold"

     For i = 1 To nhall
       Sheets("ReportP").Cells(14, 1 + i) = i
       Sheets("ReportP").Cells(14, 1 + i).Font.FontStyle = "Bold"
       Sheets("ReportP").Cells(15, 1 + i) = Format(min_pmedian(i), "#,##0")
     Next i

     Sheets("ReportP").Cells(17, 1) = "Deviation +/- (%):"
     Sheets("ReportP").Cells(17, 1).Font.FontStyle = "Bold"
     Sheets("ReportP").Cells(17, 2) = Format((min_devgap / 2), "#,##0")

     ReDim hall_area(1 To nhall)

     For i = 1 To nhall
       hall_area(i) = 0
     Next i

     Sheets("ReportP").Cells(20, 1) = "Block Plan:"
     Sheets("ReportP").Cells(20, 1).Font.FontStyle = "Bold"
     For i = 1 To nhall
       hall_area(i) = 0
       For j = 1 To min_pmedian(i) - min_pmedian(i - 1)
         Sheets("ReportP").Cells(20 + j, i + 1) = Format(min_hallass(j, i), "#,##0")
         hall_area(i) = hall_area(i) + Area(min_hallass(j, i))
       Next j
     Next i

     Sheets("ReportP").Cells(27, 1) = "Areas:"
     Sheets("ReportP").Cells(27, 1).Font.FontStyle = "Bold"

     For i = 1 To nhall
       Sheets("ReportP").Cells(27, 1 + i) = Format(hall_area(i), "#,##0.00")
       Sheets("ReportP").Cells(27, 1 + i).Font.FontStyle = "Bold"
       For j = 1 To min_pmedian(i) - min_pmedian(i - 1)
         Sheets("ReportP").Cells(27 + j, i + 1) = Format(Area(min_hallass(j, i)), "#,##0.00")
       Next j
      Next i
    Else
      Sheets("ReportP").Cells(12, 1) = "NO FEASIBLE SOLUTIONS FOUND FOR THE GIVEN
FLEXIBILITY PERCENTAGE !!!"
    End If
   Close

End Sub
```

Figure 15: Visual basic code to implement permutation based heuristic solution method (continued)

```
Function permute(x(), n)
  'Creates the next permutation in the "natural sequence"
  'Returns 0 if permutation is n, n-1, ..., 1
  'Default is to return 1
  permute = 1
  bigfix = n
  'Done = 1 indicates next permutation is complete, 0 not.
  done = 0
  Do While (done = 0)
    done = 1
    'Find the index of bigfix
    For i = 1 To n
      If x(i) = bigfix Then bigindx = i
    Next i
    descend = 1
    If bigindx <> n Then
      For i = bigindx To n - 1
        If x(i) <x(i + 1) Then descend = 0
      Next i
    End If
    If descend And bigindx = 1 Then permute = 0
    If descend Then
      'Work left
      current = x(bigindx - 1)
      candidx = bigindx
      'Find element to switch with x(bigindx-1)
      For i = bigindx To n
        If x(i) > current And x(i) < x(candidx) Then candidx = i
      Next i 'Switch them
      temp = x(candidx)
      x(candidx) = x(bigindx - 1)
x(bigindx - 1) = temp
      temp = sort(x(), bigindx)
    End If
    'End of work left

    'Work right
    If descend = 0 Then
      done = 0
      bigfix = findlarg(x(), bigindx + 1)
    End If
    'End of work right
  Loop
End Function

Function findlarg(x(), start)
  'Finds largest x(i) from i = start to i = n
  candid = x(start)
  ub = UBound(x)
  For i = start To ub
    If x(i) >candid Then candid = x(i)
  Next i
  findlarg = candid
End Function
```

Figure 15: Visual basic code to implement permutation based heuristic solution method (continued)

```
Function sort(x(), start)
  'Sorts x() from i = start to i = n
  ub = UBound(x)
  For i = start To ub
    For j = i To ub
      If x(i) > x(j) Then
        temp = x(i)
        x(i) = x(j)
        x(j) = temp
      End If
    Next j
  Next i
End Function
```

Figure 15: Visual basic code to implement permutation based heuristic solution method (continued)

APPENDIX D – VISUAL BASIC CODE TO IMPLEMENT RANDOM SAMLING BASED HEURISTIC SOLUTION METHOD

```
Sub random()
    ' +/- 0 % Gap limit for the deviation between the lengths of the halls.
    Mgap = 200 * Sheets("Generate").Cells(13, 3)

    Sheets("SolveR").Range("A1:Z10000").ClearContents
    Sheets("ReportR").Range("A1:Z10000").ClearContents
    Sheets("Imp_Random").Range("A1:Z10000").ClearContents

    ' Read the number of departments in the facility.
    ndept = Sheets("Generate").Cells(6, 3)
    Sheets("SolveR").Cells(2, 1) = "Number of Departments"
    Sheets("SolveR").Cells(2, 1).Font.FontStyle = "Bold"
    Sheets("SolveR").Cells(2, 2) = ndept
    Sheets("SolveR").Cells(2, 2).NumberFormat = "#,###;[Red](#,###)"

    ' Read the number of halls in the facility.
    nhall = Sheets("Generate").Cells(2, 3)
    Sheets("SolveR").Cells(3, 1) = "Number of Halls"
    Sheets("SolveR").Cells(3, 1).Font.FontStyle = "Bold"
    Sheets("SolveR").Cells(3, 2) = Format(nhall, "#,###")

    ' Read the length of halls in the facility.
    hallength = Sheets("Generate").Cells(3, 3)
    Sheets("SolveR").Cells(4, 1) = "Length of Halls"
    Sheets("SolveR").Cells(4, 1).Font.FontStyle = "Bold"
    Sheets("SolveR").Cells(4, 2) = Format(hallength, "#,##0.00")

    ' Read the width of halls in the facility.
    hallwidth = Sheets("Generate").Cells(4, 3)
    Sheets("SolveR").Cells(5, 1) = "Width of Halls"
    Sheets("SolveR").Cells(5, 1).Font.FontStyle = "Bold"
    Sheets("SolveR").Cells(5, 2) = Format(hallwidth, "#,##0.00")

    ' Find the total area of the facility.
    totalarea = nhall * hallength * hallwidth
    Sheets("SolveR").Cells(6, 1) = "Total Area"
    Sheets("SolveR").Cells(6, 1).Font.FontStyle = "Bold"
    Sheets("SolveR").Cells(6, 2) = Format(totalarea, "#,##0.00")

    ' Read the felxibility factor
    Sheets("SolveR").Cells(7, 1) = "Flexibility Percentage"
    Sheets("SolveR").Cells(7, 1).Font.FontStyle = "Bold"
    Sheets("SolveR").Cells(7, 2) = Format(Mgap / 2, "#,##0.00")

    ' Read the area requirements of the departments in the facility.
    ReDim areareq(1 To ndept) As Single
    Sheets("SolveR").Cells(10, 1) = "Department"
    Sheets("SolveR").Cells(10, 1).Font.FontStyle = "Bold"
```

Figure 16: Visual basic code to implement random sampling based heuristic solution method

```
    Sheets("SolveR").Cells(10, 2) = "Area"
    Sheets("SolveR").Cells(10, 2).Font.FontStyle = "Bold"
    For i = 1 To ndept
       areareq(i) = Sheets("Generate").Cells(i + 1, 6)
       Sheets("SolveR").Cells(10 + i, 1) = i
       Sheets("SolveR").Cells(10 + i, 1).Font.FontStyle = "Bold"
       Sheets("SolveR").Cells(10 + i, 2) = areareq(i)
       Sheets("SolveR").Cells(10 + i, 2).NumberFormat = "#,##0.00_);[Red](#,##0.00)"
    Next i

    ' Read the inter-departmental unit handling cost data.
    ReDim uc(1 To ndept, 1 To ndept) As Single

    Sheets("SolveR").Cells(10 + ndept + 5, 1) = "Unit Handling Costs"
    Sheets("SolveR").Cells(10 + ndept + 5, 1).Font.FontStyle = "Bold"
    For j = 1 To ndept
       Sheets("SolveR").Cells(10 + ndept + 5, 1 + j) = j
       Sheets("SolveR").Cells(10 + ndept + 5, 1 + j).NumberFormat = "#,###_);[Red](#,###)"
       Sheets("SolveR").Cells(10 + ndept + 5, 1 + j).Font.FontStyle = "Bold"
    Next j

    For i = 1 To ndept
       Sheets("SolveR").Cells(10 + ndept + 5 + i, 1) = i
       Sheets("SolveR").Cells(10 + ndept + 5 + i, 1).NumberFormat = "#,###_);[Red](#,###)"
       Sheets("SolveR").Cells(10 + ndept + 5 + i, 1).Font.FontStyle = "Bold"
       For j = 1 To ndept
uc(i, j) = Sheets("Generate").Cells(i + 1, 8 + j)
          Sheets("SolveR").Cells(10 + ndept + 5 + i, 1 + j) = uc(i, j)
          Sheets("SolveR").Cells(10 + ndept + 5 + i, 1 + j).NumberFormat =
"#,##0.00_);[Red](#,##0.00)"
       Next j
    Next i

    ' Read the inter-departmental handling frequency data.
    ReDim hf(1 To ndept, 1 To ndept) As Single

    Sheets("SolveR").Cells(10 + 2 * ndept + 10, 1) = "Handling Frequencies"
    Sheets("SolveR").Cells(10 + 2 * ndept + 10, 1).Font.FontStyle = "Bold"
    For j = 1 To ndept
       Sheets("SolveR").Cells(10 + 2 * ndept + 10, 1 + j) = j
       Sheets("SolveR").Cells(10 + 2 * ndept + 10, 1 + j).NumberFormat = "#,###_);[Red](#,###)"
       Sheets("SolveR").Cells(10 + 2 * ndept + 10, 1 + j).Font.FontStyle = "Bold"
    Next j

    For i = 1 To ndept
       Sheets("SolveR").Cells(10 + 2 * ndept + 10 + i, 1) = i
       Sheets("SolveR").Cells(10 + 2 * ndept + 10 + i, 1).NumberFormat = "#,###_);[Red](#,###)"
       Sheets("SolveR").Cells(10 + 2 * ndept + 10 + i, 1).Font.FontStyle = "Bold"
       For j = 1 To ndept
hf(i, j) = Sheets("Generate").Cells(i + 1, 35 + j)
          Sheets("SolveR").Cells(10 + 2 * ndept + 10 + i, 1 + j) = hf(i, j)
          Sheets("SolveR").Cells(10 + 2 * ndept + 10 + i, 1 + j).NumberFormat =
"#,##0.00_);[Red](#,##0.00)"
       Next j
    Next i
```

Figure 16: Visual basic code to implement random sampling based heuristic solution method (continued)

```
'-----------------------------------------------
  ' Open file for outout.
  Open "C:\Users\ydurd\DesktopPermutations.txt" For Output As #1

  ' Set initial permutation {1,2,...,n}
  n = ndept
  W = hallwidth
  ReDim Area(1 To n)
  ReDim L(1 To n)
  ReDim Flow(1 To n, 1 To n)
  ReDim Distance(1 To n, 1 To n)

  Total_Area = 0
  Total_Length = 0
  Total_Flow = 0
  For i = 1 To n
    Area(i) = areareq(i)
    Total_Area = Total_Area + Area(i)
    L(i) = Area(i) / W
    Total_Length = Total_Length + L(i)
    For j = 1 To n
Flow(i, j) = uc(i, j) * hf(i, j)
    Next j
  Next i

  ReDim x(n), acculen(n)
  ReDim pmedian(n), min_pmedian(n), Max_pmedian(n)
  ReDim Min_x(n), Max_x(n)
  ReDim xc(n), yc(n)
  ReDim e(n)
  ReDim c(n)
  ReDim hallass(1 To n, 1 To n), min_hallass(1 To n, 1 To n)
  For i = 1 To n
     For j = 1 To n
hallass(i, j) = 0
     Next j
  Next i
  For i = 1 To n
    x(i) = i
  Next i

  pno = 2
  tlen = 0
acculen(0) = 0
  For i = 1 To n
    tlen = tlen + L(x(i))
    acculen(i) = acculen(i - 1) + L(x(i))
  Next i

pmedian(0) = 0
  i = 1
For j = 1 To n - 1
    limit = (tlen - acculen(pmedian(i - 1))) / (nhall - i + 1)
    If (acculen(j) - acculen(pmedian(i - 1))) <= limit And limit <= (acculen(j + 1) -
acculen(pmedian(i - 1))) Then
```

Figure 16: Visual basic code to implement random sampling based heuristic solution method (continued)

```
        d1 = Abs(limit - (acculen(j) - acculen(pmedian(i - 1))))
        d2 = Abs(acculen(j + 1) - acculen(pmedian(i - 1)) - limit)
        If d1 < d2 Then
          pmedian(i) = j
          i = i + 1
        Else
          pmedian(i) = j + 1
          i = i + 1
        End If
      End If
    Next j
    If i = n Then pmedian(nhall) = n

e(0) = 0
    hallcolumn = 1
    hallrow = 1
    For i = 1 To n
      If pmedian(hallcolumn - 1) < i And i < pmedian(hallcolumn) Then
hallass(hallrow, hallcolumn) = x(i)
        e(hallrow) = e(hallrow - 1) + L(x(i))
        xc(x(i)) = (hallcolumn - 0.5) * hallwidth
        yc(x(i)) = (e(hallrow) + e(hallrow - 1)) / 2
        hallrow = hallrow + 1
      Else
        If i = pmedian(hallcolumn) Then
hallass(hallrow, hallcolumn) = x(i)
          e(hallrow) = e(hallrow - 1) + L(x(i))
          xc(x(i)) = (hallcolumn - 0.5) * hallwidth
          yc(x(i)) = (e(hallrow) + e(hallrow - 1)) / 2
          hallrow = 1
          hallcolumn = hallcolumn + 1
        End If
      End If
    Next i

    For i = 1 To n
      For j = 1 To n
        Distance(x(i), x(j)) = Abs(xc(x(i)) - xc(x(j))) + Abs(yc(x(i)) - yc(x(j)))
      Next j
    Next i
    Objective = 0
    For i = 1 To n
      For j = 1 To n
        Objective = Objective + Flow(x(i), x(j)) * Distance(x(i), x(j))
      Next j
    Next i

Min_Obj = Objective
    Max_Obj = Objective
    Tot_Obj = 0
    min_gap = 100
    max_gap = 0
    Solution_found = 0
    Satir = 1
```

Figure 16: Visual basic code to implement random sampling based heuristic solution method (continued)

```
  ' Determine maximum number of iterations for random sampling.
  max_it = 40000 * n
  If n >= 10 Then max_it = 400000 * (n / 10)

  For it0 = 1 To max_it

    tlen = 0
acculen(0) = 0
    For i = 1 To n
      tlen = tlen + L(x(i))
      acculen(i) = acculen(i - 1) + L(x(i))
    Next i

pmedian(0) = 0
    i = 1
    For j = 1 To n - 1
      limit = (tlen - acculen(pmedian(i - 1))) / (nhall - i + 1)
      If (acculen(j) - acculen(pmedian(i - 1))) <= limit And limit <= (acculen(j + 1) -
acculen(pmedian(i - 1))) Then
        d1 = Abs(limit - (acculen(j) - acculen(pmedian(i - 1))))
        d2 = Abs(acculen(j + 1) - acculen(pmedian(i - 1)) - limit)
        If d1 < d2 Then
          pmedian(i) = j
          i = i + 1
        Else
          pmedian(i) = j + 1
          i = i + 1
        End If
      End If
    Next j
    If i = n Then pmedian(nhall) = n

e(0) = 0
    hallcolumn = 1
    hallrow = 1

For i = 1 To n
      If pmedian(hallcolumn - 1) < i And i < pmedian(hallcolumn) Then
hallass(hallrow, hallcolumn) = x(i)
        e(hallrow) = e(hallrow - 1) + L(x(i))
        xc(x(i)) = (hallcolumn - 0.5) * hallwidth
        yc(x(i)) = (e(hallrow) + e(hallrow - 1)) / 2
        hallrow = hallrow + 1
      Else
        If i = pmedian(hallcolumn) Then
hallass(hallrow, hallcolumn) = x(i)
          e(hallrow) = e(hallrow - 1) + L(x(i))
          xc(x(i)) = (hallcolumn - 0.5) * hallwidth
          yc(x(i)) = (e(hallrow) + e(hallrow - 1)) / 2
          hallrow = 1
          hallcolumn = hallcolumn + 1
        End If
      End If
    Next i
```

Figure 16: Visual basic code to implement random sampling based heuristic solution method (continued)

```
    For i = 1 To n
      For j = 1 To n
        Distance(x(i), x(j)) = Abs(xc(x(i)) - xc(x(j))) + Abs(yc(x(i)) - yc(x(j)))
      Next j
    Next i

    Objective = 0
    For i = 1 To n
      For j = 1 To n
        Objective = Objective + Flow(x(i), x(j)) * Distance(x(i), x(j))
      Next j
    Next i

    Tot_Obj = Tot_Obj + Objective

    max_dev = 0
    min_dev = 100
    For i = 1 To nhall
      hall_dev = 100 * (acculen(pmedian(i)) - acculen(pmedian(i - 1))) / hallength
      If hall_dev > max_dev Then max_dev = hall_dev
      If hall_dev < min_dev Then min_dev = hall_dev
    Next i
    Deviation_gap = max_dev - min_dev

    If Objective < Min_Obj And Deviation_gap <= Mgap Then
      Solution_found = 1
      Min_Obj = Objective
      min_pmedian = pmedian
      Min_x = x
      min_devgap = Deviation_gap
      min_hallass = hallass

      ' Print current improved permutation
      Print #1, (pno - 1); Tab(10); ": ";
      For i = 1 To n
        Print #1, x(i); "["; Area(x(i)); "]";
      Next i

max_dev = 0
      min_dev = 100
      ' Print current pmedian partition
      Print #1, Tab(10); ": ";
      For i = 1 To nhall
        hall_dev = 100 * (acculen(pmedian(i)) - acculen(pmedian(i - 1))) / hallength
        Print #1, "|"; pmedian(i); "["; hall_dev; "]";
      Next i

      Print #1, "| Deviation Gap: "; Deviation_gap; "     ";

      ' Print line feed
      Print #1, "Objective:"; Format(Objective, "Standard"); Tab(100); "Minimum Objective:";
Format(Min_Obj, "Standard"); ""
    End If
```

Figure 16: Visual basic code to implement random sampling based heuristic solution method (continued)

```
If Deviation_gap > max_gap Then max_gap = Deviation_gap
    If Deviation_gap < min_gap Then min_gap = Deviation_gap

    If Objective > Max_Obj Then
      Max_Obj = Objective
    End If

    ' Find next random sample
    ReDim x_prob(1 To n)
    For i = 1 To n
       x_prob(i) = Rnd
    Next i

    sorted = 1
    Do While sorted = 1
      sorted = 0
      For i = 1 To n - 1
        If x_prob(i) > x_prob(i + 1) Then
          dummy = x(i)
          x(i) = x(i + 1)
x(i + 1) = dummy
          dummy = x_prob(i)
          x_prob(i) = x_prob(i + 1)
          x_prob(i + 1) = dummy
          sorted = 1
        End If
      Next i
    Loop

    pno = pno + 1
t = ((pno / 1000) - Int(pno / 1000))
    If t = 0 Then
      Sheets("SolveR").Cells(1, 2) = Format((pno - 1), "#.###")
      Sheets("Imp_Random").Cells(Satir, 1) = pno
      Sheets("Imp_Random").Cells(Satir, 2) = Min_Obj
      Sheets("Imp_Random").Cells(Satir, 3) = min_devgap / 2
      Sheets("Imp_Random").Cells(Satir, 4) = Max_Obj
      Sheets("Imp_Random").Cells(Satir, 5) = max_gap / 2
      Sheets("Imp_Random").Cells(Satir, 6) = min_gap / 2
      Satir = Satir + 1
     End If
  Next it0

  Avg_Obj = Tot_Obj / (pno - 1)
  Print #1, "A total of "; pno - 1; " enumeration iterations performed."
  Print #1, ""

  Sheets("ReportR").Cells(1, 1) = "Total Number of Iteration:"
  Sheets("ReportR").Cells(1, 1).Font.FontStyle = "Bold"
  Sheets("ReportR").Cells(1, 2) = Format((pno - 1), "#.###")

  Sheets("ReportR").Cells(3, 1) = "Maximum Objective Value:"
  Sheets("ReportR").Cells(3, 1).Font.FontStyle = "Bold"
  Sheets("ReportR").Cells(3, 2) = Format(Max_Obj, "#,##0.00")
```

Figure 16: Visual basic code to implement random sampling based heuristic solution method (continued)

```
Sheets("ReportR").Cells(4, 1) = "Maximum Deviation +/- (%):"
  Sheets("ReportR").Cells(4, 1).Font.FontStyle = "Bold"
  Sheets("ReportR").Cells(4, 2) = Format((max_gap / 2), "#,##0.00")

  Sheets("ReportR").Cells(5, 1) = "Minimum Deviation +/- (%):"
  Sheets("ReportR").Cells(5, 1).Font.FontStyle = "Bold"
  Sheets("ReportR").Cells(5, 2) = Format((min_gap / 2), "#,##0.00")

  Sheets("ReportR").Cells(6, 1) = "Average Objective Value:"
  Sheets("ReportR").Cells(6, 1).Font.FontStyle = "Bold"
  Sheets("ReportR").Cells(6, 2) = Format(Avg_Obj, "#,##0.00")

  Sheets("ReportR").Cells(8, 1) = "Minimum Objective Value:"
  Sheets("ReportR").Cells(8, 1).Font.FontStyle = "Bold"
  Sheets("ReportR").Cells(8, 2) = Format(Min_Obj, "#,##0.00")

  Sheets("ReportR").Cells(9, 1) = "Minimum Deviation +/- (%):"
  Sheets("ReportR").Cells(9, 1).Font.FontStyle = "Bold"
  Sheets("ReportR").Cells(9, 2) = Format(min_devgap / 2, "#,##0.00")

  If Solution_found = 1 Then
    Sheets("ReportR").Cells(12, 1) = "Minimum Permutation Solution:"
    Sheets("ReportR").Cells(12, 1).Font.FontStyle = "Bold"

    For i = 1 To n
      Sheets("ReportR").Cells(11, 1 + i) = i
      Sheets("ReportR").Cells(11, 1 + i).Font.FontStyle = "Bold"
      Sheets("ReportR").Cells(12, 1 + i) = Format(Min_x(i), "#,###")
    Next i

    Sheets("ReportR").Cells(15, 1) = "P-median Partition:"
    Sheets("ReportR").Cells(15, 1).Font.FontStyle = "Bold"

    For i = 1 To nhall
      Sheets("ReportR").Cells(14, 1 + i) = i
      Sheets("ReportR").Cells(14, 1 + i).Font.FontStyle = "Bold"
      Sheets("ReportR").Cells(15, 1 + i) = Format(min_pmedian(i), "#,##0")
    Next i

    Sheets("ReportR").Cells(17, 1) = "Deviation +/- (%):"
    Sheets("ReportR").Cells(17, 1).Font.FontStyle = "Bold"
    Sheets("ReportR").Cells(17, 2) = Format((min_devgap / 2), "#,##0")

    ReDim hall_area(1 To nhall)

    For i = 1 To nhall
      hall_area(i) = 0
    Next i

    Sheets("ReportR").Cells(20, 1) = "Block Plan:"
    Sheets("ReportR").Cells(20, 1).Font.FontStyle = "Bold"
```

Figure 16: Visual basic code to implement random sampling based heuristic solution method (continued)

```
    For i = 1 To nhall
      hall_area(i) = 0
      For j = 1 To min_pmedian(i) - min_pmedian(i - 1)
        Sheets("ReportR").Cells(20 + j, i + 1) = Format(min_hallass(j, i), "#,##0")
        hall_area(i) = hall_area(i) + Area(min_hallass(j, i))
      Next j
    Next i

    Sheets("ReportR").Cells(27, 1) = "Areas:"
    Sheets("ReportR").Cells(27, 1).Font.FontStyle = "Bold"

    For i = 1 To nhall
      Sheets("ReportR").Cells(27, 1 + i) = Format(hall_area(i), "#,##0.00")
      Sheets("ReportR").Cells(27, 1 + i).Font.FontStyle = "Bold"
      For j = 1 To min_pmedian(i) - min_pmedian(i - 1)
        Sheets("ReportR").Cells(27 + j, i + 1) = Format(Area(min_hallass(j, i)), "#,##0.00")
      Next j
    Next i
  Else
    Sheets("ReportR").Cells(12, 1) = "NO FEASIBLE SOLUTIONS FOUND FOR THE GIVEN
FLEXIBILITY PERCENTAGE !!!"
  End If

  Close
End Sub
```

Figure 16: Visual basic code to implement random sampling based heuristic solution method (continued)

CURRICULUM VITAE

PERSONAL INFORMATION

Surname, Name: Suoad Y. Ali El.MAGSSABI
Date and Place of Birth: 25 March 1973, Benghazi
Marital Status: Married
Email: souad1973.2014@gmail.com

EDUCATION

Degree	Institution	Year of Graduation
M.Sc.	ÇANKAYA University Department of Industrial Engineering	2018
B.Sc.	University of BENGHAZI Faculty of Engineering, Department of Industrial Engineering (Benghazi – Libya)	2002
High School	Benghazi High School	1992

FOREIN LANGUAGES
English

HOBBIES
Reading, Design, Swimming