Ranjan Kumar Hasda

UNIVERSITE BRETAGNE LOIRE

CENTRALE NANTES

Thèse de Doctorat

Ranjan Kumar HASDA

Mémoire présenté en vue de l'obtention
du grade de Docteur de l'Ecole Centrale de Nantes
Sous le sceau de l'UNIVERSITÉ BRETAGNE LOIRE

École doctorale : Sciences pour l'Ingénieur, Géosciences, Architecture

Discipline : Génie mécanique
Unité de recherche : **Laboratoire des Sciences du Numérique de Nantes**

Soutenue le : 17 octobre 2017

Contribution to the optimization of Unequal Area Rectangular Facility Layout Problem

JURY

Président :	**FOURQUET Jean-Yves**, Professeur, Ecole Nationale d'Ingénieurs de Tarbes
Rapporteurs :	**POISSON Gérard**, Professeur, IIT de Bourges, Université d'Orléans **FISCHER Xavier**, Professeur, Ecole Supérieure des Technologies Industrielles Avancés, Bidart
Examinateurs:	**TARALOVA Ina**, Maitre de conférences, Ecole Centrale de Nantes **BHATTACHARJYA Rajib Kumar,** Professeur, Indian Institute of Technology Guwahati
Invité:	**RAVAUT Yannick**, Docteur-Ingénieur, Thales Communications & Security, Cholet
Directeur de thèse :	**BENNIS Fouad**, Professeur, Ecole Centrale de Nantes

CONTRIBUTION TO THE OPTIMIZATION OF UNEQUAL AREA RECTANGULAR FACILITY LAYOUT PROBLEM

A Thesis

Presented to

École Doctorale SPIGA « Sciences Pour l'Ingénieur, Géosciences, Architecture »

LS2N « Laboratoire des Sciences du Numérique de Nantes »

Ecole Centrale de Nantes, Nantes, France

In Partial Fulfillment

of the Requirements for the Degree

Doctorate

---

by

Ranjan Kumar Hasda

2017

Keywords: Facility Layout, Continuous Optimization, Discrete Optimization, Greedy Method, Hybrid Construction and Improvement Approach

COMMITTEE MEMBERS

Président :

Directeur de thèse : Fouad Bennis, Doctorate

Professeur de génie mécanique

Ecole Centrale de Nantes, Nantes

Rapporteur : Damien Chablat, Doctorate

Professor of Robotics

Ecole des Mines de Nantes

Rapporteur : Laurent Pino, Doctorate

Maître de conférences

École nationale d'ingénieurs de Brest

ABSTRACT

A facility layout design is one of the most commonly faced problems in the manufacturing sectors. The problem is mixed-integer in nature and usually an NP-hard problem, which makes it difficult to solve using classical optimization techniques, which are better for local search. To overcome these limitations, two algorithms have been proposed for solving static facility layout problems with the unequal size compartments. The objective function of the problems considered is nonlinear in which the sum of the material handling cost has been minimized.

In the first approach, a hybrid constructive and improvement model has been proposed where an advanced bottom-left fill technique was used as constructive approach. The constructive model proposed also acts as a local search method based on greedy algorithm. For improvement approach a hybrid genetic algorithm has been proposed, where the crossover and mutation operator are specially designed to handle the solution representation which itself is used as constructive model.

In the second approach, a combined local and global search model was proposed where a rotation operator was used to avoid mixed-integer formulation of the problem. Use of rotation operator has also reduced the number of variables significantly. Apart from the conventional evolutionary operators this model has also used exchange and rotation operators.

The performances of both model are tested over a previously solved problem selected from the literature. The evaluation of the results shows that the performances of the proposed models are better than many existing algorithms and has the potential for field applications.

ACKNOWLEDGMENTS

I would like to express my sincere gratitude and appreciation to my supervisor, Professor Fouad Bennis, for his guidance, encouragement, constructive critiques, enthusiasm and patience throughout this research. He has been a great mentor and guide to me. This thesis would not have been possible without him.

I am also grateful to visiting Professor Rajib Bhattacharjya from IIT Guwahati, for his valuable comments, insights and feedback at LS2N. I benefited greatly from his both professional and less professional discussions, giving me both inspiration and encouragement. I am also thankful to Ramakrishnan Ramachandran, Post. Doctorate and Prabandh Chakraborty, Masters, for their valuable suggestions during my thesis work.

I am also greatly indebted to generous financial support from India4EU II, Erasmus Mundus Action 2 project funded by the European Commission and Institut de Recherche en Communications et Cybernétique de Nantes, Ecole Centrale de Nantes

Above all, I wish to thank my mother Kanaka Hasda, my brother Santosh Kumar Hasda and my sister Sushma Hasda for their constant moral support, encouragement and love and my friends Ritanjali Sethy, Deep Seth and Ranjan Jha and others for their constant support and help in all my endeavors.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1:

INTRODUCTION

1.1. Motivation

The continuous evolution of worldly-wise and extensive applications such as Buildings, Circuits and Human-Machine Interfaces design has given rise to a strong appeal in formulating and automating layout design algorithms and guidelines. Many areas of Operations Research and Decision Sciences has been motivated from these applications and built up a significant research in formalizing layout design algorithms, preferences, and fitness measures [DOW02; YOU03; TOM10]). However, despite being an active research area, layout design field is still unclearly defined. The available research mostly provides design algorithms and guidelines in a very rigid and simple framework, without a detailed methodology for applying [TOM10]. The usefulness of such immense scattered knowledge is further limited by cognitive limitations of users. To address some imperfection of the existing research, this dissertation presents a new research and solution methodology for undertaking the Layout Design problem. It approaches some important issues faced in layout design by providing means to comprise complex, subjective, and evolving preferences into the design process and fast generation and manipulation of superior layout alternatives.

1.2. Facility Layout Design

In Facility Layout Design, usually the location of the facilities in the manufacturing plants are determined with the aim of finding the best possible arrangement while fulfilling certain criteria or objectives with constraints.

The Facility Layout Design also plays a key role in attaining production efficiency as it directly affects the manufacturing cost, lead times and productivity [KOU92]. The facilities with best possible layout contribute to the overall efficiency of the operations that reduces about 20-50% of the final operating cost [TOM10]. There are different types of facility layout problems that has been detailed in the next chapter. Some researchers also have classified facility layout designs, for example [KUS87]; [CAG98]; [DRI07] and [KUL07].

1.3. Problem Statement

The objective of the research is to develop a generic framework for the layout design. However, we tend towards two-dimensional static unequal area facility problem for analysis and illustration purpose. The formulation of the two-dimensional Layout problem is very difficult which can be easily and largely adapted with certain rules and preferences giving a generic approach to the layout design problems [GAR81]; [DYC90]; [LIG00] and [BUR04]. In the rectangular facility layout problem, the rectangular departments are located in a rectangular layout space in an orthogonal manner. The layout problem has been formulated differently by researchers and some have been in listed in the Section 2.4. The investigation carried out is conditional and are limited to the following.

1. The research focus is on unequal area rectangular fixed dimension facility layout problem where the total facility area is restricted within the layout space.

2. The areas of the rectangular facilities are known in advance.
3. The problems under consideration are taken from the previous literature and facility size varies from 8-20.
4. The research is focused on the constructive and iterative technique using an evolutionary algorithm for solving the facility layout problem.

1.4. Thesis Objectives

The aim and objective of the thesis is to:

1. To provide a literature review of the facility layout problem alongside discovering the most adequate unexplored proposal for research.
2. To provide the methodologies for solving facility layout problems.
3. To formulate and evaluate a constructive and iterative approach for solving the facility layout problem with improved performance.
4. To improve the solution of layout problem by inclusion of the local search technique with the global technique which helps in finding the global minima.

1.5. Organization of the Thesis

The thesis is organized in 6 parts which is described as follows:

CHAPTER 1:. Provides motivation and reasoning for this thesis.

CHAPTER 2: Provide a literature review of the placement problem and tools with their significance for the research.

CHAPTER 3: Provide a general study of the several solving techniques necessary for solving the UA-FLP.

CHAPTER 4: Proposes a constructive approach combined with an evolutionary algorithm with implementation on a UA-FLP problem from the literature.

CHAPTER 5: Proposes a combined local iterative search and global search evolutionary algorithm for solving UA-FLP.

CHAPTER 6: Concludes the thesis with explanation of the results and insights achieved from the research in addition with the future lines of research.

1.6. Conclusions

In this chapter, the description and introduction of the facility layout problem with its significance have been described. We presented a review of the existing research on FLPs and approaches to solve the FLPs. We also presented the study of the approaches to solve the FLPs helped in discovering the adequate proposal for research. We also indicated the importance of both constructive and iterative approach of solving the facility layout problem. Moreover, the research is expected to develop algorithms for solving the FLPs. On review of first algorithm, it improves the results obtained from the literature but has certain limitations to its approach. In the second algorithm, the limitation faced by the previous algorithm was tackled in an attempt make a generic approach.

CHAPTER 2:

LITERATURE REVIEW

2.1. Introduction

In this thesis, a wide range of disciplines has been added such as the layout design, ergonomics, operation research, computer-aided designs, expert systems, intelligent systems, production research etc. Therefore, a comprehensive review of all the literature, concepts, efforts are beyond the scope of this thesis. In general, an overview of the literature and basic concepts are provided from the facility layout problems. In particular, the unequal area facility layout problems are emphasized. Furthermore, some of the limitations of the existing layout problems has been outlined from which a promising research methodologies is proposed to overcome these limitations.

2.2. Packing Problem

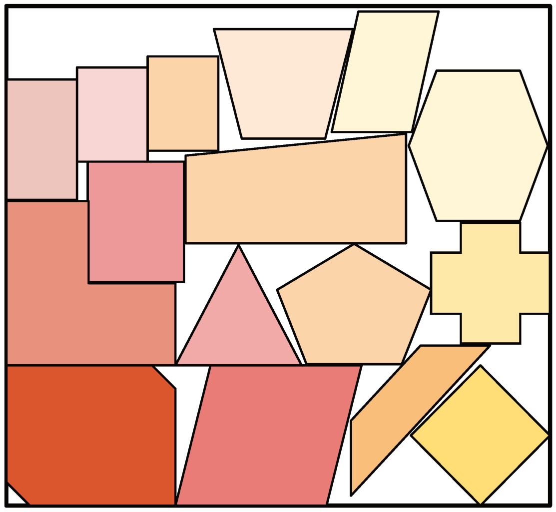
The primary objective in a packing problem is to pack all the components without overlap into a least possible packing space for maximising the total space utilization. Some of the industrial applications of packing problems are space utilization in sheet metals, paper, plastic, strip packing and textile industries. Certain applications also need additional constraints and assumptions for packing of components inside the packing space. For example, in the optimization of rectangular bin packing problems the following conditions are also satisfied.

- The edges of the components and the bin should be parallel to each other.
- The orientation of the components is fixed and cannot be rotated.
- No overlaps between the components are allowed.
- Minimum numbers of bins are to be allocated.
- All components should be placed inside the bin.

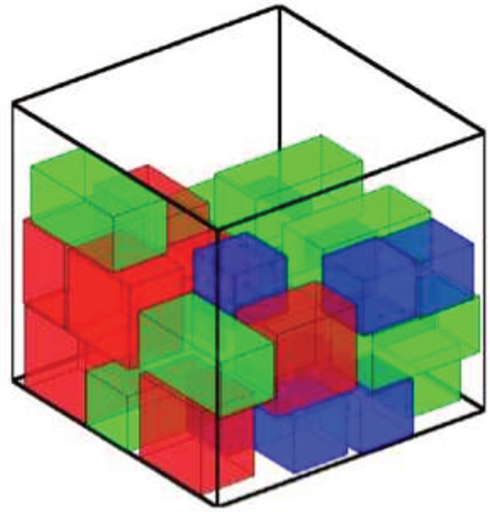
The main objective of a packing problem is the compactness. Normally, the system compactness is designed with respect to some geometrical constraints such as non-overlap and some other functions. On the other hand, several difficulties are faced in modelling specific constraints and formulation of the objective. Moreover, Different packing types such as bin packing, strip packing have dissimilar constraints and objectives.

The dimension of a packing problem also influences the difficulty in solving the optimization problem. The two-dimensional packing problems are relatively easier to optimize compared to the three-dimensional packing problems. To begin with, survey related to packing problems, consolidation of researches relevant to modelling and solution of layout problems in two and three dimensions was carried out by Dowsland et.al, 1992 [DOW92]. They have also consolidated the research carried out on exact and heuristic solution approaches. They have reported several works on packing problems such as two and three-dimensional rectangular packing, non-rectangular packing, pallet loading and strip packing. Moreover, they have recommended that there is still plenty of scope for the researcher into packing problems in spite of the extent of existing methodology.

The three-dimensional problems are more complicated than the two-dimensional problems due to additional constraints, and more variables in the objective and constraint functions. Moreover, special mathematical solvers are required to solve the intricate problem. A typical three-dimensional packing problem with cuboidal blocks is as shown in Figure 2.1(b). To optimize the 3D packing problem, Szykman and Cagan, 1997 [SZY97] developed a simulated annealing based computational algorithm. They have used spatial constraints in the optimization model that are characteristics of a 3D packing problem. These constraints are flexible enough that allows the user to restrict translation or rotation of the components with respect to the global origin or relative to other components. Similarly, an extended pattern search algorithm [YIN00] was used for efficient 3D component packing optimization. Extended pattern search allows the algorithm to find the global optimal solution in a 3D layout while bypassing the several local minimal solutions.



(a) 2D irregular packing



(b) 3D regular cuboids packing

Figure 2.1. Examples of Packing Problems.

Similarly, to solve a different three-dimensional spatial packing problem, Sachdev et al., 1998 [SAC98] described how modular A-Team based optimization is utilized. Spatial layout problem involves arrangement of components in an enclosure such that a set of objectives and constraints is satisfied. Constraints such as non-interference of objects, accessibility requirements and connection cost limits are commonly used in a 3D packing optimization. The A-Teams method synergistically combines the approaches such as traditional, genetic algorithms, simulated annealing, etc. in a modular agent based fashion.

The irregular blocks in a packing problem are illustrated with Figure 2.1(a). Two approximate algorithms were proposed by Hifi and M'Hallah, 2003 [HIF03] to solve the two-dimensional packing problem with irregular shapes of blocks. A new heuristic based constructive approach was developed for irregular shapes. They found that developed algorithm also works well with regular shaped blocks. In the second method, a hybrid approach was used that displays the layouts corresponding to the chromosomes yielded by the genetic algorithm. Author claimed that the computational time is reduced drastically when compared to the computational time found in related literatures. Albano, 1997 [ALB77] solved a problem where the main objective is to cut shapes from a given sheet of material so that, minimum wastage is attained. He proposed an algorithm that automatically generates tentative solutions and then conversational display unit to make interactive improvements.

Bin and strip packing are the two common packing problems available in the literatures. In several industrial applications, it has been seen that bin and strip packing problems have similarity in their algorithmic approaches in obtaining the optimal solution. Conversely, they do differ a lot based on application. Strip packing involves cutting a single standardized unit (a roll of cloth) into multiple strips with minimum waste or obtaining the required items by using the minimum roll length in a textile or paper industry. In warehousing (packing) contexts, the standardized stock units are commonly considered as rectangular items and the objective is to pack all the rectangular items into a large sized rectangular bins or shelves. The resulting optimization problem is called as bin packing problem [LOD02a]. In section below, we will see in detail and literatures relevant to different bin packing method and strip packing problems.

2.2.1. Bin Packing

Bin packing problem aims at packing a maximum number of items into a certain quantity of bins such that constraints value (say total weight, volumes) does not exceed some maximum value [HOP99]. There exists a large contribution in this area. To solve specific bin packing problems, we focus on some well-known algorithms such as Best-Fit algorithm and First-Fit algorithm [DÓS14]. According to Best-Fit algorithm, a new bin is opened if the item does not fit into currently opened bin. This method ensures that each bin is completely filled with Best-Fit items. Similarly, First-Fit algorithm fits each item into the currently opened bin. A new bin is opened only if the item is not fitting into the first opened bin [DÓS13; BOY12]. An example of a bin-packing problem is shown in Figure 2.2. In this case 20 unequal area rectangular blocks are packed in a rectangular bin size of 200×250 mm. The blocks in the bin completely satisfies all the conditions mentioned listed above i.e., blocks to be placed inside the bins, the edges of the blocks and bin are parallel to each other, no-overlap between the blocks, no rotation involved and orientation remains the same. Some of the objective functions in the bin-packing problem can be the minimisation of packing density or Euclidean distance between each block.

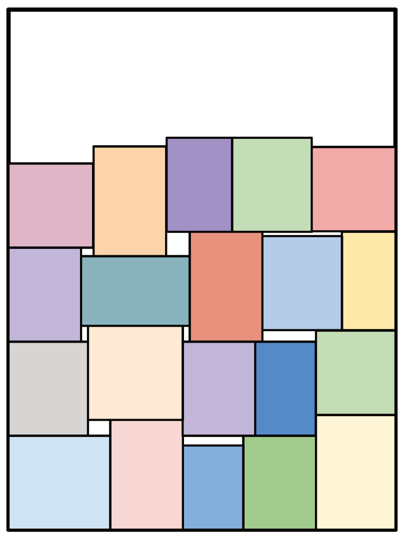


Figure 2.2. Bin Packing.

There are several types of algorithms that are developed by researchers for different kinds of two-dimensional bin packing problems. A hybrid genetic algorithm for solving 2D rectangle packing problem was introduced by [HOP99]. The first algorithm uses the heuristic technique called Bottom-Left routine, where the components are moved to the bottom and as far as possible to the left side of the bin. The major disadvantage of the Bottom-Left -routine is the creation of empty areas in the layout, when larger items block the movement of successive one. In order to overcome this drawback, the Bottom-Left algorithm has been modified as Bottom-Left-Fill placement algorithm. This algorithm allows placing each item at the lowest available position of the object. In order to achieve high quality layouts in an industrial placement problem, the Bottom-Left-Fill heuristic algorithm is recommended over a sufficient number of iterations.

A relaxation placement algorithm proposed by Jacquenot et al., 2009 [JAC09] handles the 2D multi-objective placement problem with complex geometry components. It is based on the hybridization of a genetic algorithm and a separation algorithm. Moreover, it can solve a placement problem with several types of placement constraints. They have claimed that high quality solutions will be obtained when appropriate parameters are used in genetic algorithms. They have also studied the influence of initial population and parameters of genetic algorithm on optimization results.

To solve a simple two-dimensional rectangle-packing problem, a two-level search algorithm was developed by Chen and Huang, 2007 [CHE07]. In this algorithm, the blocks are placed in a container one by one and corner-occupying action was followed to place each rectangle in a position. This action touches two items without overlapping the other already packed blocks. Initially, a simple A0 algorithm selects and packs one rectangle according to the highest degree first rule at every iteration of packing. Then in the second level, the benefit Candidate Corner-Occupying Action (CCOA) was evaluated by A0 to a more global level. Similarly, A1 packing algorithm developed by same authors produces high-density solutions within short running times.

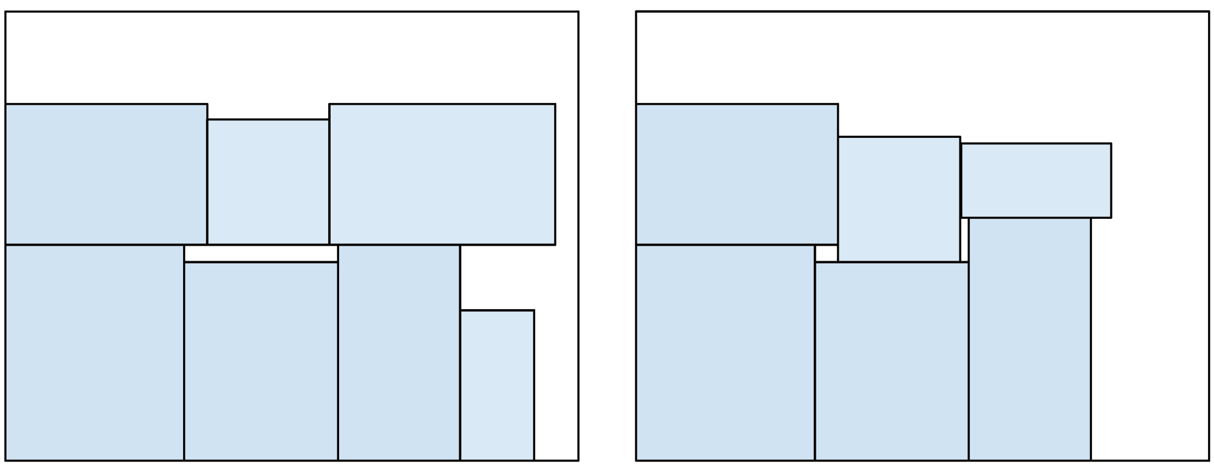
Similarly, Jain and Gea, 1998 [JAI98] presented a method based on Genetic Algorithm (GA) that can be used for solving 2D problems with convex, concave and complex shaped objects, including objects with holes. A concept named two-dimensional genetic chromosome was introduced. The total packing space was made into a finite number of cells so that it maps into the 2D genetic algorithm chromosome. In order to reduce the creation of faulty generations, the mutation and crossover operators were modified.

Three-Dimensional problem consists of allocating a given set of three-dimensional rectangular items to the three-dimensional identical finite bins (minimum number of bins) without overlapping [LOD02b]. Literatures pertaining to three-dimensional packing problem generally use the extended algorithm used in case of 2D problem. However, Martello et al., 2000 [MAR00] compared the conventional accurate and heuristic approaches to solve the three-dimensional packing problems. Industrial applications such as, e.g., container and pallet loading, material packaging design commonly faces three-dimensional packing problems. Zhu et al.,2012 [ZHU12] used finite spheres to approximate the components in a cube container and optimized the problem using finite circle method to get a solution.

2.2.2. Strip Packing

Strip packing problem is a kind of packing problem where a set of n components are packed in an open-ended bin with a fixed width and infinite height [CHU82; BER87]. In rectangular strip packing problem, the condition of packing remains same as the rectangular bin packing problem i.e. the components should not overlap with each other and a set should take a minimum packing space as possible. The only difference between the strip packing and two-dimensional bin packing is that, in strip packing the minimisation packing space is done only for one bin whereas in bin packing the minimisation is done for a finite number of bins. Strip packing problem is generally a NP-hard where all items have the same height is equivalent to the one-dimensional bin packing [GAR79; MAR03]. Some of the application of 2D Strip packing problem in real-world applications are in paper, cardboard, glass and metal industries, whenever the stock to be cut comes in large rolls which can be considered to be of infinite length.

Strip packing problem involves two different types of cutting the strip; Guillotine cut and Non- guillotine cut. Guillotine cut is basically an extraction of all items from one edge to the opposite edge in a straight line. A guillotine pattern (As seen in Figure 2.3 (a)) extracts all the items by guillotine cuts. Non-guillotine pattern is observed when items cannot be extracted by guillotine cuts as shown in Figure 2.3 (b). Non-guillotine is usually performed in a flame-cutting machine [BEK09]. The general objective of the strip-packing problem is to cut a set of rectangular pieces in a given packing space (in two or three dimensions) so as to minimize the total trim loss.



(a) Guillotine Cut (b) Non - Guillotine

Figure 2.3. Strip Packing.

Generally, strip-packing problems are handled with classical algorithms. However, Kierkosz and Luczak, 2014 [KIE14] developed a hybrid evolutionary algorithm for solving the two-dimensional non-guillotine packing problem. The algorithm uses two types of quality functions and three mutation operators. The initial solution in a tree search improvement procedure proposed by them determined by the best solution obtained by the evolutionary algorithm.

Similarly, Bekrar and Kacem, 2009 [BEK09] solved a two-dimensional strip-packing problem with the guillotine constraint. They proposed a dichotomic algorithm that uses lower bound, an upper bound, and a feasibility test algorithm. Computational results obtained show that the dichotomic algorithm, using the new bounds gives good results compared to existing methods.

2.2.3. Component Packing

Component packing problems are a common problem in many engineering applications such as layout of automobile engine compartments and design of mechanical or electromechanical assemblies. Component packing problems are also known as component layout problems as there are certain relationship or constraints within each component but it is categorized as packing problem as the general objective is to achieve high degree of compactness. Component packing tasks are characterized by three problem-independent objectives: achieving high packing density (due to trends in product miniaturization), fitting components into a specified container, satisfying spatial constraints on component placements. Packing Optimization of components is a key and common problem in several engineering applications such as layout design of an automobile engine parts assembly [MIA08], space shuttle cargo bay, and pallet loading and ship container packing [SZY95]. Component layout problem deals with three major objectives:

- Achieving product miniaturization through high packing density

- Optimal placement of components in a structured container
- Satisfying the constraints on placement of components

Miao et al., 2008 [MIA08], has presented a strategy to optimize several components in a military truck like three axles, engine, transmission system, tank, carburettor, drums etc. using Non-Dominated Sorting Genetic Algorithm-II (NSGA-II). The problem has three main objectives, which relate to maintainability, survivability, and rollover propensity. The knowledge associated with the problem allowed them to construct a parameterized model, on which a multi-objective optimization was performed. The intersections between components are evaluated using the CAO ACIS 1 software. The positioning of the components is taken as real variables whereas the orientation as discrete. The simulated binary crossover operator and a polynomial mutation operator were used in the genetic algorithm. A two-bit binary encoding is used to represent the four 90 degree rotations of a component around an axis. As noted by Grignon and Fadel, 2004 [GRI04], the use of relative benchmarks reduces the search space and limits the number of possible intersections between components.

He et al., 2012 optimized a thermal engine layout to improve the compactness in a compact layout [HE12]. Moreover, compact system optimization resulted in optimal position of the external bladder is at the tail while the optimal of the front bellows is at the nose. The Optimum layout decreased the moving distance of the mechanical moving object by 24%. Moreover, they have found that buoyant centre is placed at an optimal position near accumulator and the middle bellows. It has been reported that the overall moving distance is reduced to 43%.

In 1998, Cagan *et al*., 1998 [CAG98] have presented and illustrated an approach to 3D component placement problem through various test cases and applications. For optimal layouts, they have used the simulated annealing search to efficiently approximate intersections of complex geometric shapes. Models of components were arranged in a hierarchical fashion. The optimization algorithm developed has the capability to optimize any irregular shaped geometry inside an irregular shaped packing space with multiple objectives and constraints. The researchers claimed that algorithm was successful in variety of practical industrial layout problems. Faster solving capability of layout problems with better up front prediction of performance, layout costs and production feasibility are the implication of this technique.

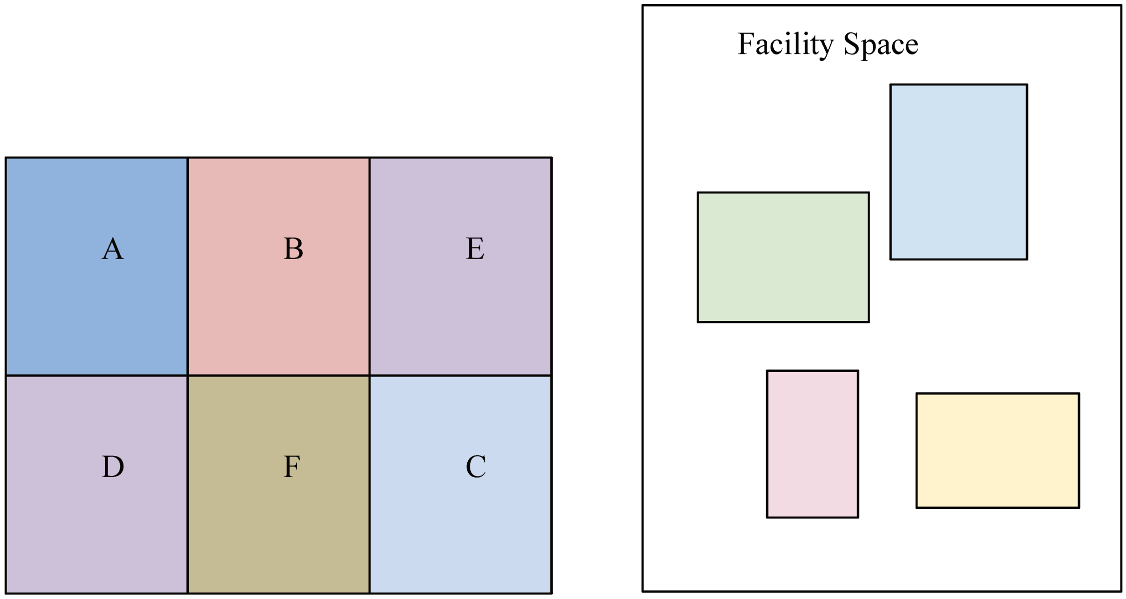
A simulated annealing based optimization approach proposed by Szykman and Cagan, 1995 was generally used for solving the three-dimensional packing problem [SZY95]. However, they have further extended their simulated annealing based algorithm for solving three-dimensional layout problems as well [SZY97]. The new algorithm developed by them deals with all three objectives required as discussed above for the optimization of component layout problem. The inclusion of spatial constraints for the components allows the designer to carryout various activities like setting desired component proximity, restrict translation motion, rotation of components with respect to global origin and so on.

2.3. Layout Problem

Layout problems are generally found in different of manufacturing systems and have a great impact on the performance of the system [DRI07]. Location of facilities i.e. machines and departments are the typical known layout problems in an Industry. Layout problems are complex and known to be generally Non-Deterministic Polynomial-time (NP) hard. In a typical 3D layout of a workspace, a rectilinear shape, but not necessarily a convex polygon represents each cabinet. As the cabinet heights are same, the 3D problem can be simplified and conceptualized to a two-dimensional layout problem. Characteristics of a layout problem are well defined by the compactness, dimensions, objectives and associated constraints. This subsection will consolidate famous literatures on different characteristics of a layout problem.

The departments or workstations are considered to be of rectangular shapes by many researchers, though others have assumed that the departments or workstations have irregular-shaped areas [BUK14]. They provided a new MILP approach for the facility process layout design. Since desired data is not available, this problem is limited to distance and accessible services to consumers. The distance between a new facility and existing facility is modelled in Euclidean distance.

The difficulty of solving the layout problem increases with the degree of dimension and shape of the components. An example of two-dimensional layout with identical rectangular blocks is as shown in Figure 2.4(a). Generally, the positions for the facilities are fixed in this case. Using Quadratic Assignment Problems can solve the desired allocation of the facilities in a location. Similarly, an example of unequal area layout problems can be seen in Figure 2.4(b), which are more complicated in solving that the equal area layout. Usually in this case the distance between the blocks should be minimised along with no overlap condition and compact placement. A common objective function for both the case may be the material handling cost between the blocks.



(a) Identical Shape - Equal Area Layout (b) Non-identical Shape - Unequal Area Layout

Figure 2.4. 2D Layout Problems.

2.3.1. Architectural Layout

Architectural layout is a kind of layout problem where aesthetic and accessibility qualities are given importance along with the engineering objectives like architectural design, cost and performance [DRI07]. Every facility of an architectural layout is resizable as they do not have any predefined dimensions. To solve this kind of problems Michalek and Papalambros, 2002 [MIC02] used an interactive method for the conceptual design of architectural layout which was possible by integrating human decision making with mathematical optimization. In this method, the designer interacts with the object-oriented representation of the physical relevant facilities during optimization. The designer's interaction with the program involves addition, deletion and modification of the object-oriented representation along with the objectives, constraints and the structural units. This way the designer has appropriate control over the subjective and qualitative judgements to achieve creative exploration.

2.3.2. Facility Layout

The facility layout problems are concerned with the arrangement of a predetermined number of departments or activities and find an optimum relative location of facilities on a planar site. A good facility layout increases the performance of the job. A facility can be a machine tool, a work centre, a manufacturing cell, a machine shop, a department, a warehouse, etc. [HER97]. The Layout that doesn't change upon the time is called a Static Facility Layout Problem (SFLP). SFLPs were introduced by Koopmans and Beckmann, 1957 [KOO57] for the first time where the shapes and areas of all facilities or departments are same. Armour and Buffa, 1963 [ARM63] further developed the problems of STLPs later where they stated that dimension of the departments or facilities can be different. They also assumed that during the iteration of algorithm the shapes of facilities or department could also change. Imam and Mir, 1989 [IMA89] further developed this problem and stated that the dimension can be different for different department or facilities but fixed the dimensions during iterations of the algorithm. Neghabi et al., 2014 [NEG14] proposed a new model for robust multiple row facility layout problem called RABSMODEL to capture the uncertainty in size of the facilities. According to the authors, it is imperative that a robust layout can be defined by different approaches. In the model proposed by them, a robust layout is defined as the layout that allows the decision maker to change the dimensions of departments within a pre-established range. Robust layouts save the rearrangement cost and avoid the re-layout of facilities. Robust layout minimizes the expected demand over the planning horizon. They generated a set of problems to test the proposed mathematical model. Saraswat et.al, 2015 [SAR15] presented a new framework that exploit the recent advances in the facility layout literature. In the multi-objective framework three objectives were taken into consideration i.e. the flow distance, average work in progress and the number of material handling devices. They conducted a number of experiments to study the trade-offs between different objectives in order to indicate that it is critical to pursue multiobjective analysis.

Facility layout problem is divided into three main sections [TOM10] and they are:

- Layout design
- Facility system design
- Material handling system design

Layout design and facility system design concentrate more on the architecture and structure of the layout to reduce the amount of transportation of materials or products. However, the material handling system design primarily deals with minimizing of cost of material handling. Several literatures describing the characteristics of facility layout problems are available. Researchers have used different solution approaches to counter the layout problem.

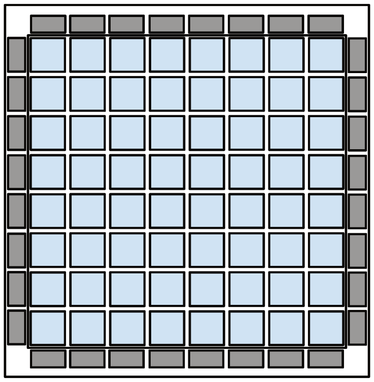
A new technique introduced by Sangchooli and Akbari, 2013 accrues an initial placement of facilities on an extended plane [SAN13]. Authors claimed that a good initial solution has a significant impact on final solution. The initial solution is obtained through graph theoretic facility layout approaches and graph drawing algorithms. Then, the initial solution is applied to the rectangular facility layout and improved further using analytical methods. This methodology is tested on several facility layout problems and it was found as an effective technique to solve real time industrial problems. Similarly, Azadivar and Wang, 2000 developed a technique to encounter the facility layout problem that considers the dynamic characteristics and operational constraints of the system [AZA00]. Based on a system's performance measures, such as the cycle time and productivity, the facility layout problem is solved. Researchers demonstrated that the test result with proposed approach could overcome the limitations of traditional layout optimization methods. Moreover, it is also capable of obtaining an optimal or near optimal solutions.

Generally, classical approaches mainly focus on minimization of material handling cost in facility layout problems. However, in real problems in the industry, the designer faces many multiple conflicting objectives in order to design the facility. There are some works in literature, which deal with multi-objective facility layout problems. In 2005, a Genetic Algorithm (GA) based approach by Lee *et al.*, 2005 [LEE05] was used in a multiple objective multi-floor design layout. Their objectives were to minimize departmental material handling cost and also to maximize closeness rating. They used weighted sum method to solve the problem. Most of the researchers proposed integrated approaches for determination of block layout and locations of Input/output points in a facility layout problem. Arapoglu *et al.*, 2001 developed an algorithm using GA to determine block layout and I/O points in a flexible bay environment [ARA01]. Kim and Goetschalckx, 2005 presented a Simulated Annealing based algorithm, wherein, a mixed integer programming (MIP) formulation is used to determine the optimal layout [KIM05]. However, Jaafari *et al.*, 2009 found that facility layout problem has multi objective functions: minimizing departmental material handling cost and maximizing closeness rating [JAA09]. They determine location of Input/output (I/O) point with multi-objective approach.

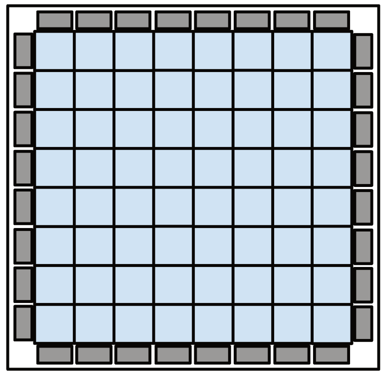
2.3.3. VLSI Circuit Layout

The problem of large scale integration of circuits, also abbreviated as VLSI (Very Large Scale Integration) is the subject of numerous studies. The growth of challenging applications as VLSI led to the efforts in automating the Circuit Layout Design. The layout configuration of circuits contains around hundreds of millions of components with strong interactions between them, due to which designing this problem is a very hard problem. Also, the designing process of a VLSI circuit can be broken down into steps such as macrocells, connectivity, placements etc. Usually, the VLSI circuit layout consists of macrocells that contains a group of circuit elements that are interconnected with a connectivity or functionality criteria [MAZ99]. These macrocells are later defined as blocks for developing a layout where the locations of the macrocells are specified. These blocks layout deigning problems are very similar to the bin packing problems with the objective to minimise one or more functions. Similarly, the lengths of connection between all the components are to be minimised to achieve minimum communication time between them. Some of the other objectives may include the minimization of the heat dissipated among the components and maximization of first vibration frequency of the electronic card. Therefore, this problem can also have viewed as multi-objective problems. In automating VLSI circuit layout designs also leads to the well-optimized VLSI layout resulting in shorter development cycle time with improvement of various performance parameters [YOU03].

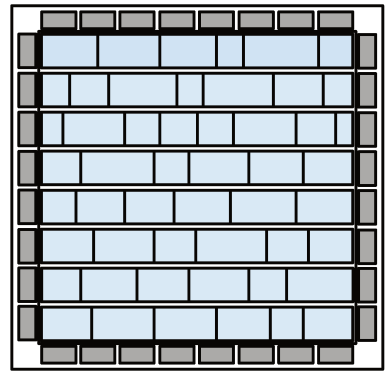
Depending on the geometry of the components, there are several types of VLSI problems. Figure below shows the different types of problems. Cases (a) to (d), the modules are of identical size. The problem then becomes an assignment problem that is purely combinatorial in nature and the geometry is no longer involved. Some of the single objective problems [KUH55, MUN57] in this category are solved by the Hungarian algorithm. Deb et al., 2004 [DEB04] developed a multi-objective problem based on a genetic algorithm to solve this kind of problem. For case (c) and (d), a sequential placement method for positioning the modules one after the other is used. They are modelled combinational form and solved using heuristics. The general



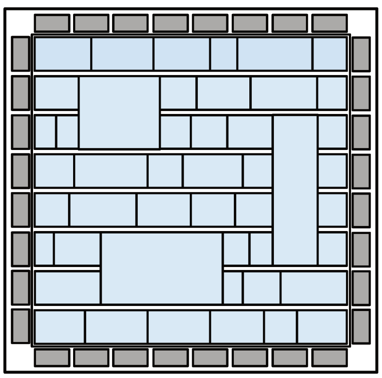
(a) Gate array.



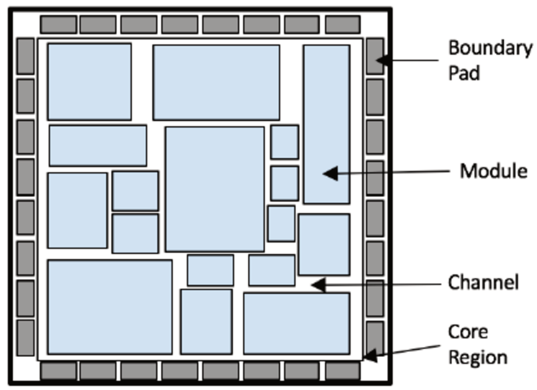
(b) Sea-of-gates.



(c) Standard-cells.

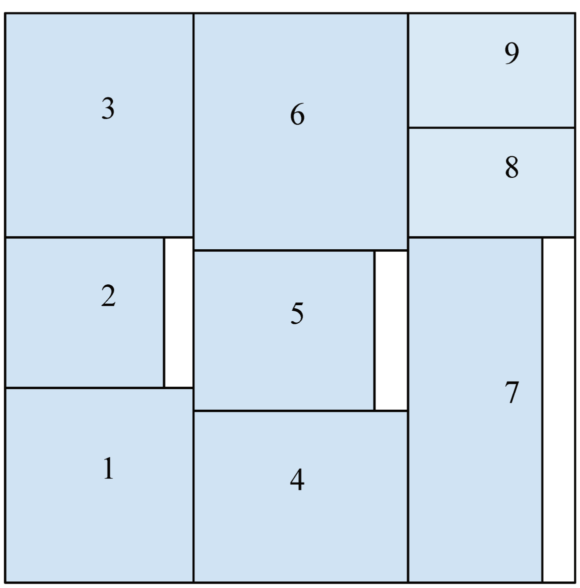


(d) Mixed cell layout.

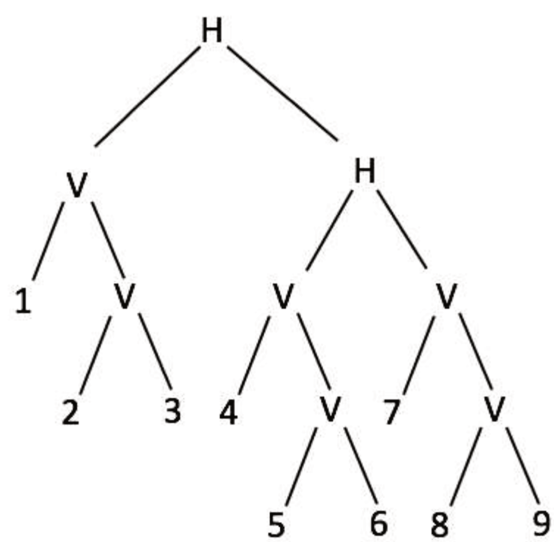


(e) General-cell layout

Figure 2.5. Examples of different problems of placement of modules in VLSI [EGE03].



(a) Layout of Rectangles



(b) Tree Representation

Figure 2.6. Arrangement of 9 rectangles with IPN notation 123VV456VV789VVHH.

case, where the modules are different in size, is shown in subfigure (e). This case can be treated in a combinatorial manner or with a formulation in real variables. The modelling of these problems is based on minimizing the length of connections with a goal of achieving compactness. Similarly, when the assembly of the components are made compact, the sums of the lengths of connections become smaller. From this, different encoding schemes have been developed. The information of the position of component is then represented in a coding system, used to define the coordinates of each component. The coding system can be a permutation, a set of permutations or a Polish Notation [ONO99]. The most popular encoding schemes are Reverse Polish Notation (IPN), Ordered Tree [GUO99] (O-Tree) and Sequence Pair [PIS07] (SP). The simulated annealing algorithm [KIR83] is mainly used to optimize the layout of the circuits [EGE03]. An example of arranging a set of rectangles encoded using a IPN is shown in Figure 2.6.

2.4. Mathematical Representation

Different types of representation for the layout problems are proposed in the literature. The layout problem was first introduced by Koopmans and Beckmann, 1957 [KOO57] and was represented as Quadratic Assignment Problem (QAP). Various formulation of the facility layout problem can have divided into two main categories: discrete representation and the continuous representation. In the discrete representation, the facility and the departments are usually represented in a grid structure. The dimension of the facility is usually fixed and the departments are composed of integer number of grids. By this way, the FLP is simplified using the discrete representation with the consequence of eliminating most of the solutions from being considered. In the continuous representation, the department are represented in a continuous manner, unlike that of the discrete representation where the dimensions are restricted to grid structure. Continuous representation has the capability to find the "real optimal" best layout solution and is more accurate and realistic than the discrete representation. Though in this representation the complexity of the FLP increases [LIU07]. The first known exact approaches are the QAP and MIP models. Other different types of layout representation include the Quadratic Set-Covering problem [BAZ75], Two-Dimensional Bin-Packing Problem [HER91].

Neghabi et al., 2015 [NEG15] proposed a new mathematical model for multi-floor layout with unequal department area. The proposed model can be helpful for optimal arrangement of departments in multi-floor process plants where the existence of adjacencies between departments is useful or essential due to possible establishment of transferring pipes. Maximizing the number of useful adjacencies among departments is considered as the objective function. The adjacencies are divided into two major categories: horizontal and vertical adjacencies. The horizontal adjacency may be occurred between the departments assigned to same floors while the vertical can be happened between departments assigned to any consecutive floors. A minimum common boundary length (surface area) between any two horizontal (vertical) adjacent departments is specified. The objective function of the optimization problem is set as maximizing the number of useful adjacencies among departments. The efficiency of the model is evaluated and demonstrated by six illustrative examples. The results of the computational experience reveal the efficiency of the proposed model.

Tari and Neghabi, 2015 [TAR15] further developed a new version of adjacency for layout problem with more flexible design. The new adjacency rating can also be considered as the generalized version of the traditional adjacency. The new version of adjacency is mainly a new continuous variable, called adjacency degree, which is used to measures the adjacency degree between any two departments. In this adjacency rating system, the departments that are not adjacent but close to each other within a pre-specified are also considered adjacent with a smaller adjacency value. The adjacency degree is inversely correlated to Tchebychev distance between two departments and takes the value between intervals of [0,1].

2.4.1. Quadratic Assignment Problem

Layout Problems are diverse in nature, due to which they are defined differently in the literature. The Quadratic Assignment Problem (QAP) was first proposed by Koopmans and Beckmann, 1957 [KOO57] and the model is a special case of FLP as the departments are assumed to be of equal areas and the locations are fixed as in a grid and known a priori. In QAP every department are assigned to one location and the at most one department to each location. The cost of locating a department is dependent with the other interacting departments. Although, the QAP is the simplified version of a facility layout problem but in reality, it is not applicable for industrial application [DRE04]. Koopmans and Beckmann, 1957 [KOO57] defined the layout problem as an industrial problem, where the objective was to find the location of plants in a set of fixed locations in order to minimise the material handling cost associated with them. They formulated this allocation problem as QAP. The QAP is considered to be a special case of facility layout problem as it assumes the plant has equal shape and area with known possible location of plants. In QAP formulation the plants or facilities are assigned to a location and at least one facility to a particular location. In a facility layout problem, the cost of placing the facility at a particular location is dependent on the placement of the other facility interacting with it.

In a variation of QAP, the distances between the departments are assigned for inter-module interaction. However, the general principle remains the same, as a number of departments has to be assigned to a number of locations and at least one department should have at most one location. The total cost can be formulated as:

$$Cost(A) = \sum_{i} F\big(i, S(i)\big) + \sum_{i}\sum_{j}\big[C(i,j)\ \delta\big(S(i), S(j)\big)\big] \qquad 2.1$$

where, C*(i, j)* represents the cost between the pair of department *(i, j)* for inter-module interaction. $\delta(i,j)$ represents the cost because of spatial separation of the location pairs *(k, l)*. *F(i, k)* is the possible fixed cost if present for placement of department i in location *k*. *S(i, j)* represents the department *i* assigned in a mapping A of activities to sites.

The QAP for an unequal area facility layout problem are quite uncommon than the equal area facility layout problem. Gilmore, 1962 [GIL62] and Lawler, 1963 [LAW63] had initially proposed a variation of branch-and-bound method for solving the QAP while a pairwise exchange method was used by Armour and Buffa, 1963 [ARM63] to solve the unequal-area FLP. In 1992, other researchers like Bazaraa, 1975 [BAZ75]; Hassan, Hogg & Smith, 1986 [HAS86]; Kusiak & Heragu, 1987 [KUS87] formulated the unequal area problem as QAP by decomposing the facilities into small squares of equal areas and assigning a large artificial flow cost among the squares of the same department to ensure that they are close to each other. Bozer & Meller, 1997 [BOZ97] proved that the artificial flows (cost of travel within the departments) dominates the real flows (cost of travel between the departments) as the artificial flows are set much larger than the real flows which affects negatively the obtained solutions. As a case in point, regardless of the real flows assigned by the analyst, each department is predisposed to assume a certain shape.

In other words, high artificial flows implicitly add department shape constraints. As a case in point, regardless of the artificial flows that were supplied by the analyst, each facility is inch to form a particular shape. This is to say that the high artificial flow cost substantially adds a shape constraint to the facility [BOZ97]. The discrete representations of the floor area also lead to the irregular shape facilities in the layout that are not feasible in practice.

2.4.2. Mixed Integer Problem

The first mixed integer programming formulation for solving the facility layout problem on a continuous plane was proposed by Montreuil, 1991 [MON91]. Using this method Montreuil was able to solve a problem with a maximum of six facilities. The disjunctive constraints were used in the model to prevent the facility to overlap with each other and bounded perimeter constraints were used keep in check the facility areas and the shape. The author used a distance based objective function that was similar to QAP. The objective was to minimise the material handling cost by decreasing the weighted distance between the facilities.

In the QAP the equal sized departments of discrete number are assigned to group of discrete locations whereas in MIP the unequal size departments of discrete numbers are positioned in a continuous space. The main advantage of MIP is the flexibility it provides to the department sizes, though constraints are given to the department size and orientation of the department known a priori. As the MIP uses unequal area departments it is more realistic representation of the industrial problem compared to the equal area departments used in QAP. However, solving the MIP becomes more complex than QAP when the number unequal area departments in the problem increase.

Salmani et.al, 2015 [SAL15] developed a bi-objective MIP model was for facility layout problem (FLP) under uncertain conditions. In this model, it was assumed that the length and width of each department were not exactly determined, and both of them could change according to the deviation coefficients and also is assumed that it has dynamic and uncertain values for departments' dimensions. According to these parameters, a definition for layout in uncertain environment is presented and a mixed integer-programming (MIP) model is developed. Moreover, two new objective functions are presented and their lower and upper bounds are calculated with four different approaches. It is worth noting that one of the objective functions is used to minimize the total areas, which is an appropriate criterion to appraise layouts in uncertain conditions. Finally, there are no predetermined areas needed for layout and departments and their areas will be determined according to a mathematical model.

Xie et.al 2016 [XIE16] proposed an improved MIP formulation using inner approximation to reach the objective of unequal area facility layout problems (FLP) in order to minimize the total material handling cost.

Izadinia et.al, 2016 [IZA16] defined a special class of multi-floor layout problem called uncertain multi-floor discrete layout problem. The new model is considered to have realistic assumptions, so the uncertainties with predefined demands, department location and material handling cost were also taken in consideration. In this problem, an underground store is utilized to contain main storages of a multi-floor building and the other floor s contains different departments in predetermined locations. A MIP model was developed to generate the robust solution for the newly defined problem and a hybrid ACO algorithm was used to solve the problem.

2.4.3. Quadratic Set-Covering Problem

The facility layout problem can also be formulated as a Quadratic Set Covering problem. The data required for the formulations are the size of each department and a set of locations for each department. The possible sets of locations for each department are given by the designer that helps them to eliminate undesirable locations. The set of locations are usually formed user's intuition and expertise that helps in the reduction of computational efforts by limiting the search space. However, in this method a large number of inputs are required for each department [LIG00].

For example, if we have a layout area which may have different departments with different functionalities. Then the interaction u_{ij} can be realised between any two departments i and j. The general objective is to locate the departments by minimizing the total interaction weighted by the distance between the departments along with the fixed cost if there is any. The QSC problem is similar to QAP with some differences. In QAP the objects are usually equal area but in QSC the departments are of different areas and designs, an example of which can be shown in the figure below.

Figure 2.7. Example of departments in Quadratic Set Covering Problem.

The layout area is also divided into small blocks in order to accommodate the departments shown in Figure 2.8. The designer usually assigns the candidate locations for each department in the layout space. For example, suppose the layout space is divided into 60 blocks as shown in fig. Then the candidate positions for Object A in Figure 2.7 can be written as

1. 1, 2, 11, 12, 13, 14, 21, 22, 23, 24
2. 10, 20, 30, 40, 9, 19, 29, 39, 28, 38
3. 55, 56, 57, 58, 45, 46, 47, 48, 35, 36
4. 39, 40, 27, 28, 29, 30, 17, 18, 19, 20
5. 21, 31, 41, 51, 22, 32, 42, 52, 23, 33
6. 13, 14, 23, 24, 25, 26, 33, 34, 35, 36

Every set of blocks mentioned above represent the position of object A in the layout space. The user takes the advantage of experience from the beginning by assigning the desirable candidate locations and eliminating undesirable location which also helps in reducing the computation effort by reducing the search space.

1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60

Figure 2.8. Layout space in Quadratic Set Covering Problem.

For formulation of the QSC problem with total cost function can be written as:

Minimise

$$\sum_{l=1}^{I(j)} \sum_{k=1}^{I(i)} \sum_{j=1}^{m} \sum_{i=1}^{m} u_{ij} x_{ik} x_{jl} d(k_i, l_j) + \sum_{k=1}^{I(i)} \sum_{i=1}^{m} f_{ik} x_{ik} \qquad 2.2$$

Subject to

$$\sum_{l=1}^{I(j)} x_{ik} = 1, for\ i\ = 1, 2, \ldots, m \qquad 2.3$$

$$\sum_{l=1}^{I(j)} \sum_{l=1}^{I(j)} x_{ikt} x_{ik} \leq 1, for\ each\ block\ t \qquad 2.4$$

$$x_{ik} \epsilon \{0,1\}, for\ each\ i\ =\ 1, 2, \ldots, m,\ for\ each\ j\ =\ 1, 2, \ldots, I(i)$$

where $\alpha_{ikt}$is l if block $t\ \epsilon Ji(k)$ and 0 otherwise.

where, u_{ij} is the interaction between the department i and j. x_{ij} is 1 if department i is assigned to its location j, otherwise it is 0. $D(k_i,\ l_j)$ is the distance between the k^{th} and l^{th} position of departments i and j. $I(i)$ is the total possible location for the department i. $J_i(k)$ represents the set of blocks occupied by the department i, if it is assigned to location k. If fixed cost f_{ik} is available, then the of assigning department i to location k.

2.4.4. Sequence Pair Representation

The sequence pair method was implemented by Murata et.al, 1995 [MUR95] for the representation of VLSI Layout. In this method, the modules are coded and taken as pairs named sequences. A P-admissible solution space was introduced for the search to be effective. The coded representation also represented a set of desirable properties for the solution space resulting in faster and better search on the family of codes. The minimum requirement for the solution space to be P-admissible are: a) Solution space is finite. b) All solution is feasible. c) Code realization is done in a polynomial time. d) one of the code from solution space correspond to an optimal solution. In 1996, Murata et.al [MUR96] reformulated the problem taking a set of two permutation of length n was used to define the VLSI Layout completely. The first permutation contained the sequence of the modules and the second permutation contained the position of the modules. This representation greatly improved the solution.

2.4.5. Two-Dimensional Bin-Packing Problem

The layout problem can be formulated as a two-dimensional rectangular bin-packing problem. The traditional bin-packing problem is the problem of maximizing the number of blocks in a bin or minimization of total number of bins to pack a number of blocks. However, in other objectives is to maximize the total utility of the blocks packed in one or more bins. Two-dimensional bin packing problems were first formulated by Gilmore and Gomory,1965 [GIL65] as the extended work of one dimensional bin packing problem [GIL61, GIL63]. The bin packing problem can be formulated same as a knapsack problem [XIA10]. The linear programming formulation of the bin-packing problem can be written as:

$$min \sum_{i-1}^{n} y_i \tag{2.5}$$

subject to

$$\sum_{j-1}^{n} w_j x_{ij} \leq c y_i, i \in N = \{1, \ldots, n\}, \tag{2.6}$$

$$\sum_{i=1}^{n} x_{ij} = 1, \qquad j \in N, \tag{2.7}$$

$$y_i = 0 \; or \; 1, \quad i \in N, \tag{2.8}$$

$$x_{ij} = 0 \; or \; 1, \quad i \in N, j \in N, \tag{2.9}$$

where

$$y_i = 1 \; if \; bin \; i \; is \; used, 0 \; otherwise,$$

$$x_{ij} = 1 \; if \; item \; j \; is \; assigned \; to \; bin \; i, 0 \; otherwise,$$

The weights w_i are assumed to be positive integers. That being so, without loss of generality, it is also assumed that c is a positive integer and $w_j \leq c \ for \ j \in N$.

2.4.6. Graph-Theoretic Formulations

The concept of graph theoretic in layout design was introduced by Seppannen and Moore, 1970 [SEP70]. Graph-theoretical approaches assume that the preferences for locating any pair of departments to be placed adjacently are known [KUS87]. In graph-theoretic approach identifying the maximal planar sub graphs shows the relationships between the facilities. The layout in this approach is constructed as the dual of a planar graph ignoring the area and shape where nodes represents the departments and the links represents the adjacencies between the departments. However, the construction of the layout from the planar graph is possible when adjacency requirements between the departments and departments & boundary areas are met. In this method, the layout may or may not achieve the shape and size assigned to each department [LIG00]. They also added that this approach doesn't guarantee that the department having strong relationship are adjacent to each other and it may also produce irregular shape departments. The layout efficiency generally depends upon the material handling cost. The material handling cost for two adjacent departments can be written as:

$$Max \sum_i \sum_j (r_{ij}) x_{ij} \qquad 2.10$$

$x_{ij} = 1 \ if \ departments \ i \ and \ j \ are \ adjacent, 0 \ otherwise,$

where, $r_{ij}$is a numerical value for the closeness rating between department i and j.

The objective helps in translate to constructing a graph with department pairs (nodes) having maximum weight on the adjacencies (arcs). The steps for forming a layout in a graph theoretic approach requires: (1) development of an adjacency graph from the inter-module interaction of adjacent departments, (2) construction of dual graph of the adjacency graph which represents the adjacent departments as regions having specific boundaries, and (3) conversion of dual graph to block layout with specific shape and area of the departments. In the second step the combinatorial nature of number of adjacencies makes the problem difficult to solve. The objective of the graph theoretic approach is maximised when the arc between the department pairs have a positive flow between them [HAS87].

2.5. Conclusions

This chapter provides a basic knowledge about FLPs has been revised that is essential for the research. For this purpose, the previous research published in this area has been analyzed taking into account the characteristics and the resolution approaches considered by the researchers for solving the FLPs. From the literature review it can be concluded that it's still open and active area for research. This motivates the author to work in the FLPs research. In the next chapter various techniques for solving the UA-FLPs has been described and the remaining thesis is focused on UA-FLP research.

CHAPTER 3:

SOLVING LAYOUT PROBLEMS

3.1. Introduction

Various methods have been used for solving the facility layout problem. Though the approaches for solving the problem can be divided into two main class i.e. exact approach and heuristic approach. A few examples of different approaches have been discussed in the section follows. But, first we aim to recall the fundamentals of design optimization.

3.2. Design Optimisation

Design optimization can be taken in as the methods and tools that allow the designer to improve a product or a system. These methods help in achieving the best possible result of the performance criteria with all the available resources. The solution in general refers to the combination of the design variables and the set of parameters useful for improvement of the product. It can be characterized based on the design constraints and performance criteria related to the product specifications.

Design optimization is currently an area of research that is the subject of many studies. Indeed, more and more industrialists are beginning to implement a process of optimization in their company because they are continually seeking to improve the "cost, quality, time". This industrial need can be explained by various reasons, including:

- The strong global competitiveness between companies,
- The rapid evolution of technologies and production systems,
- Improved interactions with the client.

Thus, for the designer, the advantages associated with the use of an optimization method are various:

- Finding new solutions that meet the product specifications,
- Look for solutions that achieve the best compromise in terms of performance and design requirements,
- Justify its technological choices by quantitative data to the decision-maker on the performance and constraints related to the problem.

In brief, an optimization method consists firstly, of writing and formulating the problem by converting all the best possible requirements of the designer. Next, the approach suggests designer the use of an appropriate optimization strategy. Finally, the final step consists of taking a decision, in terms of design choices, with respect to the different optimal alternatives proposed by the algorithm.

3.3. Formulation of an optimization problem

The mathematical formulation of an optimization problem can be written in non-linear programming format:

$$\left.\begin{array}{lll} & Minimise\ f(x) & \\ \text{subject to} & g_j(x) \leq 0 & j = 1,2,\dots,J; \\ & h_k(x) = 0 & k = 1,2\dots,K; \\ & x_i^{(l)} \leq x_i \leq x_i^{(U)} & n = 1,2\dots,N; \end{array}\right\} \quad 3.1$$

The equation 3.1 contains the following elements:

- $x = (x_1, x_2, \dots, x_N)^T$ is the design variable as the column vector.
- $f\ (x)$ is the objective function of the optimization problem. Here, the design variables x are mapped to real values through the objective function representing the desirability of this solution to the decision-maker. Generally, the objective functions represent cost that has to be maximized or minimized.
- J is the total inequality constraints for $g_j(x) \leq 0$.
- K is the total equality constraints for $h_k(x) = 0$.

The constraints have zero-valued vectors on the right-hand side to properly match the vectors on the left-hand side. Both equality and inequality defines the feasibility region of the optimization problem. Equation 3.1 determines, the solution to be feasible and provide a lowest value of the objective function. It is also worth mentioning that the four tasks in the equation are not independent of each other. During the formulation of the problem the designer may decide to add or delete any constraints.

In many cases an additional design variables are included while formulating the constraints to make the overall formulation easier. The design variables, constraints, objective function and the variable bounds are updated till the designer is satisfied with an acceptable formulation. The knowledge of the optimization algorithm also helps in this update to solve the problem. The practice of implementation of optimization algorithm is necessary for any modification of the formulation procedure. However, after the formulation the optimization algorithms is taken as the optimal solution and an optimal solution of the NLP is obtained.

3.4. Evolutionary Algorithm

Evolutionary algorithms are the algorithms with the underlying ideas where the techniques with a population of individuals which under the environmental conditions cause natural selection leading to the increase in fitness of the population. For minimization of a function, randomly generated candidate solutions are created; the candidate solutions are an element of the function's domain applied in the function to get a measure of fitness, the lower the better. According to the fitness value the best candidates are selected by applying recombination and/or mutation to seed for the next generation. Recombination is an operator applied to two or more selected candidate solution that generates one or more new candidate solution. Mutation is applied to a selected candidate giving a new candidate. Both recombination and mutation gives new candidate solutions and based on the fitness value they are used for competing in the next generation. The algorithm reiterates till a better quality solution is achieved or the computational time is reached.

Evolutionary algorithms cover a wide range of families of algorithms, among which we distinguish:

- Evolution strategies (ES). Based on selection and mutation operators, these are the first evolutionary algorithms. Since their emergence in 1965, several variants have been developed, among which (1 + 1) - ES, ($\mu + \lambda$) - ES, [REC65] as well as PAES [KNO00], a multi-objective version of evolution strategies;
- Genetic algorithms (GA). The GAs are the best known of the evolutionary algorithms, they are presented in detail hereafter [DEB01];
- Differential evolution (DE) is another population metaheuristics that emerged in 1997 based on the concept of vector mutation [PRI06];
- Memetic algorithms (MA). These are hybrid evolutionary algorithms using a local search method at the end of optimization, to reach the global optimum.

Many other families of evolutionary algorithms exist, but these go far beyond the goal of our research. The advantages of evolutionary algorithms are numerous:

1. Their implementation is generally simple,
2. They are robust. They are not as sensitive as the deterministic optimization methods,
3. They allow to integrate different types of variables during the optimization,
4. The calculations can be easily parallelizable, unlike most other methods,
5. 5. They allow dealing with multi-objective problems.

For all these reasons, evolutionary algorithms arouse great interest in the scientific community.

3.4.1. Genetic Algorithm

Genetic algorithm (GA) is originated from the work of Holland [GLO89A; HOL92]. GA is a part of population metaheuristics i.e. an evolutionary algorithm that mimics the natural evolutionary process into a computer system. As GA is based on the principles of survival and reproduction described by Charles Darwin [DAR59], GAs seek to improve the current population. They generally break down into two stages: an exploratory or research phase and an intensification phase.

Like all evolutionary algorithms, genetic algorithms work on a set of solutions, called individuals. All these individuals form a population and the objective of the GAs is to change the individuals over generations towards one or more optimal global [HOL75]. GAs are a very robust optimization algorithm as they can handle any type of variables or mixed type (real, integer, Boolean) and are particularly suited to the problems where the initialization doesn't have to be intuitive. As the GA is based on the principals of survival and reproduction, several biological terms are used to illustrate the functionality of Genetic Algorithms.

There are different types of genetic algorithm, for example single-objective GA, multi-objective GA, steady-state GA, multimodal GA, parallel GA. Unlike generational algorithms, continuous generation algorithms allow newly calculated solutions to be used to generate new solutions.

3.4.1.1. Principles of genetic algorithms

The principles of GA can be represented in different stages as shown in the Figure 3.1. The different stages of generational GA are population initialization, selection of individuals for the generation of the new population, and genetic crossover and mutation operations. The algorithm stops as soon as the termination criterion is satisfied such as for example a maximum number of generations, a detection of convergence of the problem.

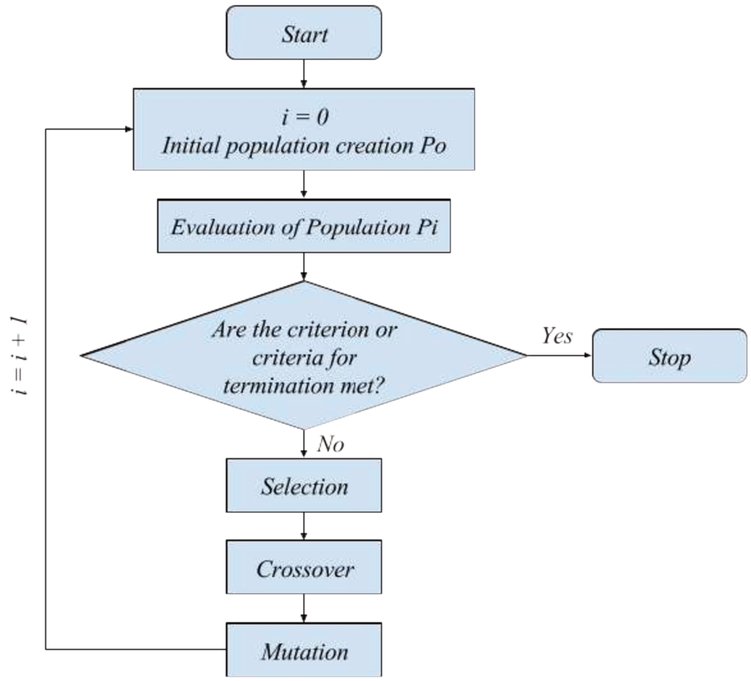


Figure 3.1. Working principles of genetic algorithm.

3.4.1.2. Genetic operators

Genetic operators are the set of operations performed on a population or individuals to sort or generate new individuals. These operators are important, because they are the ones who change individuals from one generation to another. The performance of genetic algorithms depends greatly on the choice of these operators and their settings. The genetic operators usually depend on the coding of the optimization variables.

3.4.1.2.1. Selection operators

The objective of the selection operator is to identify and select the best individuals and to eliminate others while maintaining the size of the population. Several methods have been developed, the best known being the proportional selection method also known as roulette selection. Another selection method namely tournament selection uses pair-by-pair comparisons to select the best individuals. This is the technique most used when optimizing constrained problems, where the set of constraint values is grouped together in a constraint violation index.

3.4.1.2.2. Crossover operators.

In crossover operation the exchange of attributes between individuals to generate a new individuals takes place. The Crossover operator is defined by particular type of variables (real, binary, permutation…). The individuals from the population are chosen at random and are mixed and chained again to build the new individuals. The individuals are usually of fixed coding, so the bit-strings from the parents are chosen at the same positions to preserve the overall length of the bit-string.

3.4.1.2.3. Mutation operators.

In mutation operation the individuals from the population are selected at random and a single bit-string are shifted. The mutation is usually applied randomly with a small probability on the population.

After undergoing these creations and altering processes by the genetic operators, the new individuals form the next generation. A generation is completed at this stage and is repeated till a satisfying solution is found or the termination criteria is achieved.

3.4.2. Discussion

The advantage of GAs is that they are robust, efficient, easy to implement and can be applied to a vast variety if optimization problem i.e. continuous, discrete, mixed, combinatorial, mono- and multi-objectives. They also allow an efficient exploration of the search space for highly complex problems and generate a useful exploration history for the final choice of the designer giving quick approximate solutions. GAs can also be very well incorporated with other local search algorithms where the combined search helps in exploiting the strengths of each of the methods.

The disadvantages of GAs are that the optimal solution cannot be ensured, due to which it comes under heuristic search methods. The convergence of GA is problem oriented. To find out the range in which the model is efficient sensitivity analysis are usually done. Good programming skill is also required for the implementation of the techniques.

3.5. Solving Layout Problems

3.5.1. Exact approaches

Exact algorithm are the algorithms developed to obtain an optimal solution in theory for a facility layout problems. The exact algorithm considers the whole solution spaces and guarantees the optimality of the final layout solution. Though the models are not much of practical value as the focus on small size problems i.e. less than 10 unequal size departments whereas in an industrial problem there can be more than 30 departments or facilities [DRE04]. If the size increases even more than it will be impossible to solve the problem because of the computational complexity of the layout problem.

Most well-known representation that uses exact algorithm are quadratic assignment problem and mixed integer programming models. The branch and bound algorithm is an exact algorithm used to solve the facility layout problem. The algorithm promises a high level of optimality for large number of departments with comparison to other exact approaches that have more difficulty to solve the problem efficiently. Some of the listed literature using branch and bound algorithm using the listed models are reviewed below.

Gilmore, 1962 [GIL62] and Lawler, 1963 [LAW63] were the first to develop a branch and bound algorithm for solving the quadratic assignment problem. In this algorithm, the optimal solution is obtained by implicitly evaluating all the possible solutions of the problem. In this algorithm, the increase in integer and variable require a large amount of memory and computational time. The facilities are allocated stage by stage in this method. The evaluated partial layout is compared with the lower bound. If the cost of partial layout is higher than the lower bound, then it is discarded and the branch is fathomed otherwise it is kept and used as a lower bound for the subsequent iteration.

Roucairol, 1987 [ROU87] suggested a parallel branch and bound algorithm for solving the quadratic assignment problem. In this method for finding the optimal solution, the searches are made concurrently. From the results it was concluded that for departments more than 12 the parallel branch and bound method requires more computational time. Bozer and Rim, 1996 [BOZ96] developed a branch and bound model to address the bidirectional circular layout problem (Bi-CLP). In this formation, the departments are arranged alongside the closed-loop aisle, and the flow between the departments are either clockwise or anticlockwise direction.

Solimanpur and Jafari, 2008 [SOL08] represented a two-dimensional facility layout problem as mixed-integer nonlinear mathematical programming model for determining the optimal solution. A branch and bound algorithm was used to obtain the optimal solution for the proposed mathematical programming model. Though it was concluded that the approach is inefficient for large-sized problems and proposed the use of meta heuristics models such as genetic algorithms, tabu search, ant colony optimization etc.

Huang and Wong, 2016 [HUA16] used a binary mixed integer-linear programming (BMILP) to solve a discretized cell optimization model of FLP. In this model, to effectively model irregularities the facilities and the site areas are represented as small unit cells. Due to which the variables are taken as binary type. A branch and bound algorithm was used to solve this model to obtain the global optimal solution.

3.5.2. Heuristic and Metaheuristics

The approximate solution strategies for solving facility layout can be mainly divided into three categories i.e. constructive initial placement strategies, iterative improvement strategies and hybrid strategies [LIG00]. In the constructive initial placement strategy, the solution is formed by locating the departments or activities one by one from the start whereas in the improvement strategy solution begins with an initial arrangement of all the departments in the layout and the solution improves incrementally through iteration. The hybrid approach is the combination of both constructive and improvement approach. In addition to the above strategies other intelligent solution approaches have also been classified and reviewed.

3.5.2.1. Constructive approaches

In this approach, the solution is built from the scratch by adding one by one in a stepwise manner the elements of layout using an n-stage decision process. Some methods use intelligent assignments at each stage by automating a set of "rules of thumb" with the thought process of the designer. A step for selection of a department or activity can be the maximum connectivity to the already placed department. The selection of a department can be done by using any thumb rule e.g., a department can be placed at an empty location and the remaining department are placed in a clockwise manner with respect to the already placed departments or some complicated criteria can be by selecting the department having minimum criteria function with respect to the already placed department [LIG00]. A handful of literature is listed and described below.

3.5.2.1.1. Bottom left algorithm

The bottom-left algorithm was first introduced by the Baker et.al, 1980 for bin packing problem where each module is pushed to the bottom position and far left of the packing space which are done one after one for every block. The blocks are usually introduced from the top right position with successive vertical and horizontal movements of the blocks, they are moved to the feasible position. The placement of the blocks is repeated until all the blocks occupy a stable position where a block cannot be move bottom or left. This technique has gained a considerable attention from researchers [JAK96; LIU99; HEA99; HOP01; DOW02]. The advantage of Bottom Left method is its simplicity and speed Dowsland et al., 2002 [DOW02]. The disadvantage remains the poor packing space utilization and the method tends to leave holes in the packing pattern. The input sequences of the blocks were examined and resulted that sorting the blocks with decreasing width resulted in a layout with nearly one-third the height of the previous optimum layouts [HOP01].

In the method by Jakobs et.al 1996 [JAK96], the pieces are introduced in the bin from the top right position and kept in the lowest position possible, then it is moved left possible position then again it is pushed towards the bottom. This method is repeated for the other pieces till all pieces reaches a stable position in the bin when no more pieces can be introduced. A set of permutations of the rectangle sequences was then taken as a population for genetic algorithm that was used for bottom-left heuristics and was evaluated based on the total height. Jakobs et.al stated that for a packing pattern of known number of blocks the block sequence couldn't be always written. He also stated that for a given fitness function for example inter modular distance, the layout configurations giving same compact height may not have the same fitness value.

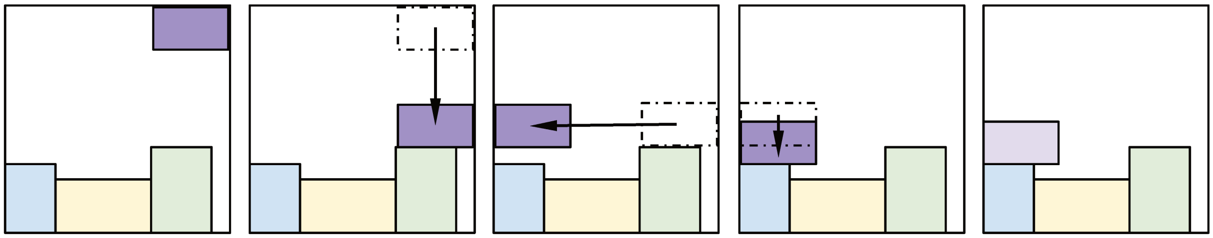


Figure 3.2. A Bottom-Left Method [JAK96].

The advantages of these approaches are its simplicity and speed [DOW02]. The downside to this algorithm is that is leaves empty spaces in between the pieces being packed in the bin leading to poor utilization of space.

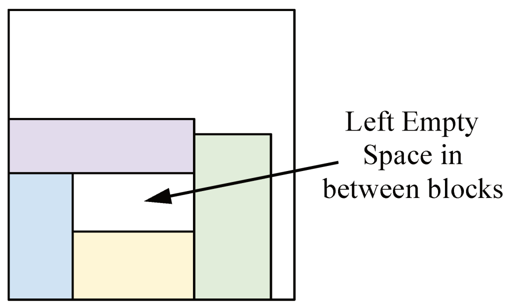


Figure 3.3. Left out empty space in between blocks in bottom-left heuristic.

3.5.2.1.2. Improved bottom-left algorithm

Liu and Teng, 1999 [LIU99] proposed an improved version of bottom-left heuristic known as Improved Bottom-Left (IBL) heuristic which is efficient and effective. The improvements done in this heuristic leads to filling up the empty spaces left between the blocks by Bottom-left heuristic and also gives a better aesthetic content. For this the strategy includes the refinement of placement decisions by allowing the blocks to move towards the bottom and adjusted with rotation. The downward motion of the blocks was given the priority unlike in the heuristic proposed by Jakobs et.al, 1996 [JAK96] and the blocks moved leftwards only if it cannot move downwards.

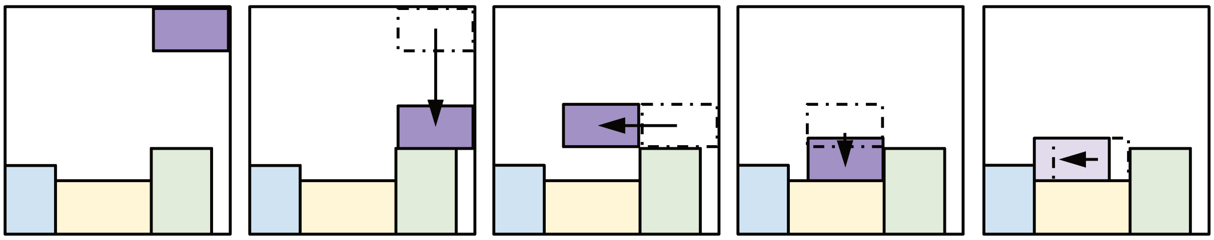


Figure 3.4. An improved Bottom-Left Method [LIU99].

3.5.2.1.3. Bottom-left fill algorithm

Bottom-Left Fill is another modified version of Bottom-Left heuristic. In this strategy, placing a block in the lowest available position and left justifying it fills the empty spaces in the packing. This leads to denser packing as it fills the existing gaps between the packing. Here the placement position of the candidate is indicated by a list of location points maintained through bottom-left ordering. In this strategy, the placement of the blocks starts with the extreme bottom point and the extreme left point then overlap and boundary conditions are checked. Then the list of candidate placement locations is updated if there is no violation when the blocks are placed. The blocks are always tested with the list of placement location and if there is overlap then it moves towards to the next placement location till there is no overlap. As a result, the bottom-left fill overcomes poor space utilization faced by bottom-left and improved bottom-left. However, the disadvantage is the time complexity, which remains without significant improvement in aesthetic content of the solution [HOP01, BUR04].

Bottom-Left Fill method is another advanced version of the bottom-left method, which helps in completely filling the empty spaces in between the blocks and results in denser packing. In this heuristic, the placement starts with positioning the first block at the lowest left position of the bin. Then placing another block either the right side of the previously placed blocks or the upper left position of the blocks. These positions are saved for the placement of remaining blocks and the filling continues for rest of the incoming blocks. During the placement of every block the overlap of the blocks with the already placed blocks are always checked. The placement is always done at the positions with no overlap condition and the best position.

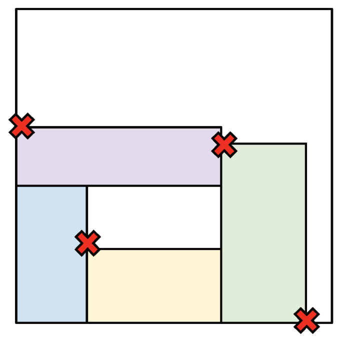


Figure 3.5. Storing placement location for one implementation of bottom-left-fill.

The Figure 3.5 shows a general layout of the bottom-left fill heuristic and also shows the available placement positions of the next block. The fifth block to be placed can be easily placed in the empty space available if it doesn't overlap with any other blocks as shown in the Figure 3.6(a) which would have been impossible for the bottom-left and improved bottom-left heuristics as shown in Figure 3.6(b). The quality of solution in Bottom-left and Improved bottom left totally depends on the sequence of the blocks placed in the bin. This way the bottom-left fill heuristics overcomes the problem of poor space utilisation faced by Bottom-Left and Improved Bottom-Left. The disadvantage is the amount of computation time needed for solving [CHA83, HOP01, BUR04].

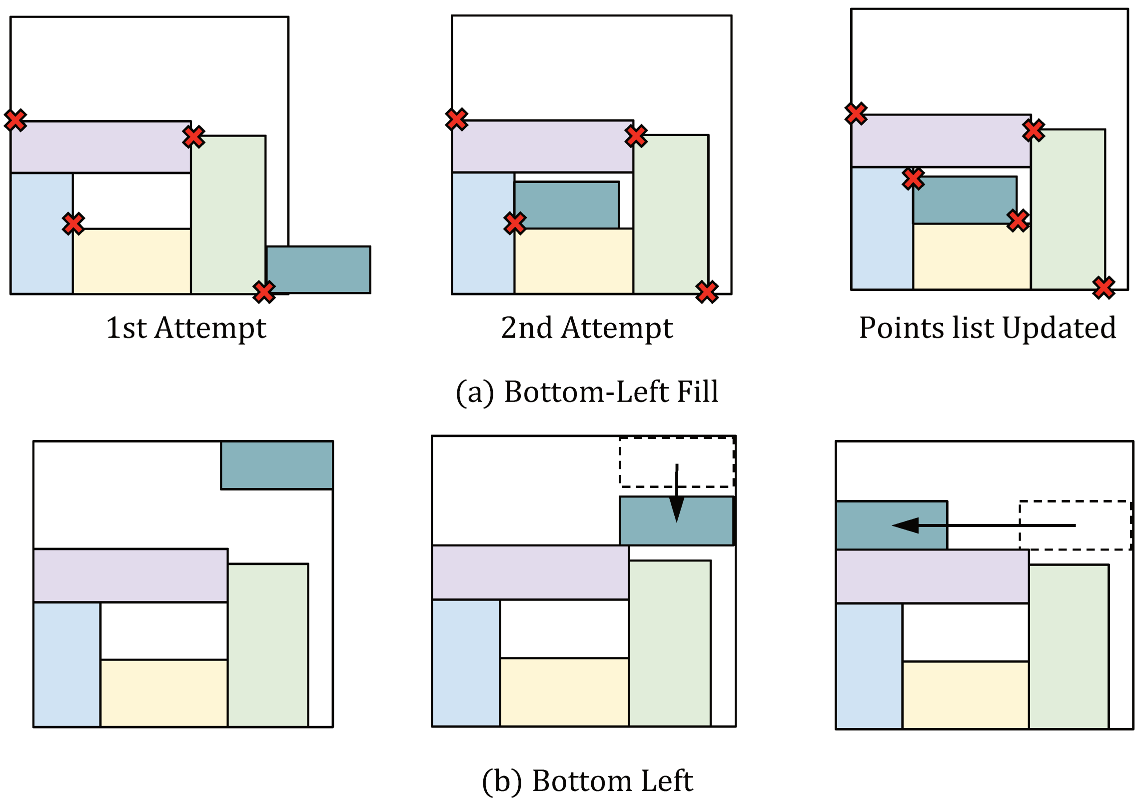


Figure 3.6. A comparison of the Bottom-Left and Bottom-left Fill placement heuristics when adding a rectangle.

In the later section a new heuristic is made from the idea taken from bottom-left fill. The bottom-left, improved bottom-left and bottom-left fill method were for implementation in bin packing problems. The new heuristic is made for facility layout problem where the first facility is kept at the centre of the layout space and the rest of the blocks are arranged or placed surrounding it. The details of the facility growth heuristic are described from next section onwards.

3.5.2.2. Improvement approaches

In Improvement approach type algorithm starts with a solution improving itself incrementally. The simplest version of improve incremental strategy is “pair exchange”. From the initial solution, it starts systematically evaluating for probable exchanges between two activities and the exchange occurs if the solution improves and meets the criteria. The variants of pair exchange strategy include improving the quality of solutions obtained and reducing the computational effort. The variants often involve the selection method of departments for possible exchange and which exchange to make, for e.g., whether to make the first exchange or not which leads to an improvement or to evaluate all possible exchanges and select the exchange that results in the maximum cost improvement. The method where the improvement is chosen over all the possibility of exchange has more computational time and is expensive [LIG00].

A lot of improvement techniques usually converge towards the local optima. As all the improvement techniques starts with an initial solution from which the local optima are generated and are compared using different starting configurations. Elshafei, 1977 [ELS77] used a technique that selects the move that retreat from the local minima and results in minimum cost increase. Then the process was repeated from the last position, which hopefully directs towards a new local optimum.

3.5.2.2.1. Simulated annealing

Simulated Annealing (SA) is an effective stochastic optimization technique well known of its high performance for solving combinatorial problems. It is also very effective in solving large and complex facility layout problems [AHM05]. The analogy of simulated annealing is motivated from the phenomenon of crystallization. The algorithm starts with a random solution, and the new solutions are obtained incrementally when the genes from one location moves to a new location. The solutions with decreased cost are accepted whereas the solution with increased cost are also accepted with a probability that decreases exponentially with time. Consequently, at the beginning many inferior solutions are accepted which decreases over time. This helps the algorithm to avoid local optima by accepting inferior solutions.

Mir and Imam, 2001 [MIR01] proposed a hybrid optimization algorithm for an unequal area facility layout problem. They used a multi-stage optimization process where Simulated Annealing was used to optimize the randomly generated initial placements and an analytical search technique steepest descent was used to determine the optimal locations of facilities. The optimization starts with initial randomly placed departments in an extended plane. For controlled convergence, the optimization was carried out using magnified envelop blocks which were gradually reduced in sizes until their dimensions become equal to those of the actual facilities.

Matai et al., 2013 [MAT13a] proposed a new heuristic approach for solving FLP where they applied a heuristic procedure to solve FLP from sets of linear assignment problem (LAP) solution. Since, FLP was formulated as LAP and the solutions of the LAP provided a lower bound on corresponding QAP formulation of FLP. Matai et al., 2013 [MAT13b] also consider a multi-objective QAP but use simulated annealing as the solution procedure for solving UA-FLPs.

Kulturel et al., 2015 [KUL15] introduced a cyclic facility layout problem (CFLP), is a special case of the dynamic facility layout problem (DFLP) in which there are several production periods and the production cycle repeats itself by going to the first period after the last one because of the seasonal nature of products. In this problem, a mixed integer programming formulation is developed for the CFLP. In the DFLP literature, department shapes are assumed to be given or fixed. However, this assumption does not hold in the case of the CFLP because the facility size is limited and the area requirements of the departments change significantly throughout the planning horizon. Therefore, department dimensions and sizes are considered as decision variables in the CFLP. Since the large-scale hybrid simulated annealing algorithm (LS-HSA) operates directly on the decision variables of FLP formulations, it does not require encoding of layouts in the computer. In this article, the LS-HSA is successfully applied to solve four different FLPs on the continuous plane: the CFLP, the DFLP with fixed department dimensions, the DFLP with variable department dimensions and the single-period FLP. In all cases, the LS-HSA was shown to be very effective, versatile and competitive with the other approaches from the literature.

Wang et.al, 2015 [WAN15] used an improved-SA to solve a problem of dynamic double row facility layout problem. A mixed coding scheme was suggested to represent a feasible solution and to express the sequence of facilities and the exact location of each facility. The problem was formulated as a mixed-integer programming model. To resolve the problem a methodology combining an improved simulated annealing with mathematical programming was used. The mathematical programming was used to determine the exact location of each facility.

Matai, 2015 [MAT15] presented a modified simulated annealing algorithm for solving multi-objective facility layout problem. Unlike the previous multi-objective algorithm, in this method the layout design process is independent of the decision maker. Also, in the proposed method any number of qualitative or quantitative objectives can be used. The weights of each objective are determined by the approach defined by Singh and Singh [SIN10] in order to converts the objectives into single objective function.

3.5.2.2.2. Ant colony optimization

Ant Colony Optimization (ACO) is a Swarm Intelligence technique based on the foraging behaviour of ants [COR99; DOR99]. Each ant taken into consideration a probabilistic choice that all the ant colony members left a trail of pheromones when it preceded it course. The pheromone trails are a smell trace left by every ant on its way, which evaporated during time due to which for each ant the probabilistic choice also changes with time. The path for the food is determined from many ant courses leaving higher pheromone trace leading to other ants follows the same path. The collective behaviour of all colony ants based on their shared memory can be used to solve combinatorial optimization problems. The analogies for ant colony optimization are:

- Ants form the solution space for the combinatorial problem.
- The quantity of food from a source forms the evaluation of the objective function.
- The pheromone trails form the adaptive shared memory.

This mechanism for solving the discrete optimization problems in various engineering domain is used by ACO.

Ant colony optimization (ACO) problems could hence be encoded as finding the shortest path in a graph. One of the first applications of ACO was the travelling salesman problem. The first ACO algorithm was proposed in the nineteenth century where it attracted the attention of increasing numbers of researchers and many successful applications. Besides, a substantial corpus of theoretical results is obtainable that provides useful guidelines to researchers and practitioners in further applications of ACO.

Chen, 2013 [CHE13] developed the work of McKendall and Shang, 2006 [MCK06] with a large number of department, n = 30, with a new data structure of DFLP solution representation where binary and hexadecimal numbers have been used to represent the solutions of DFLP which benefits to less memory usage. The proposed data structure for the DFLP facilitates the swapping and sorting activities when a meta-heuristic is applied.

Zhao et al., 2014 [ZHA14a] introduced a novel improved hybrid PSO-based GA (HPSO-GA) on the basis of parallel GA where chaos initialization and multi-subpopulation evolution are adopted based on improved adaptive crossover and mutation. Identically the characteristics of different classes of subpopulations, different modes of PSO update operator are introduced. It pursues making full use of the fast convergence property of PSO. The presented adjustable arithmetic progression rank-based selection can prevent the algorithm from premature in the early stage and benefit accelerating convergence in the later stage.

Asl and Wong, 2015 [ASL15] suggested a modified PSO to solve UA-FLPs with fixed departments shapes and areas throughout the time horizon. This algorithm implemented the department swapping method two and local search methods to prevent local optima for static and dynamic problems and to improve the quality of solutions. It also utilized the period swapping method to improve the solutions for dynamic problems.

Izadinia et.al, 2016 [IZA16] defined a special class of multi-floor layout problem called uncertain multi-floor discrete layout problem. The new model is considered to have realistic assumptions, so the uncertainties with predefined demands, department location and material handling cost were also taken in consideration. In this problem, an underground store is utilized to contain main storages of a multi-floor building and the other floor s contains different departments in predetermined locations. A MIP model was developed to generate the robust solution for the newly defined problem and a hybrid ACO algorithm was used to solve the problem.

3.5.2.2.3. Genetic algorithms

Genetic algorithm (GA) is a metaheuristic that mimics the mechanisms of the Darwinian evolution based on the concept of the survival of the fittest strategy [DEB01, GOL89]. Most important component of GA is the solution representation, also known as individual or chromosome, as it represents the complete solution of the problem. GA is also known as a population based method as it takes a set of random individuals that evolves over generations by the repeated execution of certain genetic operators similar to that of natural evolution such as selection, crossover and mutation. The selection operator helps us to keep the good individuals and eliminate the worst individuals from a temporary population known as mating pool. Sometime some worst individuals are also selected from the mating pool and are meant to take part in further evolution in order to maintain a diverse population. The crossover operator helps in finding the better solution by generating two off springs from mating of two individuals from the population with a predefined probability known as crossover probability. The mutation operator, help in generating a random individual from an individual and replacing them in the population with a mutation probability. Through these operators, the population goes through a series of generation till a termination criterion is met. In each generation, the individuals improve towards the optimum value of the fitness function. The fitness function is also a measure of quality for an individual. The termination criteria can a fixed maximum generation or a desired value of improvement attained from the objective function.

Jannat et.al, 2010 [JAN10] proposed a multi-objective genetic algorithm in order to solve both qualitative and quantitative aspect of a facility layout problem. In the qualitative approach aims at maximization of the closeness rating whereas the quantitative aspect aims at the minimization of the total material handling cost. The solutions are represented a special encoding representing the complete facility layout, which are used for genetic algorithm. Finally, a set of non-dominated solution was found for the multi-objective facility layout problem with this approach.

Ripon et.al, 2010 [RIP10] presented an evolutionary approach for solving multi-objective dynamic facility layout problem. A non-dominated sorting genetic algorithm - 2 was used to find the pareto optimal layouts. The two objectives are material handling cost and the closeness rating. Combining both the objectives solved previous research for the problem. Thus, the results obtained were compared with the previously solved, and the Pareto optimal solutions were used to find the wide range of alternative layout choices.

Datta et.al, 2011 [DAT11] proposed a permutation based genetic algorithm for solving a single row facility layout problem. The fitness function was taken to find the minimum cost for arranging a number of facilities in a single line. The solution was represented as a random order of facility in a line and the population of these individuals are improved towards the optimum from the specially designed crossover and mutation operators. As any generated solution always remains a valid solution for the single row facility layout problem, thus the GA treats this problem as an unconstrained optimization problem.

Aiello et.al, 2012 [AIE12] proposed a multi objective genetic algorithm with the slicing structure for solving unequal area facility layout problems. Four objectives were taken into consideration i.e. the material handling cost, aspect ratio, closeness rating and distance. Each solution were represented two chromosome which represented relative location of the facilities. The block layout solution was created by splitting the floor using guillotine cuts into sets of rectangular facilities. The results were compared with the self previously solved method using bay structures [AIE06].

Kulturel-Konak and Konak, 2013 [KUL13] proposed a hybrid genetic algorithm (GA) together with linear programming (LP) approach to solve the UAFLP with an attempt to further increase the computational efficiency. Where GA searches for the relative locations of the departments and the LP model determines their exact locations and shapes in which the first step is trying to decrease the number of binary variables and then solve the improved model in the second step.

García-Hernandez et.al, 2013 [GAR13] proposed an Interactive Genetic Algorithm for solving the unequal area facility layout problem. The proposed method takes the decision makers knowledge to direct the search process, where he adjusts his/her solution preference at each generation. In the problem, a large number of departments were considered with 20 generations and in order to prevent fatigue on the decision maker. The decision maker makes preference using his/her subjective evaluation of the solution representation, which are made sufficiently different and are chosen using the c-means clustering method.

Pourvaziri et al., 2014 [POU14] developed an effective novel solution approach for DFLP in which a hybrid multi-population genetic algorithm (HMPGA) with an effective structure which is used to generate initial populations. It was found that the proposed approach gives promising solution in reasonable CPU time and performs well. It was also discovered that the quality of the solution is largely related to the initial setting of parameters such as crossover and mutation rate, population size and migration rate. For this reason, it accomplishes a comprehensive exploration by Taguchi method to find best value of these control parameters. The perfectly tuned algorithm is then compared with 11 available algorithms in the literature using well-known set of benchmark instances. Different analyses conducted on the results, show that the proposed algorithm enjoys the superiority and outperformance over the other algorithms. The results show proposed method generally act more effectively than presented algorithm in literature. Although the performance of algorithms are worse as the size of the problems increase; but the HMPGA can eliminate this deterioration better than other algorithms and perform more robustly. In other word, by increasing the size of the problem and search space, efficiency of the algorithm will be more and more revealed.

Zhang et al., 2014 [ZHA14b] adopted the genetic algorithm to solve the functional areas layout optimization problem of the railway logistics parks. After getting the comprehensive relationship chart of the different functional areas, the paper solved the layout problem with mathematical methods instead of the traditional manual adjustment method. Combined with relevant constraint conditions, the paper constructed the model taking the maximal arithmetic product of comprehensive relationship and adjacency degree as the objective function. Then the article coded with Matlab based on genetic algorithm. In this paper, combining qualitative analysis with quantitative analysis, the functional area layout problem of the logistics park was regarded as a mathematical optimization problem and the uncertainties of layout affected by subjective factors was reduced to a certain extent. The application of genetic algorithm in the layout optimization model greatly improved the quantifiable accuracy of the problem that provided a new thought for the functional areas layout of railway logistics parks.

Gonçalves and Resende, 2015 [GON15] proposed a biased random key genetic algorithm for solving unequal area facility layout problem. Most of the objectives of the problems were to find the location and dimension of the blocks so as to minimize the weighted distance between the blocks. For objective both constrained and unconstrained problems were taken. The solution was represented as the sequence of facilities in which they are placed in the layout. The placements of the facilities are done according to empty maximal space strategy using difference process to generate the spaces. According to the author the unconstrained problem took less time to solve as compare to the constrained problem as a linear programming model was used to improve them in terms of cost and feasibility.

3.5.2.2.4. Particle swarm optimization

Particle Swarm Optimization (PSO) was designed and developed by Kennedy, 1995 and Eberhart, 1997 [EBE95, KEN97]. PSO is a stochastic, population based search algorithm belonging to evolutionary computation techniques. As this is a population based technique the individuals or flock of particles are distributed randomly over the search space. The population of PSO simulates the movement and flocking of birds during the optimization process. In this algorithm, the best individuals in the swarm influence the social behaviour of the particles. The movement of the particles are defined by a certain law to find by best solution from some iterations. During each iteration, the velocity vector of the particles is adjusted based on its momentum, best solution (pbest) and neighbouring best solution (gbest) in order to compute the new point to examine.

Ohmori et.al, 2010 [OHM10] proposed a method to solve facility layout problem using particle swarm optimization. The designed novel continuous optimization approach does not use any special encoding techniques to represent the layout as it searches coordinates of each department continuously. The technique searches the optimal coordinate of each department continuously to overcome the possibility of missing the coordinate the search opportunity caused by encoding techniques. Comparing the results with the previously solved it was shown that the algorithm shows better results for small-sized problems.

Nasab and Emami, 2013 [NAS13] developed a hybrid particle swarm algorithm to solve the dynamic facility layout problem. The facilities in the dynamic facility layout problem were considered to be of equal area. The proposed hybrid PSO was used to find the near optimal solution. The technique uses a coding process that translates the discrete feasible space to continuous space where PSO can work efficiently for exploration. To overcome the drawback of

PSO the algorithm is hybridized by implementation of simulated annealing to search the solution locally.

Zhao et.al, 2014 [ZHA14a] proposed a human interactive particle swarm optimization based immune algorithm to solve a packing and layout in order to realize the man-machine synergy. The initial population in this method are created by the human intelligence through chaotic strategy. Further during the process the evolved artificial generated individual takes over the inferior initial generated individuals. In process of PSO an immunity principle was implemented, where the update operator uses a hybrid strategy i.e. modified rank-based selection and adaptive crossover and mutation for evolution of multi-sub population. The technique was found to provide quality solution and was computationally more efficient for large problems as in the simpler problems the human-computer interaction may require high percentage of total time cost.

Asl et.al, 2016 [ASL16] proposed an improved covariance matrix adaptation evolution strategy (CMA ES) to solve unequal area stochastic FLPs. In this method, the product demands are stochastic with a known variance and expected value and during the iteration the shapes of departments are fixed. In the improved CMA ES two local search methods and a swapping technique was used to change the positions of the departments in order to avoid the local optima and improve the quality of solutions. The results obtained in this method were compared with the results obtained from the two proposed improved particle swarm optimization and genetic algorithm.

3.5.2.2.5. Tabu search

Tabu Search is a global optimization method and a meta-heuristic developed by Glover, 1989 [GLO89A]. In this approach, the current solution $x_t$updates to the best solution x_{t+1}in the feasible space in the neighbourhood $N(x_t)$in each iteration t. As there is no guarantee that the best solution is better than the previous solution, hence a tabu mechanism is used to prevent repetition of the previous solutions. The simplest way to prevent repetition is to prevent a return to all the produced solutions. Though the memory required to store all the solution is excessive. The best possible way to prevent repetition is to save some specification of the previously obtained solutions in the memory and prevent the repetition of such specifications in the next h iteration of the algorithm. The described method is known is as Short Term Memory. The other mechanism used by tabu search is diversification and intensification. In diversification, the algorithm search vast area before finally converging towards a solution. In intensification, the algorithm search for desirable specification in the neighbourhood of the solutions comprehensively [GLO97].

Kothari and Ghosh, 2013 [KOT13] presented two implementations of the tabu search in a single row facility layout problem. In the first implementation involves an exhaustive search of 2-opt neighbourhood whereas the second searches the insertion neighbourhood. Both implementations are parallel multi-start implementations to solve combinatorial optimization problems. These two implementations help in significantly search the two neighbourhoods. Bozorgi et.al, 2015 [BOZ15] proposed a method, which uses both data environment analysis and tabu search to find the material handling cost of a dynamic facility layout. In the process, DEA is applied first to calculate the efficiency of the solutions, and then the tabu search method is used for creating the neighbourhood. The objective of the problem was to find the material handling cost, adjacency and the distance requested. For DEA, the cost was taken as the input whereas adjacency and the distance were taken as the outputs. If two or more solutions have same efficiency, then the solution with lowest material handling cost was chosen as the most efficient layout.

Zuo et.al, 2014 [ZUO14] proposed a method which combined linear programming with multi-objective tabu search for solving extended double row facility layout problem (EDRLP). In the standard double row layout problem (DRLP), the objective is to determine the sequence and location of machines in order to minimize the material handling cost, the problem was extended to take machine floor area for symmetric material flow into consideration, which remains an important factor for many industries.

3.5.3. Hybrid Approaches

Teo and Ponnambalam, 2008 [TEO08] developed a hybrid ACO and PSO heuristic to solve to solve a single row facility layout problem. The clearance and the machine dimensions were also considered as variable (non-linear) to make the representation more realistic which were ignored in the previous researches. Here the ACO is taken as the constructive heuristic for better performance and PSO is used as an improvement heuristic to guide the ants towards the best solution. To further improve the solution a 2-Opt local search method was also implemented.

McKendall et.al, 2010 [MCK10] proposed a hybrid approach to solve the unequal area dynamic facility layout problem. The objective of the problem was to minimize the sum of material handling cost and rearrangement cost for multiple period by finding the best location of the facilities. In the constructive approach a boundary search technique is used to locate the departments one after another in the boundary of already placed departments then tabu search heuristic is used as the improvement approach for finding the best solution.

Buscher et.al, 2014 [BUS14] presented a genetic algorithm based on space filling curve to solve single floor unequal area facility layout problem. The objective of the problem was to minimize the material handling cost. The problem was solved as a discrete layout problem, where the floor area was discretized into rectangular equal area blocks and connected using a continuous space-filling curve. A modified peano curve was used for filling up the floor space. For the constructive approach the blocks were placed in a sequence on the space-filling curve according to the area requirements, the formation may not be a rectangle. The population of the solutions formed by the space filling curve technique was used by genetic algorithm for the improvement approach.

Tasadduq et.al, 2015 [TAS15] proposed a construction-cum-improvement algorithm containing a boundary search heuristic and steepest descent analytic method for solving the facility layout problem. In the construction approach the algorithm places the module one after another in an optimal location on the boundary of the previously placed cluster of modules. In the improvement approach the algorithm alternated between the heuristic boundary search and analytic steepest descent method until it converges towards a local optimum.

Goncalves and resende, 2015 [GON15] proposed a hybrid approach comprised of combined constructive and improvement approaches to solve both unconstrained and constrained cases of unequal area facility layout problem. The constructive decision approach, an empty maximal space (EMS) was used through which the departments are places one by one and for the improvement approach a biased random-key genetic algorithm (BRKGA) was used. In the constructive greedy approach, a list of ems was defined and updated after placement of every department. The department was positioned in the ems where the weighted distances between the departments are minimized. In the improvement approach a BRKGA was used to determine the insertion, their dimensions and the best order of the departments. In the constrained problems, a linear programming model was used to fine-tune the selected solutions.

Xiao et.al, 2016 [XIA16] proposed a combined zone-linear programming and SA algorithm, for solving large-sized UA-FLP. The zone-linear programming is a two-phase technique to construct the facility layout. First phase consists of the zoning algorithm which is used to find the relative positions between the departments. In this phase, the departments are considered as rectangles with allowable aspect ratio. In the second phase the relative positions are then used as the input for Linear programming to determine the exact location and the dimension of the department. The simulated annealing was used to determine the best sequence for placing the departments.

Paes et.al, 2017 [PAE17] implemented simple Genetic Algorithm and a GA combined with partial solution deconstructions and reconstructions decomposition strategy to solve unequal-area facility-layout problems. In the decompose phase a greedy heuristic was used to insert the facilities and the facilities were not allowed to cross the central X and Y axes in the layout space. This strategy produces better results for medium and large instances though the quadrant restriction also deteriorates the value of best achievable solution.

Jacquenot et al., 2009 [JAC09] proposed a hybrid metaheuristics where exploration of the search space and positioning of the free-form components were optimized by genetic algorithm while a separation algorithm [IMA08] was used for relaxation from the placement constraints and validate the solution obtained from GA. Here the geometry of the components for placement was characterized as circles for 2D and spheres for 3D. An example of the 2D representation can be seen in Figure 3.7. The proposed generic method provided high quality solutions with appropriate parameters for the genetic algorithms.

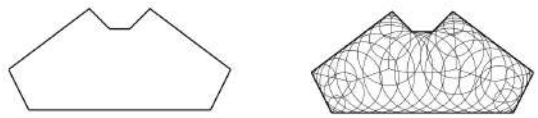


Figure 3.7. 2D representation with circles for a polygon.

3.5.4. Interactive Approaches

Interactive layout optimization design problem is a type of problem in which a user has the liberty to change the parameters and criteria whenever needed for optimization by interaction of user [TAP01, MIE00]. Ultimately, the user-interaction aims at finding the global optimal solution of a problem under target. In layout optimization problem, sometimes it is difficult to obtain a global optimum due to inability of interaction during the process of optimization. However, by provision of interaction tools for parameters or criteria as per the requirement of design based on expertise of designer, it amplifies the possibility to reach the global optimum. The platform allows changing the objective functions, constraints, generations, population and individuals in case of using evolutionary algorithm and other related parameters of optimization algorithms. Generally, the criteria can be qualitative or quantitative criteria.

Michalek and Papalambros, 2002 [MIC02] devised an Interactive Weighted Tchebycheff approach which converts multi-objective layout problem into single objective layout problem by utilization of linear weights. They introduced an interaction tool for architecture layout optimization problem in which the user has the liberty to delete, add or change the objective function, constraints, units, and variables as per the wisdom of the user during the optimization. By ability to change the variable, the optimization search can be guided as per the requirement of designer.

Brintrup et.al, 2006 [BRI06] highlighted that Interactive Evolutionary Computation (IEC) can greatly contribute to improving optimized design by involving users in searching for a satisfactory solution. They used genetic algorithm to solve both single and multi-objective layout problem. The tool was developed with close loop with a provision of selection of either qualitative or quantitative criteria during the optimization of the layout problem. As per this proposed method, the user can choose an option between sequential single objective interactive genetic algorithm and multiple objective interactive genetic algorithms. Nevertheless, both have different structures. Moreover, the qualitative criteria are determined by user-defined rating (values between 0 to 9) for the fitness function. However, the quantitative criteria is determined for fitness function for the given generation count.

Liu et.al, 2008 [LIU08] developed a Human–Algorithm–Knowledge-based layout Design (HAKD) method comprising of a new interactive tool developed for spacecraft layout application. The solution provided with human, algorithm and layout schemes are unified into one string of solution. In this interactive tool, the commercial CAD file of layout is accessed in the genetic algorithm via Hough Transfer technology encoded into an evolutionary algorithm that incorporates the layout schematic made by human user. In HAKD method, creating an individual pool for each solution into a genetic algorithm does the unification for all the three solutions.

Bénabes *et al.*, 2010 [BÉN10] developed an interactive optimization strategy using genetic algorithm and coupled with a separation algorithm for solving a layout problem of a shelter with virtual components as shown in the Figure 3.8. In the problem, the concept of accessibility space was introduced in the layout problem formulation. In this method firstly, the separation algorithm optimizes a population of randomly initialized designs, and then the designer interacts with the generated solutions and selects some individuals according to design constraints.

Secondly, the multi-objective optimizer optimizes the new population by considering all the design objectives and then the designer locally modifies the computed designs to improve the objectives and keep a good diversity in computed solutions.

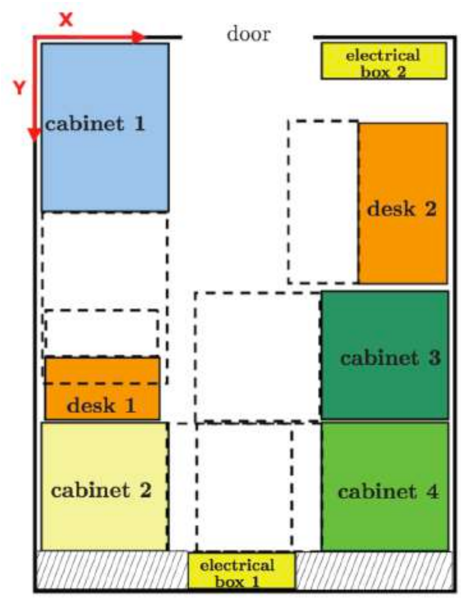


Figure 3.8. 2D representation of a shelter with virtual components.

Gracia et al., 2015 [GAR15] proposed a novel hybrid evolutionary algorithm for the unequal area facility layout problem consists of an interactive genetic algorithm that is combined with two different niching methods to allow interactions between the algorithm and the expert designer. The inclusion of niching techniques into the approach allows for the preservation of diversity, which avoids presenting similar solutions to the designer in the same iteration of the algorithm. Using the suggested approach, it is possible to include a decision makers (DMs) preferences into the design process by means of an interactive genetic algorithm. The DMs knowledge guides the search process towards their preferences without needing to specify them at the beginning of the process. The proposed approach was tested using two case studies of facility layout designs. The experimental results for the two analyzed cases and all of the groups of preferences that were tested show that the method provides satisfactory solutions that do not repeat and is faster than the comparable approach. Additionally, the new approach causes less fatigue and overload of the DM.

3.6. Critical Review

The following are the critical conclusions of the literature survey:

1. In the present literature, there are fair amounts of work containing the optimization techniques with combine effort of local and global search method to solve the facility layout problem. Usually the local search starts after the global search technique and the local search techniques have the tendency to stuck at local optima during the process. But sometimes a minor change in the local search technique may lead to a better solution that is not possible to check for the alternative in the local search.
2. The concept of constructive Bottom-left fill approach [JAK96] in packing problem has gained a lot of attention from researchers due to its compactness. Though the approach has not been applied to the layout problems.

3.7. Problem definition

The outcome of the present literature survey stimulates the following problems:

1. In our approach, a combined local and global search technique has been proposed where the alternatives of a layout solution are checked by the local search method. For this an operator has been introduced to exchange the position of the modules before local search.

2. In another approach, a hybrid constructive and improvement approach has been proposed. For the constructive approach the bottom-left fill approach has been modified to be used for the layout problem.

CHAPTER 4:

HYBRID FACILITY GROWTH HEURISTIC

4.1. Introduction

In this technique for solving an unequal facility layout problem for rectangular facilities an improved heuristic approach has been proposed. The aim is to minimize the sum of distances between the facility and the weighted material handling flows among the facilities. The algorithmic approach uses a GA combined with facility growth heuristic strategy. The heuristic technique used is an advance bottom-left fill technique that enhances the search capabilities. The improvement done in this method is that the positioning of the facilities are allowed to grow freely around the plane in every direction. Here the permutation of the facilities represents the facility layout solution that also gives the order in which the facilities are to be called within the layout space. The facility layout solution generated through this heuristic method is not achieved by most exact methods. The orders of placement of the facilities are done through a greedy construction method. Formation of each individual solution in this technique is itself a local optimization. Sets of the permutations along with equality parameter individual blocks are taken as the initial population for a modified genetic algorithm. As for local optimization, the facilities are represented in the form of nodes used for placement in an optimum way to quasi-static or stepwise minimization of the material handling cost. The stepwise calculation of material handling cost will be equivalent to the final material handling cost. Here, the randomly generated solutions go through a modified genetic algorithm developed accordingly for the solution as represented and keeping the facility layout in mind. The steps in detail for the optimization technique are summarized in the following sections along with the ideas from with which it has emerged.

4.2. Methodology

The problem considered in this study can be explained using Figure 4.1. Figure 4.1 shows a facility layout problem with two blocks of different size. The blocks are to be placed in an area, i.e. within a bigger block, so that material-handling cost between the departments (blocks) is minimum.

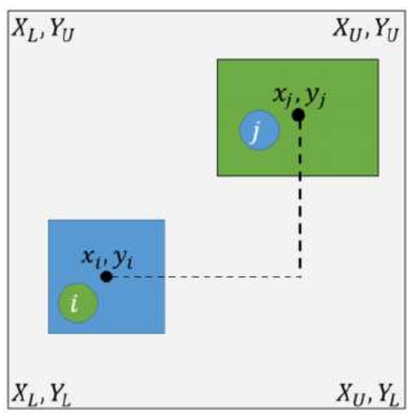


Figure 4.1. A facility layout problem with two blocks of different size.

Let coordinate of the centre of block i is (x_i, y_i) and coordinate of the centre of block j is (x_j, y_j). The distance between the blocks can be calculated by equation 4.1.

$$d_{ij} = |x_i - x_j| + |y_i - y_j| \tag{4.1}$$

If the unit material flow cost between block i and j is c_{ij}, the material flow cost can be calculated by multiplying distance with the unit material flow cost, which is $d_{ij} \cdot c_{ij}$. If there are n numbers of departments and objective of the layout optimization problem is to minimize the cost of material flow between the departments while maintaining non-overlapping constraint, the optimization problem can be formulated as,

Minimize
$$Cost = \sum_{i=1}^{n} \sum_{j=i+1}^{n} d_{ij} c_{ij} \tag{4.2}$$

subject to
$$g_1 = \sum_{i=1}^{n} \sum_{j=1}^{n} A_{ij} = 0\,, i \neq j \tag{4.3}$$

$$g_2 = x_i + \frac{l_i}{2} \leq X_U \tag{4.4}$$

$$g_3 = x_i - \frac{l_i}{2} \leq X_L \tag{4.5}$$

$$g_4 = y_i + \frac{b_i}{2} \leq Y_U \tag{4.6}$$

$$g_5 = y_i - \frac{b_i}{2} \leq Y_L \tag{4.7}$$

where, c_{ij} is the cost of material flow between the departments, d_{ij} is the distance between the departments, A_{ij} is the intersection area of the rectangular departments, X_L is the lower limit of variable x, X_U is the upper limit of variable x, Y_L is the lower limit of variable y, Y_U is the upper limit of variable y, d_{ij} is the distance between the blocks and can be calculated using equation 4.1 and A_{ij} can be calculated using equation 4.8.

$$\begin{aligned} A_{ij} = max\left[0, min\left(x_i + \frac{l_i}{2}, x_j + \frac{l_j}{2}\right) - max\left(x_i - \frac{l_i}{2}, x_j - \frac{l_j}{2}\right)\right] \\ \times max\left[0, min\left(y_i + \frac{b_i}{2}, y_j + \frac{b_j}{2}\right) - max\left(y_i - \frac{b_i}{2}, y_j - \frac{b_j}{2}\right)\right] \end{aligned} \tag{4.8}$$

where (x_i, y_i) and (x_j, y_j) are the coordinates of the centre of department i and j respectively; (l_i, b_i), and (l_j, b_j) are the width and breadth of departments i and j respectively.

4.3. Initial Facility Growth Heuristic

4.3.1. Block's Representation

The representation of the blocks is done to facilitate a possible position for the next block or the incoming block coming for placement in the layout and help in the layout growth. The two type of block representation are mentioned below. The Type II representation is an extension of the Type I representation, and is provided to find better solution and give some aesthetic view to the layout.

4.3.1.1. Type I

The primary idea was taken from bottom-left fill technique where the placement is done at the bottom right or the top left position of the rectangle(s) to the bottom-left corner of the incoming block was placed at placement positions. However, in this primary technique the placement position is taken all around the block instead of only two placement positions as in bottom-left fill technique. The designed placement positions are shown in Figure 4.2. If all the edges of the facilities are started counter clockwise. Then all the possible positions of the next incoming facility with respect to the parent facility can be shown in three sets. The first set of possible positions for the incoming blocks are formed by the placement of the blocks at the vertices of the parent block with the condition that the edges of the parent block and the incoming block should be in contact with each other. Similarly, the second set is formed by the placement of incoming blocks with the centre of the edge of parent block.

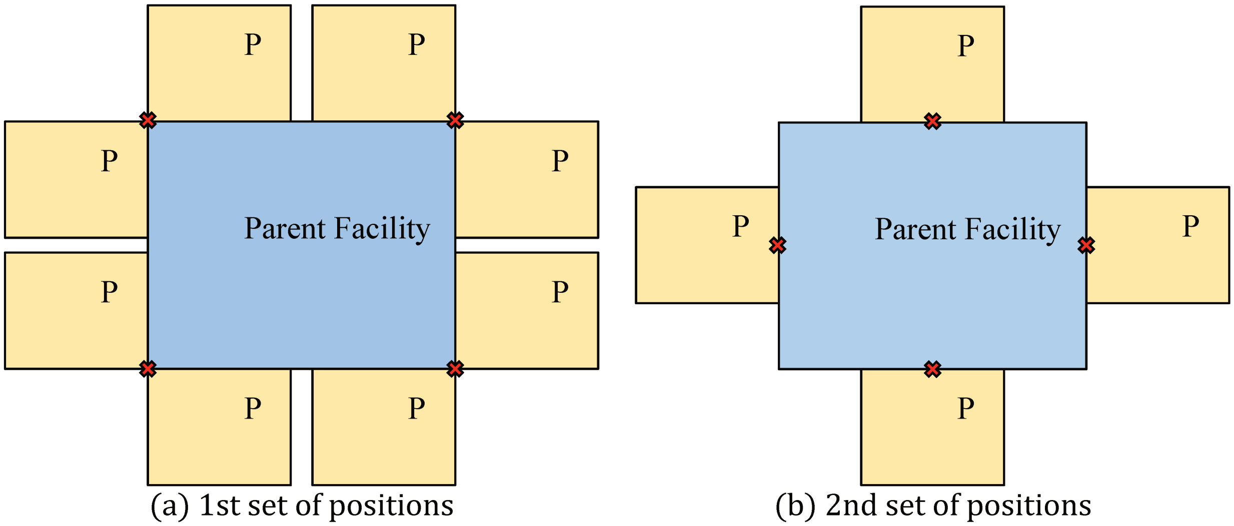
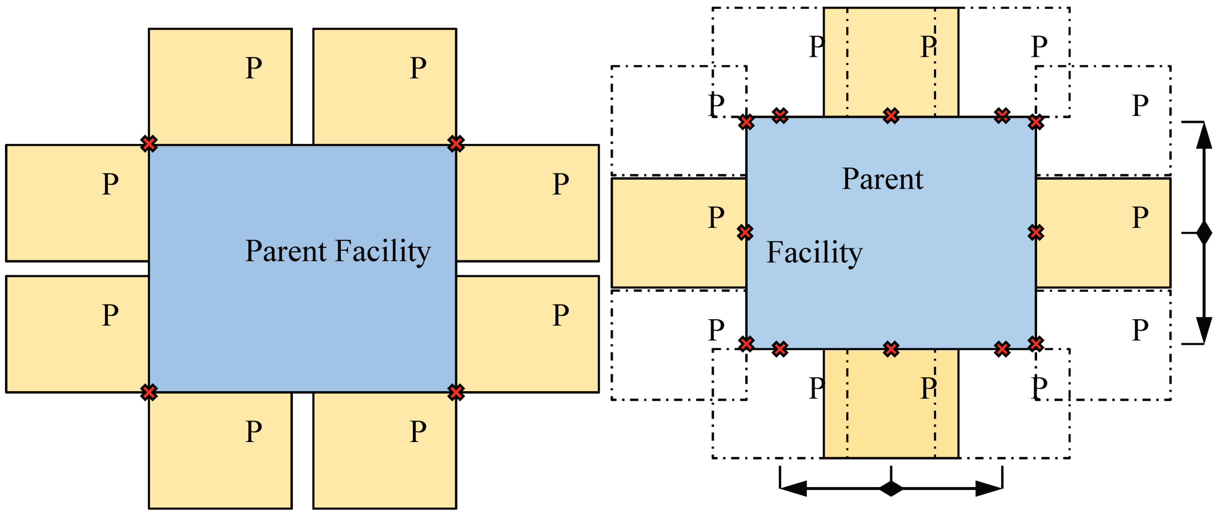


Figure 4.2. Type I possible positions of the incoming facility to a parent facility

The total possible positions for the incoming block is all the combined positions from all two-possible set of positions.

4.3.1.2. Type II

In this type of representation some extra possible positions are added to the Type I representation. The extra possible positions are added to find better solutions than Type I representation and give some aesthetic view to layout as in Type I representation the possible positions are placed only in the center and at the edge of the parent facility. In Type II representation a new discretization parameter is introduced which are useful for keeping extra possible positions which are closer to each other and helpful fill the gap between the center and edge possible position in any side of the parent block. The generalized discretization parameter value is taken as the half of the minimum of all side distance value from the total facility blocks instead of taking any random or unit value. The reference starting point for possible position is at the center of the edge of parent block and proceeds towards both the edges. The edge block being the extreme position and the blocks formed by the using the discretization parameter are not allowed to pass through.



(a) 1st set of positions (b) 2nd set of positions

Figure 4.3. Type II possible positions of the incoming facility to a parent facility

If "*a*" is the discretization parameter value then the possible positions formed from the center possible blocks can be seen in the above representation. In the above figure the number of extra possible blocks is limited to only two blocks, which can more depending upon the distance of the edge block. The total possible positions are the combination of both the sets and if there are any possible positions overlapped with it location can be removed keeping only one.

4.3.2. Solution Representation

The layout solutions are represented in the form of vector of integers and each integer represent the index of the block. Each block index in the layout solution is associated with two other properties of the block. The first property being the rotation parameter and the other is the position parameter of the block. The facility layout solution has been represented as "$X_i = [x_{i1},$

x_{i2}, x_{i3}, …, x_{im}]” where “*x*” represent each department index, “*i*” being the layout solution number and “m” is the number of departments in the layout. Each department in the chromosome is mentioned as $x_{ij} = [b_{ij}\ p_{ij}\ r_{ij}]'$, where “$b_{ij}$” is the department index represented in the form of integer which lies between *1* to *m*., “p_{ij}” is the position parameter which takes any two decimal place value between *0* and *1* and “r_{ij}” is the binary variable for rotation which takes the value either *0* or *1*. The position parameter is important for the placement of incoming blocks which is later described in the section 4.3.5. An example of the solution representation is shown in Table 4.1 in the form of a table.

Table 4.1.

An example of solution representation for the proposed hybrid algorithm.

$$x_{i-j} = \begin{array}{|c|}\hline b_{i-j} \\ \hline p_{i-j} \\ \hline r_{i-j} \\ \hline \end{array}$$

$X_1 =$

Block	b_{1-1}	b_{1-2}	b_{1-3}	b_{1-4}	b_{1-5}	b_{1-6}	b_{1-7}	b_{1-8}	b_{1-9}	b_{1-10}
Pos. Para.	p_{1-1}	p_{1-2}	p_{1-3}	p_{1-4}	p_{1-5}	p_{1-6}	p_{1-7}	p_{1-8}	p_{1-9}	p_{1-10}
Rotation	r_{1-1}	r_{1-2}	r_{1-3}	r_{1-4}	r_{1-5}	r_{1-6}	r_{1-7}	r_{1-8}	r_{1-9}	r_{1-10}

Block	3	9	7	4	10	1	6	5	8	2
Pos. Para.	0.66	0.72	0.36	0.88	0.49	0.64	0.42	0.87	0.29	0.31
Rotation	0	1	0	0	1	0	0	1	1	1

4.3.3. Local Search Algorithm

The solution representation itself is local search optimization technique as in this technique the blocks are introduced one after another according to the representation. Whenever an incoming block is called it is always placed at one of the possible position from all the already placed block giving the minimum material handling cost. As, all the placement of blocks are connected to each other at the lowest material handling cost, it can be said that the solution representation forms a local minimum. The blocks are chosen to be connected due to the following reasons:

1. To simplify the problem as the complexity of a facility layout is usually NP hard.
2. The objective is to minimize the material handling cost and the material handling cost is a function of distance. So, the connected blocks usually give less distance.
3. To overcome some the industrial constrained like the facility space. As connected blocks usually take less layout space then the non-connected blocks and this technique also helps in filling the gap between layout as in bottom-left fill algorithm.

For getting the local minima each block from the facility layout solution is kept one by one in the facility to form a local solution. The idea of the solution representation for local search is like travelling salesman problem with nearest neighbor algorithm. In travelling salesman problem with nearest neighbor algorithm, at every step the salesman always goes to the city with shortest distance from his current position keeping the total distance covered at each travel as current minimum. Somewhat similar in this method the objects are placed one after another in a location to keep the current total material handling cost as minimum. For the idea to be a success while placing the blocks the placement position was defined with stepwise calculation of material handling cost at each placement position. Then the blocks are placed at the placement position giving the current minimum material handling cost. The first block is placed at the center of the facility and then the other blocks are added one after another as shown in Figure 4.4(a-h). Every figure in the group represents local minima for the group of blocks in that figure.

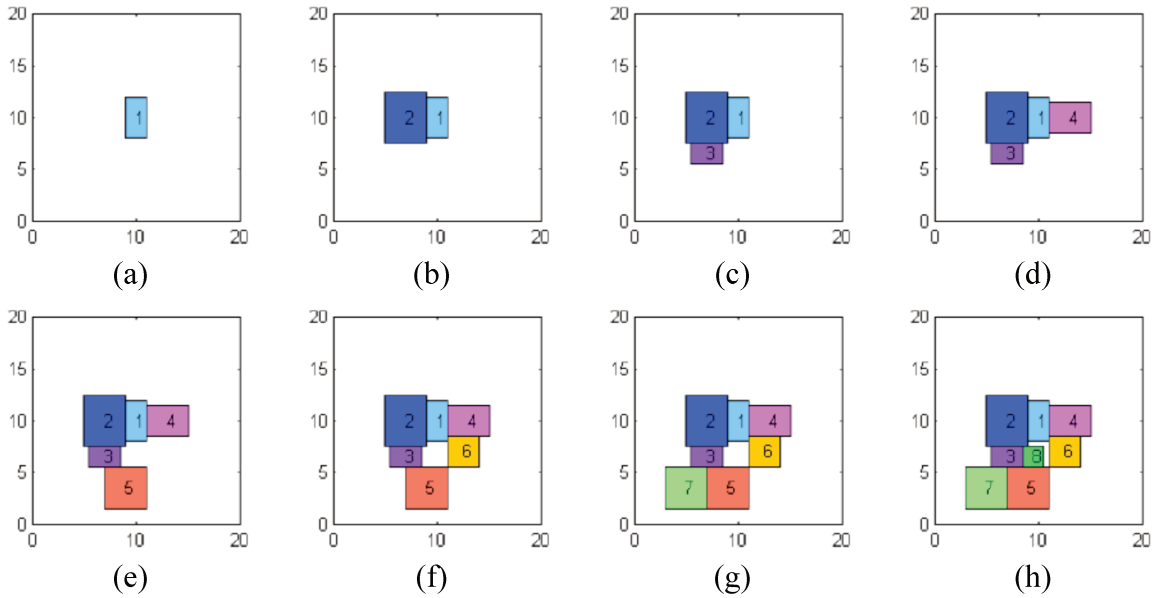


Figure 4.4. Stepwise placement of blocks according to minimum Material Handling Cost.

4.3.4. Stepwise Material Handling Cost Calculation

In this technique, the material handling cost is calculated at each possible position of incoming blocks. For this the cost data matrix is extracted from the total facility layout cost matrix data and the distance matrix from the already placed blocks and the incoming blocks are needed. The distance matrix is calculated between each already placed block and for each placement position of the incoming block with respect to the already placed block. Every time when a new distance matrix is formed, only the distance from the possible placement position is calculated with respect to already placed blocks and is added to the previous distance matrix formed between already placed blocks. Then the total material handling cost is calculated from the summation of the material handling cost from each department. If the extracted cost data is zero for the already placed block and the incoming block then there is no exchange of material between the blocks. In this case, the facility layout solution is altered. Here the incoming block is moved to the end of the facility layout solution and the next incoming blocks are called for placement.

4.3.5. Placement of Incoming Blocks

The incoming blocks are placed at the placement position with minimum material handling cost as compare to the material handling cost of other placement positions. In Figure 4.4 every sub-figure is a representation of local minima when one by one block is introduced. If there are two or more placement positions in a sub-figure giving the same minimum material handling cost, then we choose the position given by the position parameter (p_i) of the incoming block. To do so, all the placement positions of already placed blocks are arranged anticlockwise one after another starting from the reference position taking equally the values between 0 and 1, then the minimum material handling cost is chosen from the right side of the position parameter value which appears first. As, the range of position parameter is from 0 to 1.

Suppose, the number of positions or coordinates of a block is represented as:

$$C_i = (n_{i1}, \ldots, n_{ik}),$$

where, n_{ik} is the total set of positions in block i taken in anticlockwise order starting from the reference position. Then, the arrangement of the position required for the position parameter to function properly can be represented as:

$$A_k = [C_1, C_2, C_3, \ldots, C_m]$$

<or>

$$A_k = [(n_{11}, \ldots, n_{1k}), (n_{21}, \ldots, n_{2k}), \ldots, (n_{m1}, \ldots, n_{mk})]$$

where, m is the total number of facilities in a layout. Now, whenever a new facility is added to the layout the set of possible positions are just added to the right-hand side of A_k. For selection of a block the set of arranged possible positions A_k is mapped from 0 to 1 taken from left side to right side at an equidistant value between any two consecutive positions. Now from the mapped values of the available positions, the position giving the minimum material handling cost that appears to the right side of the position parameter in the mapping is chosen for the positioning the next block. In other words, the position parameter is a value between the mapping of the available positions of the already placed blocks.

The change in position parameter changes the overall layout which can be shown in the example below. The data set of 8 blocks are taken from Imam and Mir, 1993 [IMA93]. Here three chromosomes are taken permutation of the incoming blocks remains the same whereas the position parameter are different in all three chromosomes. From Figure 4.5 it can be seen that the layout of the representations different for all the three cases.

Chromosome 1:

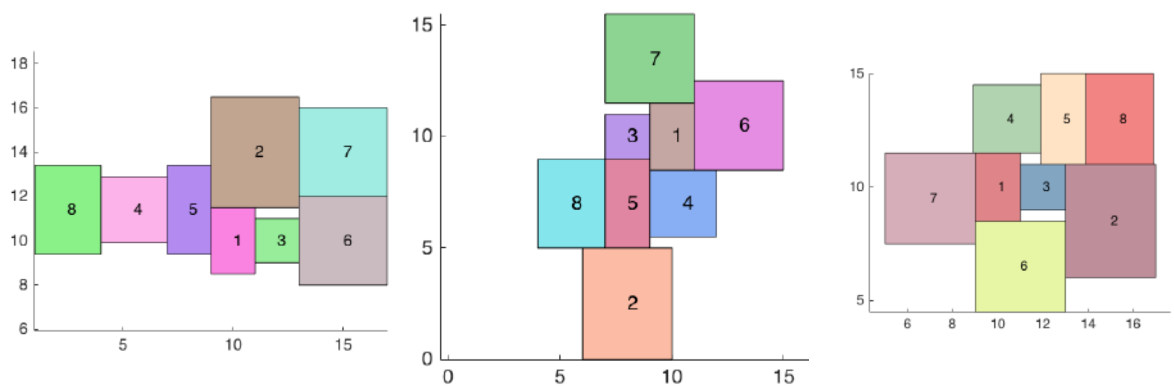
Block	1	3	5	4	8	2	6	7
Pos. Para. (P_i)	0.98	0.96	0.42	0.83	0.43	0.76	0.40	0.50

Chromosome 2:

Block	1	3	5	4	8	2	6	7
Pos. Para. (P_i)	0.14	0.49	0.69	0.34	0.51	0.37	0.31	0.13

Chromosome 3:

Block	1	3	5	4	8	2	6	7
Pos. Para. (P_i)	0.55	0.27	0.96	0.98	0.12	0.48	0.52	0.59



Chromosome 1 Chromosome 2 Chromosome 3

Figure 4.5. Layout for different chromosome with same facility order but different parameter

value.

4.4. Genetic Algorithm

Genetic Algorithms are heuristic algorithms that depend upon the population of solutions in a solution space, which undergoes artificial evolution by genetic operators, and through survival of the fittest strategy the operators help the population of solutions to converge to the optimal solution.

For the given optimization problem at any time interval t the population of solutions is maintained as $P(t) = [x_1, x_2, x_3, \ldots, x_n]$ where x_i is the feasible solution for the problem and n is the population size. This population of solution will undergo evolution towards a feasible solution and the bad solution will either die out or replaced by the offspring during the process. The basic genetic algorithm process is shown below.

begin t = 0

initialize the population P(t)

evaluate the population P(t)

while termination criteria not satisfied do

Population variant P'(t)

Evaluate the population P'(t)

Apply Genetic operators to P'(t) to get next generation population P(t+1)

t = t+1

end while

4.5. Modified Genetic Algorithm

In the modified discrete genetic algorithm, the individual layout solutions in the populations are represented in the form of integers and the operators are formed according to the representation and the layout problem that are discussed in the following section. In the algorithm at any time interval t the population is denoted as $P_t = [L_1, L_2, L_3, \ldots, L_n]$ where L_i is the feasible layout solution for the problem. Now the feasible layout solution $L_i = [x_{i1}, x_{i2}, x_{i3}, \ldots, x_{im}]$ where x represent each department and the m is the number of departments in the layout. Each department in the chromosome is mentioned as $x_{ij} = [b_{ij}\ p_{ij}\ r_{ij}]'$, where b_{ij} is the department index, p_{ij} is the position parameter and r_{ij} is the binary variable for rotation.

4.5.1. Tournament Selection

In tournament selection, the population of solutions form a mating pool and each individuals of the population $P(t)$ randomly play tournament with another individual of same position in the same but randomly arranged population $P'(t)$. After each individual tournament, the individual with the best function value is then declared as the survivor and is allowed to move forward to the participate in the remaining process. In tournament selection, an individual has the opportunity to participate twice in the tournament due to which some individuals with worst solutions are also selected to participate in further process. As sometimes this worst solution may also lead to good solutions with few alterations. The new individuals $l_i = best\ (L_i, L'_i)$ are selected at the end of tournament to form a new list of individuals.

$$[L_1, L_2, \ldots, L_n]'\ vs\ [L'_1, L'_2, \ldots, L'_n]'leads\ to\ [l_1, l_2, \ldots, l_n]'$$

Here, L_i is the layout individual in the i^{th} position of the population, L'_i is the layout individual of the i^{th} individual of the randomly arranged population for tournament selection and l_i is either of L_i or L'_i whichever giving the best function value.

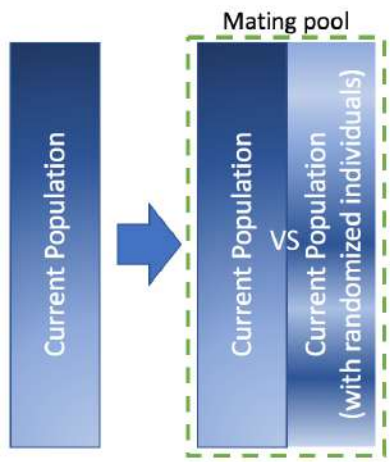


Figure 4.6. Tournament Selection.

4.5.2. Crossover Operator

A new crossover function is defined according to the layout individual representation and the rule followed by each department for placement. For crossover two layout individuals l_i and l_j are chosen at random from the population and a crossover site k is chosen at random. The modules before the crossover site are kept unchanged in the process to transfer the initial formation cluster to the offspring. The two individuals for crossover are shown below.

$$l_i = (x_{i1}, x_{i2}, \dots, x_{ik-1} \parallel x_{ik}, \dots, x_{im}) = [X_{i1} \parallel X_{i2}]$$

$$l_j = (x_{j1}, x_{j2}, \dots, x_{jk-1} \parallel x_{jk}, \dots, x_{jm}) = [X_{j1} \parallel X_{j2}]$$

k - Crossover site

Here the left hand and the right-hand side of the layout individuals from the crossover site k are denoted as X_{i1} and X_{i2}. Now, as for each layout individual as the placement of blocks start from the left side the chromosome, for crossover the left half of chromosome from each individual are exchanged from the crossover site.

$$l'_i = (x_{j1}, x_{j2}, \dots, x_{jk-1} \parallel x_{ik}, \dots, x_{im}) = [X_{j1} \parallel X_{i2}]$$

$$l'_j = (x_{i1}, x_{i2}, \dots, x_{ik-1} \parallel x_{jk}, \dots, x_{jm}) = [X_{i1} \parallel X_{j2}]$$

k - Crossover *site*

Now, after exchange as there may be some similar department remaining in both side of the chromosome, which is not allowed. Therefore, the right sides of the individuals are modified according to the procedure mentioned below keep the left side of the individuals unchanged.

$$l'_i = \left(x_{j1}, x_{j2}, \dots, x_{jk-1} \parallel x'_{ik}, \dots, x'_{im}\right) = \left[X_{j1} \parallel X'_1\right]$$

$$l'_j = \left(x_{i1}, x_{i2}, \dots, x_{ik-1} \parallel x'_{jk}, \dots, x'_{jm}\right) = \left[X_{i1} \parallel X'_2\right]$$

Here, X'_1 and X'_2 are selected from X_{i2} and X_{j2} in order to avoid repeated block in the layout individual.

$$X'_1 = (x'_{ik}, x'_{ik+1}, \dots, x'_{im})$$

Here, x is the elements from X_{j2} which doesn't exist in X_{i2}. In the process, each element is replaced in an order. In this after two offspring are generated from the two parents, the best two of the four layout individual are selected for continuing the optimization process.

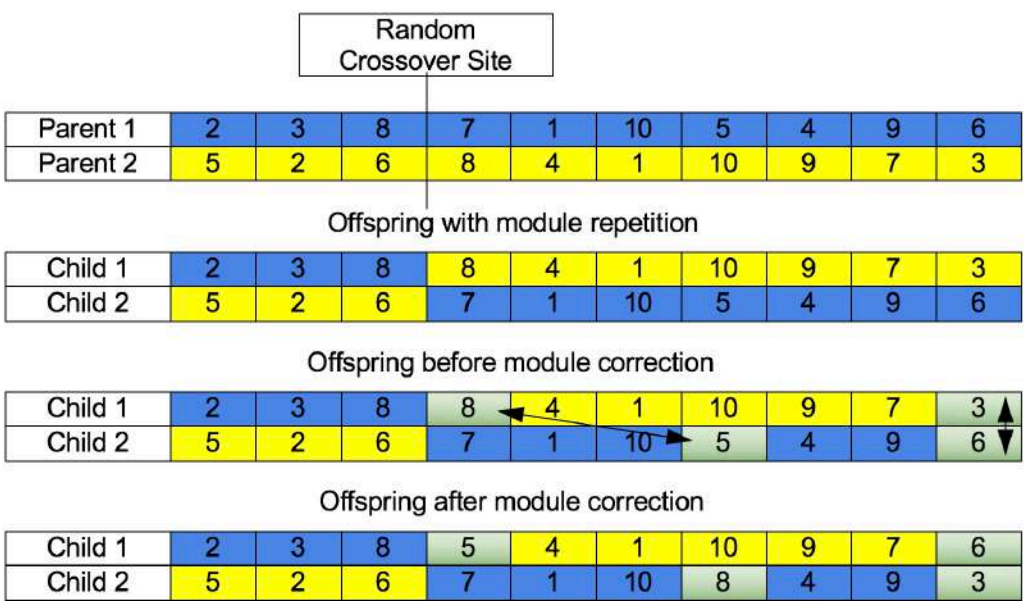


Figure 4.7. Crossover Operator

4.5.3. Mutation Operator

In mutation, a fixed random 20% of the population takes part in the process. During the mutation of a layout individual, two department and their properties are selected randomly for mutation. After, the department number b_{ij} are interchanged with each position, the departmental position parameters p_{ij} are varied randomly within 2% from the current value, and, the departmental rotational factors r_{ij} are changed either 0 or 1 with 0.5 probabilities for getting both values.

Before mutation:

$$l_i = (x_{i1}, x_{i2}, x_{i3}, \dots, x_{ih}, \dots, x_{ik}, \dots, x_{im})$$

$$x_{ih} = \{b_{ih}, p_{ih}, r_{ih}\};$$

$$x_{ik} = \{b_{ik}, p_{ik}, r_{ik}\}$$

After mutation:

$$l_i = (x_{i1}, x_{i2}, x_{i3}, \dots, x_{ik}, \dots, x_{ih}, \dots, x_{im})$$

$$x_{ih} = \begin{cases} b'_{ih} = b_{ik} \\ p'_{ih} = random\ selection\ in\ p_{ik}\ within\ 2\%\ bounds \\ r'_{ih} = 0\ or\ 1 \end{cases}$$

$$x_{ik} = \begin{cases} b'_{ik} = b_{ih} \\ p'_{ik} = random\ selection\ in\ p_{ik}\ within\ 2\%\ bounds \\ r'_{ik} = 0\ or\ 1 \end{cases}$$

Here, x_{ih} and x_{ik} are the two departments from the layout individual l_i which are selected at a random. It should be noted that if the mutation occurs with the first module of the chromosome then there is a high probability that the formation of the layout changes as the initial cluster formation changes entirely. This helps in maintaining diversity in the population

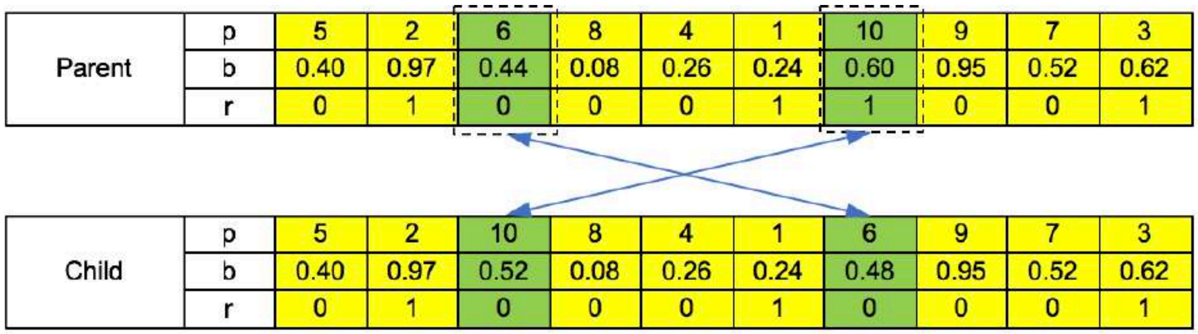


Figure 4.8. Mutation Operator

4.6. Results and Discussion

In the proposed algorithm for both type I and II the population size was set to 60, 80 and 100 and the termination criteria was fixed at 20, 30 and 60 generation respectively for 8, 11 and 20 module problems as from practical experience of the simulation it was seen that the fitness obtained has marginal difference from the previous generations. We have considered here three benchmark problems to evaluate the performance of the proposed model. Asl and Wong, 2015 [ASL15] have solved these three problems using modified particle swarm algorithm and the problems have been described below. The problem is also compared with the results obtained by Mir and Imam, 2001 [MIR01]; Imam and Mir, 1993 [IMA93]; and Imam and Mir, 1989 [IMA89]. The boundary condition for each problem was checked after the placed placement of the modules. Though the algorithm starts with a predefined layout dimension a valid layout can go beyond it as the condition are checked by measuring the difference between the maximum and the minimum coordinate of all the modules in the axes. The final layout dimension is set keeping the total layout i.e. the cluster of modules at the center. The result for the three problems has been shown below.

4.6.1. 8 Blocks problem

The problem has eight departments, which are to be placed in an area of 12x12 square unit. The unit cost of material flow between the departments and size of the departments are given in Table 4.2. The objective of the problem is to find out the best possible locations of the department so that total material flow cost is minimum. The best solution (minimum cost) obtained by Asl and Wong, 2015 was 193.7488 and the simulation time was 220.69 seconds.

Table 4.2.

Material flow from each department and the length and width of each department for 8 blocks.

Department	Cost of material flow (Cij) between the Departments								Length	Width
	1	2	3	4	5	6	7	8		
1	0	1	2	0	0	0	2	0	2	3
2	0	0	4	3	6	0	0	2	4	5
3	0	0	0	2	0	3	1	0	2	2
4	0	0	0	0	5	2	0	2	3	3
5	0	0	0	0	0	0	0	4	2	4
6	0	0	0	0	0	0	4	0	4	4
7	0	0	0	0	0	0	0	1	4	4
8	0	0	0	0	0	0	0	0	3	4

Table 4.3.

Solution of Problem 1

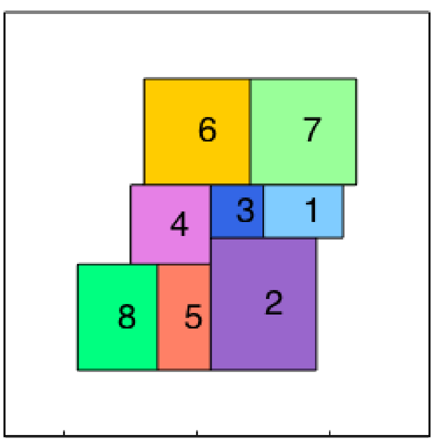
Department	1	2	3	4	5	6	7	8
xc	9.2674	7.7674	6.7674	4.2674	4.7674	5.2674	9.2674	2.2674
yc	6.1712	2.6712	6.1712	6.1711	2.6711	9.6711	9.5205	2.6711
Rotation	1	0	0	0	0	0	0	0

The objective of the problem was to find the best location of the departments to minimize the material handling cost. A total of 20 simulations were carried out for each representation (i.e. Type I and II). The best and average solution obtained by Asl and Wong, 2015 was 193.7488 and

208.74 respectively by modified PSO. The solution obtained from the proposed hybrid constructive and improvement algorithm is 191.5 for both type I and II representation which is more than the combined approach but much better than the solution obtained by Asl and Wong, 2015. The solution obtained by combined approach is better than hybrid approach as the combined approach being continuous has full has full access to the solution space unlike the hybrid approach which is discrete method. The coordinates of the solution obtained by both the block representation remains same and is listed in the table below. This maybe have occurred because of the following reasons

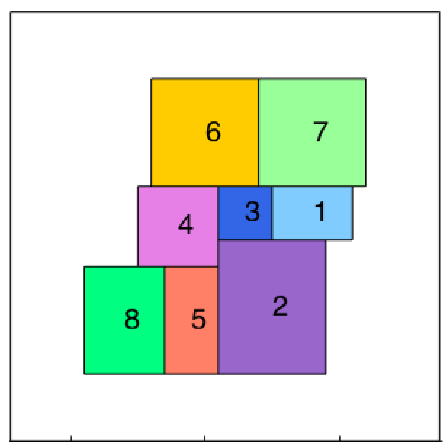
1. The initial conditions are same.
2. There are small numbers of block compared to the other problem.
3. The minima lies in 12 points Type I representation.

The data shown in the table are rotated and changed according to the original orientation of problem. The layout of the departments for both representations is shown in the Figure 4.9.



Type I: 8 block problem (191.5)

time = 264.5751s



Type II: 8 block problem (191.5)

time = 269.8297s

Figure 4.9. Optimal Solution for Problem 1

Table 4.4.

Results for 8 blocks problem for Type I and II representation

X	12.5	10.5	10.5	7.5	8.5	8.0	12.0	5.5
Y	6.0	1.0	6.0	5.0	1.0	8.0	8.0	1.0
Length	3.0	4.0	2.0	3.0	2.0	4.0	4.0	3.0
Width	2.0	5.0	2.0	3.0	4.0	4.0	4.0	4.0

4.6.2. 11 Blocks problem

The second problem considered here is more complex and has 11 departments, which are to be placed in an area of 15x15 square unit. The unit cost of material flow between the departments and size of the departments are given in Table 4.5. The objective of the problem is to find out the best possible locations of the departments so that total material flow cost is minimum. The best solution obtained by Asl and Wong, 2015 was 1286.1069 and the simulation time was 888.31 seconds.

Table 4.5.

Material flow from each department and the length and width of each department for 11 blocks.

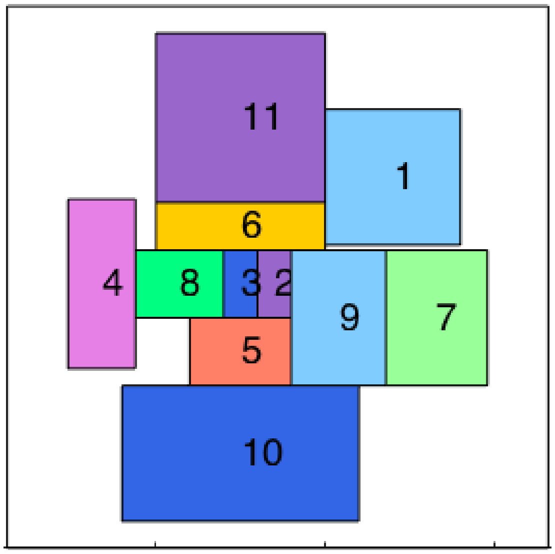
Department	Cost of material flow (C_{ij}) between the Departments										Length	Width
	1	2	3	4	5	6	7	8	9	0	1		
1	0	2	2	1	2	6	2	6	6	3	6	4.0	4.0
2	0	0	1	1	2	6	4	6	6	3	6	1.0	2.0
3	0	0	0	2	2	6	1	6	6	6	6	1.0	2.0
4	0	0	0	0	1	5	1	6	6	3	6	2.0	5.0
5	0	0	0	0	0	4	3	6	4	5	6	3.0	2.0
6	0	0	0	0	0	0	3	6	4	5	6	1.4	5.0
7	0	0	0	0	0	0	0	4	4	1	1	4.0	3.0
8	0	0	0	0	0	0	0	0	6	3	3	2.6	2.0
9	0	0	0	0	0	0	0	0	0	5	5	4.0	2.8
10	0	0	0	0	0	0	0	0	0	0	2	4.0	7.0
11	0	0	0	0	0	0	0	0	0	0	0	5.0	5.0

For initialization, the layout space was taken from (0,0) to (15,15). The unit cost of material flow between the departments and size of the departments are shown in Table 4.5. The objective of the problem is to find the best position of all the departments to minimize the material handling cost. The best and average solution obtained by Asl and Wong, 2015 was 1286.1069 and 1335.63 respectively and the simulation time for the best solution was 888.31 seconds. For this problem, we have got even better results than that of Asl and Wong, 2015. For the proposed model, the best, average and worst solution obtained for type I representation are 1208.3, 1222.8 and 1236.5 respectively and for type II representation are 1209.6, 1229.77 and 1244.6 respectively. The solution obtained from type I representation with 12 points representation has better results than type II representation with more points than Asl and Wong, 2015; Mir and Imam, 2001; Imam and Mir, 1993 & Imam and Mir, 1989. This may have happened due to the following reasons.

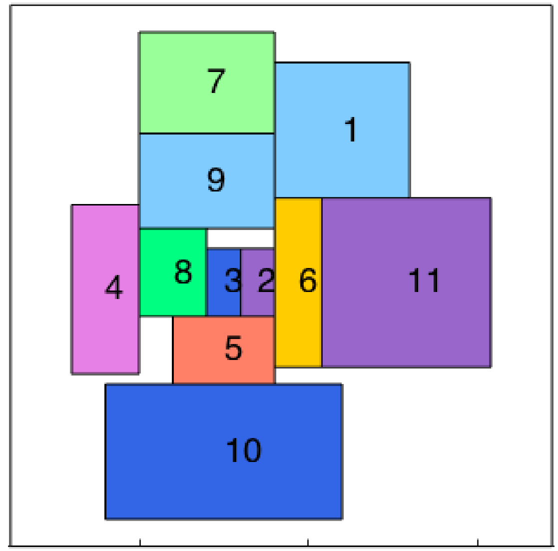
1. Since the facility layout problem are NP hard problems, the greedy constructive search method in the proposed hybrid algorithm worked well with less points.

2. The minima lies in the Type I – 12 points representations.

The data shown in the Table 4.6 Table 4.7 are rotated and changed according to the original orientation of problem. The layout of the departments for both representations is shown in Figure 4.10.



Type I: 11 block problem (1208.3)
time = 1441.1s



Type II: 11 block problem (1209.6)
time = 941.2033s

Figure 4.10. Optimal Solution for Problem 2

Table 4.6.

Results for 11 blocks problem for Type I representation

X	10.0	8.00	7.00	2.40	6.00	5.00	11.8	4.40	9.00	4.00	5.00
Y	8.65	6.50	6.50	5.00	4.50	8.50	4.50	6.50	4.50	0.50	9.90
Length	4.00	1.00	1.00	2.00	3.00	5.00	3.00	2.60	2.80	7.00	5.00
Width	4.00	2.00	2.00	5.00	2.00	1.40	4.00	2.00	4.00	4.00	5.00

Table 4.7.

Results for 11 blocks problem for Type II representation

X	9.00	8.00	7.00	3.00	6.00	9.00	5.00	5.00	5.00	4.00	10.4
Y	12.0	8.50	8.50	6.80	6.50	7.00	13.9	8.50	11.1	2.50	7.00
Length	4.00	1.00	1.00	2.00	3.00	1.40	4.00	2.00	4.00	7.00	5.00
Width	4.00	2.00	2.00	5.00	2.00	5.00	3.00	2.60	2.80	4.00	5.00

4.6.3. 20 Blocks problem

The third problem considered here is again a complex problem with 20 departments, which are to be placed in 13x13 square unit area. The unit costs of material flow between the departments are shown in Table 4.8 and the length and width of the departments are shown in Table 4.9. Asl and Wong, 2015 also solved this problem using modified particle swarm algorithm. The best solution obtained by them was 1206.6489 and the simulation time was 2250.86 seconds. The best solution obtained from hybrid method was 1148.5 and 1166.5 from Type I and II representation respectively. The average and the worst for type I representation is 1173.55 and 1198.5 and for type II is 1191.9 and 1229.5 respectively. The 20-block problem is a difficult problem to solve due to the presence of more number of variables. Again, the proposed model with type I representation has found the better solution than the type II representation and other methods compared. The coordinate of the departments corresponding to the solution obtained from both type I and II representation is shown in the Table 4.10. The layouts of the departments are also in Figure 4.11.

Table 4.8.

Material flow from each department and the length and width of each department for 20 blocks.

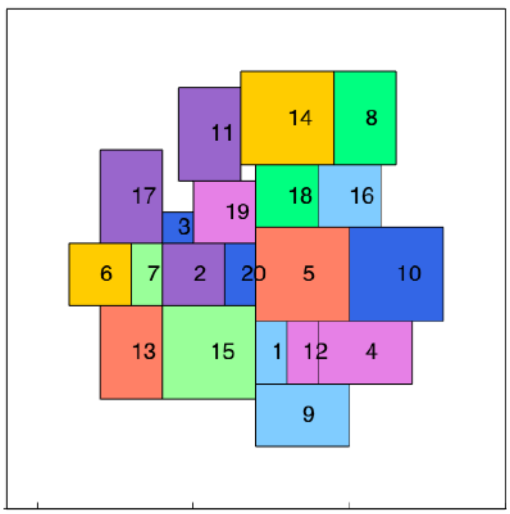
Department	Cost of material flow (C_{ij}) between the Departments																			
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	0	3	0	0	4	2	0	0	4	0	0	5	3	0	5	0	0	1	0	0
2	3	0	1	0	1	2	5	0	3	0	0	0	2	0	3	0	3	1	2	3
3	0	1	0	4	0	0	3	0	0	0	1	0	0	0	0	0	5	0	2	3
4	0	0	4	0	4	0	0	1	5	3	0	2	0	0	4	5	0	1	0	0
5	4	1	0	4	0	0	0	0	1	4	1	5	0	0	3	2	0	5	0	4
6	2	2	0	0	0	0	3	0	0	5	0	0	3	0	0	0	2	0	0	0
7	0	5	3	0	0	3	0	0	0	0	0	0	4	0	2	0	3	2	0	1
8	0	0	0	1	0	0	0	0	0	0	2	0	0	5	0	4	0	1	0	0
9	4	3	0	5	1	0	0	0	0	3	0	5	0	0	0	2	0	0	0	0
10	0	0	0	3	4	5	0	0	3	0	0	5	0	1	2	4	0	3	4	0
11	0	0	1	0	1	0	0	2	0	0	0	0	0	5	5	4	0	4	3	1
12	5	0	0	2	5	0	0	0	5	5	0	0	5	0	2	0	0	1	0	0
13	3	2	0	0	0	3	4	0	0	0	0	5	0	0	3	0	2	0	0	0
14	0	0	0	0	0	0	0	5	0	1	5	0	0	0	0	5	0	5	1	0
15	5	3	0	4	3	0	2	0	0	2	5	2	3	0	0	0	1	4	3	3
16	0	0	0	5	2	0	0	4	2	4	4	0	0	5	0	0	4	5	0	0
17	0	3	5	0	0	2	3	0	0	0	0	0	2	0	1	4	0	0	1	5
18	1	1	0	1	5	0	2	1	0	3	4	1	0	5	4	5	0	0	4	1

19	0	2	2	0	0	0	0	0	0	4	3	0	0	1	3	0	1	4	0	5
20	0	3	3	0	4	0	1	0	0	0	1	0	0	0	3	0	5	1	5	0

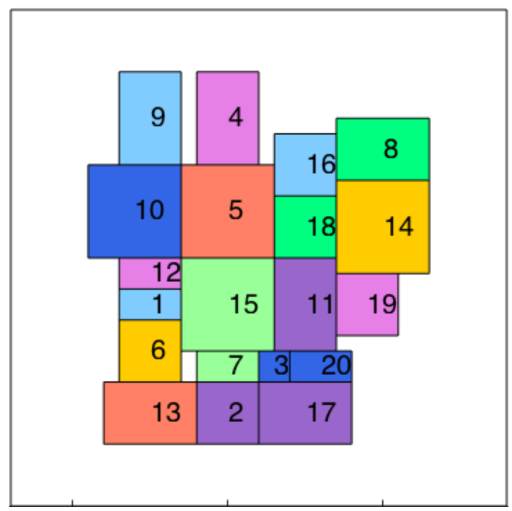
Table 4.9.

Length and width of each department of 20 blocks

Department	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Length	1	2	1	2	3	2	2	2	3	3	2	1	3	3	3	2	3	2	2	2
Width	2	2	1	3	3	2	1	3	2	3	3	2	2	3	3	2	2	2	2	1



Type I: Layout for 20 module problem
(1148.5)



Type II: Layout for 20 block problem
(1209.6)

Figure 4.11. Optimal Solution for Problem 3

Table 4.10.

Results for 20 block problem with type I representation

X	12	9	9	14	12	6	8	14.5	12	15
Y	6.5	9	11	6.5	8.5	9	9	13.5	4.5	8.5
Length	1	2	1	3	3	2	1	2	3	3
Width	2	2	1	2	3	2	2	3	2	3

X	9.5	13	7	11.5	9	14	7	12	10	11
Y	13	6.5	6	13.5	6	11.5	11	11.5	11	9

Length	2	1	2	3	3	2	2	2	2	1
Width	3	2	3	3	3	2	3	2	2	2

Table 4.11.

Results for 20 block problem with type II representation

X	6.5	9	11	9	8.5	6.5	9	13.5	6.5	5.5
Y	6.5	2.5	4.5	11.5	8.5	4.5	4.5	11	11.5	8.5
Length	2	2	1	2	3	2	2	3	2	3
Width	1	2	1	3	3	2	1	2	3	3

X	11.5	6.5	6	13.5	8.5	11.5	11	11.5	13.5	12
Y	5.5	7.5	2.5	8	5.5	10.5	2.5	8.5	6	4.5
Length	2	2	3	3	3	2	3	2	2	2
Width	3	1	2	3	3	2	2	2	2	1

4.7. General Discussion of the Proposed Model

The efficiency of the proposed hybrid model is better compared to that of the combined local and global search approach by Mir and Imam, 2001; Imam and Mir,1993 & Imam and Mir, 1989. The problem of 8 blocks unequal area rectangular layout problem was initially proposed by Imam and Mir, 1989 and was solved using an analytic method. Later the difficulty of the problem was increased to 11 departments by Imam and Mir, 1993, which was solved by using a heuristic method. Further the problem was complexity of the problem was increased to 20 blocks by Mir and Imam, 2001 which was solved by using modified simulated annealing. Asl and Wong, 2015 solved the three problems with their modified particle swarm optimization algorithm. In our approach of hybrid method to solve the three problems, the constructive approach was inspired by the bottom-left fill approach where two more available placement positions added after placement of each departments in a bin packing problem. The available positions were increased around the departments instead of right-bottom position & top-left position and represented into two types namely I and II. The number of available positions around the departments also acts as local search in the context of advanced bottom-left constructive approach. In the continuous approach the local search method takes more time to minima whereas due to correct positioning of available positions (i.e. at the center and corner of each side) in each department leads to lesser time with less points of find the minima. As for two blocks, if a department center position of longest side is positioned at the available position at the center of longest side of another block, then it achieves minimum material handling cost between the two. The corner two positions at each side make sure that the space is filled up when more department are placed side by side to each other to attain a compact packing of departments. The modified genetic algorithm was used as the improvement approach, which was modified according to the chromosome representation. This method being a discrete method with less available search space outperformed the compared continuous approach with an improvement of 1.15%, 6.04% and 4.82% for type I and 1.15%, 5.94% and 3.32% for type II then the results obtained by Asl and Wong, 2015 for 8, 11 and 20 blocks respectively. This shows that the proposed model has the capability to handle large size layout problem efficiently except smaller department problem where the limited solution space makes it harder to find the solution. The comparisons of the proposed hybrid model with other continuous models are shown in Table 4.12 -Table 4.13.

Table 4.12.

Comparison of the solutions obtained by our proposed hybrid method with other continuous methods

Methods	Problem 1: 8 modules			Problem 2: 11 modules			Problem 3: 20 modules		
	Best	Mean	Worst	Best	Mean	Worst	Best	Mean	Worst
Type I	191.5	192.5	193	1208.3	1222.8	1236.5	1148.5	1173.6	1198.5
Type II	191.5	192.25	193	1209.6	1229.8	1244.6	1166.5	1191.9	1229.5
Combined local and global	191	192.53	199.52	1253.7	1293.2	1327.8	1171.5	1219.5	1253.6
Asl & Wong (2015)	193.74	208.74	-	1286.1	1335.6	-	1206.6	1264.2	-
Mir and Imam (2001)	-	-	-	-	-	-	1225.4	1287.3	-
Imam and Mir (1993)	-	-	-	-	-	-	1264.9	1333.8	-
Imam and Mir (1989)	-	-	-	-	-	-	1320.7	1395.6	-

Table 4.13.

Percentage improvement in percentage

Problem Type	Representation	Combined Local and Global	Asl and Wong (2015)	Mir and Imam (2001)	Imam and Mir (1993)	Imam and Mir (1989)
8 blocks	Type I	-0.26	1.15	-	-	-
	Type II	-0.26	1.15	-	-	-
11 blocks	Type I	3.62	6.05	-	-	-
	Type II	3.52	5.95	-	-	-
20 blocks	Type I	1.96	4.82	6.28	9.21	13.04
	Type II	0.43	3.33	4.81	7.78	11.68

4.8. Conclusion

The time taken by the hybrid approach to find the best solution is much lesser than the combined local and global search and other continuous algorithms. The implementation of the discrete greedy was helpful in finding the minima much quicker than other local search methods leading to lesser time taken. In the chromosome representation, a new parameter was introduced to distinguish between similar sequences of departments. The parameter acted as a percentage of available position after which positions with similar minima are selected. Further, the greedy advanced bottom left construction of the departments also gave a best solution till now which shows that an improvement of 13% can be achieved from the proposed method. The problem also handled the rotation variables very well as it didn't decrease the efficiency of the algorithm. Although the algorithm was implemented on a static facility layout problem it can also be further extended to a dynamic facility layout problem or a multi-floor problem or a facility location problem.

CHAPTER 5:

COMBINED GLOBAL AND LOCAL SEARCH

In this chapter, a new evolutionary and classical algorithm based hybrid optimization method has been proposed for solving static facility layout problems with the unequal size of compartments. The facility layout problem is a mixed integer problem if the rotation of the compartments is considered in the design. To avoid the mix-integer form of the problem, this study proposed a rotation operator. Use of the rotation operator has also reduced the number of variables of the problem significantly. The objective function of the problem is non-linear in which the sum of the material handling cost has been minimized. Apart from the conventional evolutionary operators, i.e. selection, crossover, mutation and elitism, this paper has also used exchange and rotation operators. The performance of the model is tested using previously solved problems selected from the literature.

5.1. Iterative Search Importance

In the section 3.5.2 the general comparison of the Construction and Iterative approaches has been written. In construction, usually the boundaries of the departments are in contact with each other to form a cluster and finally the final layout of the problem. As they remain in contact with each other, the construction approaches are limited to solve a limited number of objectives. Some of the objective which are not possible or more complex to be solved by constructive approaches are fixed distance between two or more departments, departments in contact to the boundary of the layout space, aesthetic objectives etc.

The problem faced by the constructive approaches is easily handled by the iterative approach. The iterative approaches handle almost every type of objective functions.

For example, if in an objective function the weight(s) between distance between the departments are taken as negative then it will not be possible for the constructive or hybrid constructive and iterative approach (from previous chapter) to be able to solve it. Negative weights maybe taken in cases where certain components are kept away from each other to prevent any accident. The problem with negative weights can be easily tackled by just iterative approach. To verify, a test problem of 8 departments of equal size is considered and a unit value is set as the weightage for the distance between the departments. The weights are also taken as negative for analysis purpose. The values of negative weightage are also decreased further to show their effect in the layout. The following cases taken into consideration:

a. Unit positive weight between all departments,

b. Negative weight between two departments,

c. A department with negative weightage for all other department,

d. Negative weightage on some departments, and

e. Negative weightage for all departments.

The weights of the following cases are shown from Table 5.1 -Table 5.5. For each optimization, a maximum number of iterations of the BFGS algorithm has been set to 1000.

Table 5.1: Case a: Unit positive weight between all department.

Department	Cost of material flow (Cij) between the Departments								Length	Width
	1	2	3	4	5	6	7	8		
1	0	1	1	1	1	1	1	1	3	3
2	0	0	1	1	1	1	1	1	3	3
3	0	0	0	1	1	1	1	1	3	3
4	0	0	0	0	1	1	1	1	3	3
5	0	0	0	0	0	1	1	1	3	3
6	0	0	0	0	0	0	1	1	3	3
7	0	0	0	0	0	0	0	1	3	3
8	0	0	0	0	0	0	0	0	3	3

Table 5.2: Case b: Negative weight between two department.

Department	Cost of material flow (Cij) between the Departments								Length	Width
	1	2	3	4	5	6	7	8		
1	0	1	1	1	1	1	1	1	3	3
2	0	0	1	1	1	1	1	1	3	3
3	0	0	0	1	1	1	1	1	3	3
4	0	0	0	0	-100	1	1	1	3	3
5	0	0	0	0	0	1	1	1	3	3
6	0	0	0	0	0	0	1	1	3	3
7	0	0	0	0	0	0	0	1	3	3
8	0	0	0	0	0	0	0	0	3	3

Table 5.3: Case c: A department with negative weightage for all other department.

Department	Cost of material flow (Cij) between the Departments								Length	Width
	1	2	3	4	5	6	7	8		
1	0	1	1	1	1	1	1	1	3	3
2	0	0	-1	-1	-1	-1	-1	-1	3	3
3	0	0	0	1	1	1	1	1	3	3
4	0	0	0	0	1	1	1	1	3	3
5	0	0	0	0	0	1	1	1	3	3
6	0	0	0	0	0	0	1	1	3	3
7	0	0	0	0	0	0	0	1	3	3
8	0	0	0	0	0	0	0	0	3	3

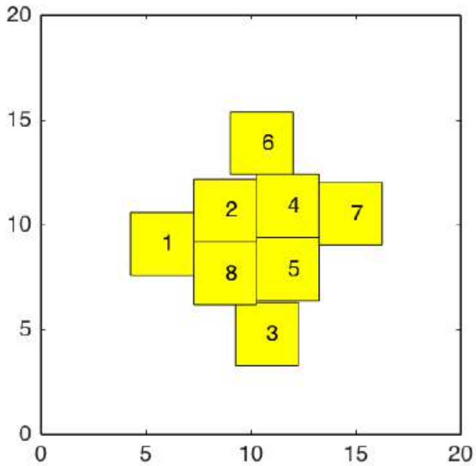
Table 5.4: Case d: Negative weightage on some departments.

Department	Cost of material flow (Cij) between the Departments								Length	Width
	1	2	3	4	5	6	7	8		
1	0	1	1	-1	-1	-1	-1	-1	3	3
2	0	0	1	-1	-1	-1	-1	-1	3	3
3	0	0	0	-1	-1	-1	-1	-1	3	3
4	0	0	0	0	1	1	1	1	3	3
5	0	0	0	0	0	1	1	1	3	3
6	0	0	0	0	0	0	1	1	3	3
7	0	0	0	0	0	0	0	1	3	3
8	0	0	0	0	0	0	0	0	3	3

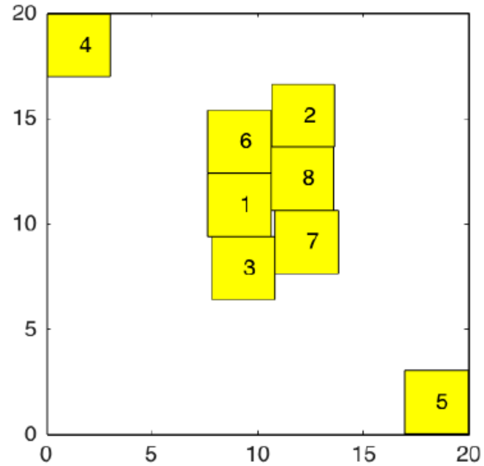
Table 5.5: Case e: Negative weightage for all departments

Department	Cost of material flow (Cij) between the Departments								Length	Width
	1	2	3	4	5	6	7	8		
1	0	-1	-1	-1	-1	-1	-1	-1	3	3
2	0	0	-1	-1	-1	-1	-1	-1	3	3
3	0	0	0	-1	-1	-1	-1	-1	3	3
4	0	0	0	0	-1	-1	-1	-1	3	3
5	0	0	0	0	0	-1	-1	-1	3	3
6	0	0	0	0	0	0	-1	-1	3	3
7	0	0	0	0	0	0	0	-1	3	3
8	0	0	0	0	0	0	0	0	3	3

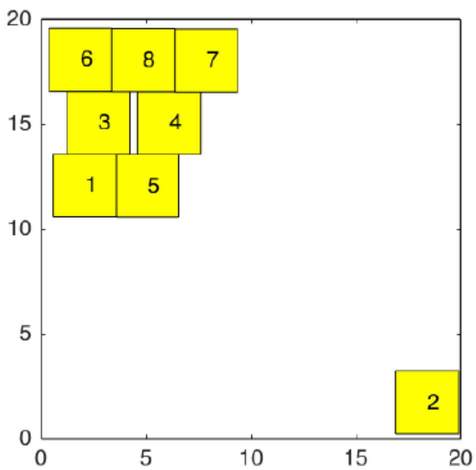
The sum of the Euclidean distance is taken as the objective function to visualize the effect of the negative weights. The constraints taken are the overlap between the departments and the departments and the layout space. The solution representation of the layout for different cases can be seen in the Figure 5.1.



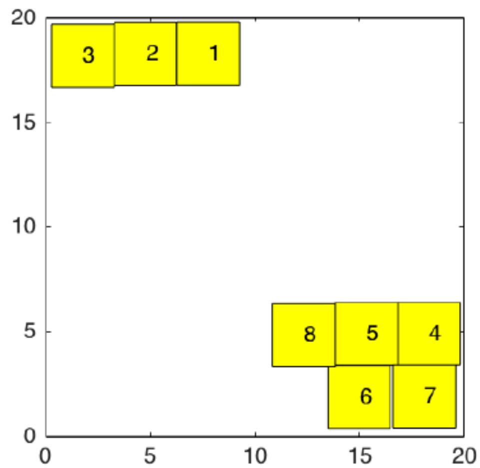
Case (a)



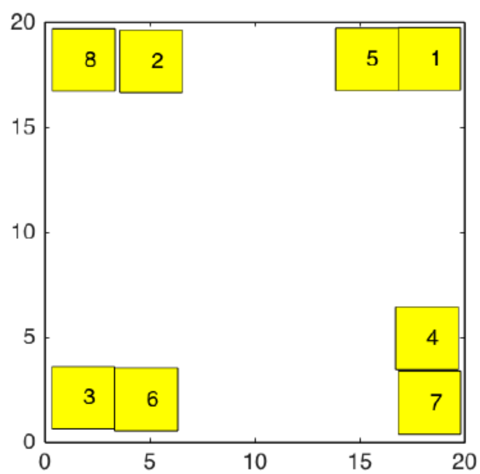
Case (b)



Case (c)



Case (d)



Case (e)

Figure 5.1. Components representation in a layout for different possibilities.

5.2. Methodology

The problem considered in this study can be explained using Figure 5.2. Figure 5.2 shows a facility layout problem with two blocks of different size. The blocks are to be placed in an area, i.e. within a bigger block, so that material-handling cost between the departments (blocks) is minimum. Let coordinate of the centre of block i is (x_i, y_i) and coordinate of the centre of block j is (x_j, y_j). The distance between the blocks can be calculated from equation 5.1.

$$d_{ij} = |x_i - x_j| + |y_i - y_j| \tag{5.1}$$

If the unit material flow cost between block i and j is c_{ij}, the material flow cost can be calculated by multiplying distance with the unit material flow cost, which is $d_{ij} \cdot c_{ij}$. If there are n number of departments and objective of the layout optimization problem is to minimize the cost of material flow between the departments while maintaining non-overlapping constraint, the optimization problem can be formulated as,

Minimize

$$Cost = \sum_{i=1}^{n} \sum_{j=i+1}^{n} d_{ij} c_{ij} \tag{5.2}$$

subject to

$$g_1 = \sum_{i=1}^{n} \sum_{j=1}^{n} A_{ij} = 0\,, i \neq j \tag{5.3}$$

$$g_2 = x_i + \frac{l_i}{2} \leq X_U \tag{5.4}$$

$$g_3 = x_i - \frac{l_i}{2} \leq X_L \tag{5.5}$$

$$g_4 = y_i + \frac{b_i}{2} \leq Y_U \tag{5.6}$$

$$g_5 = y_i - \frac{b_i}{2} \leq Y_L \tag{5.7}$$

where, c_{ij} is the cost of material flow between the departments, d_{ij} is the distance between the departments, A_{ij} is the intersection area of the rectangular departments, XL is the lower limit of variable x, XU is the upper limit of variable x, YL is the lower limit of variable y, YU is the upper limit of variable y, d_{ij} is the distance between the blocks and can be calculated using Equation 5.1 and A_{ij} can be calculated using Equation 5.8.

$$A_{ij} = max\left[0, min\left(x_i + \frac{l_i}{2}, x_j + \frac{l_j}{2}\right) - max\left(x_i - \frac{l_i}{2}, x_j - \frac{l_j}{2}\right)\right] \times max\left[0, min\left(y_i + \frac{b_i}{2}, y_j + \frac{b_j}{2}\right) - max\left(y_i - \frac{b_i}{2}, y_j - \frac{b_j}{2}\right)\right] \quad 5.8$$

where (x_i, y_i) and (x_j, y_j) are the coordinates of the centre of department i and j respectively; (l_i, b_i), and (l_j, b_j) are the width and breadth of departments i and j respectively.

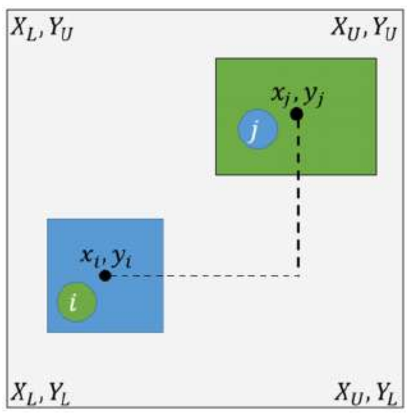


Figure 5.2. A facility layout problem with two blocks of different size.

5.2.1. Population Based Local Search Model

The classical gradient-based method can be applied to solve the problem. But it can only obtain the local optimal solution. The main reason is that the search carried out by gradient-based method traps in local optimal solutions and there is no mechanism to come out of the local optima. It may be mentioned here the problem under consideration has several local and alternate optimal solutions. On the other hand, population-based methods, such as Genetic Algorithms [BÉN10], Simulated Annealing [SZY95], Particle swarm optimization [ASL15], etc. have the mechanism to avoid local optimal solutions, but they are not good in local search, i.e. to find out the exact optimal solution of a problem. Moreover, if the rotation is considered in the layout optimization problem, the optimization problem becomes a mixed-integer programming problem and special method is necessary for handling this type of problems. Binary coded genetic algorithms can be used to handle integer variables. But it this case the number of variable in the problem will be large. It is worth mentioning here that genetic algorithms are not very efficient when the number of variables is too large. Motivating with this problem, this study proposes a population-based local search technique for searching global optimal solution of the facility layout problem.

The algorithm can be explained using Figure 5.3. In the first step, we have generated initial solution randomly. We have considered two arrays shown in Figure 5.4. The first array consists of the decision variables, *i.e.* the position of the departments and the second array contains the dimension of the departments. The upper half of Figure 5.4 shows the array containing the decision variables and the lower half of Figure 5.4 shows the array containing the dimension of the departments. Each string in the decision variable array contains the position of the centre of the departments Figure 5.5(a) and each string in the dimension of the department array contains the dimension of a department Figure 5.5(b). The initial solutions, *i.e.* the decision variable arrays are generated randomly between upper and lower bounds of the decision variables. We have considered the location of the centre of the departments as the decision variables. As such, the lower and upper bounds of the variables can be defined as,

$$\left.\begin{aligned} x_{lb}^{i} &= X_L + l_i/2 \\ x_{ub}^{i} &= X_U - l_i/2 \\ y_{lb}^{i} &= Y_L + b_i/2 \\ y_{ub}^{i} &= Y_U - b_i/2 \end{aligned}\right\} \qquad 5.9$$

Considering these initial randomly generated solutions and the dimension of the departments, we have performed local search taking each string as the initial solution for the local search algorithm. The solutions obtained by the local search are the local optimal solutions. The solutions are then checked for termination criteria. If termination criteria are satisfied, the best solution will be reported. Else the solution will go through the *Exchange*, *Rotation*, *Interchange,* and *Elitism* operators. These operators will create new initial solutions for local search including the change in orientation of the departments by the *Rotation* operator. The new initial solutions are then going through the local search algorithm to find the new set of optimal solutions. The new solutions may contain better solution than that have been obtained in the previous step. But it may create inferior solutions also. As such the *Elitism* operator is used to preserving the best individuals of the new and old solutions. This iteration is to be continued till termination criteria are not satisfied. The pseudo code of the algorithm can be written as follows.

```
// pseudo code to implement population based local search method
initialize population;
perform local search;
iteration = 0;
while iteration <= max_iteration do
    perform exchange;
    perform rotation;
    perform interchange;
    perform local search;
```

perform *elitism*;

iteration = iteration +1;

end while

return *population* (optimized)

In the following section, we have discussed the operators used in the model.

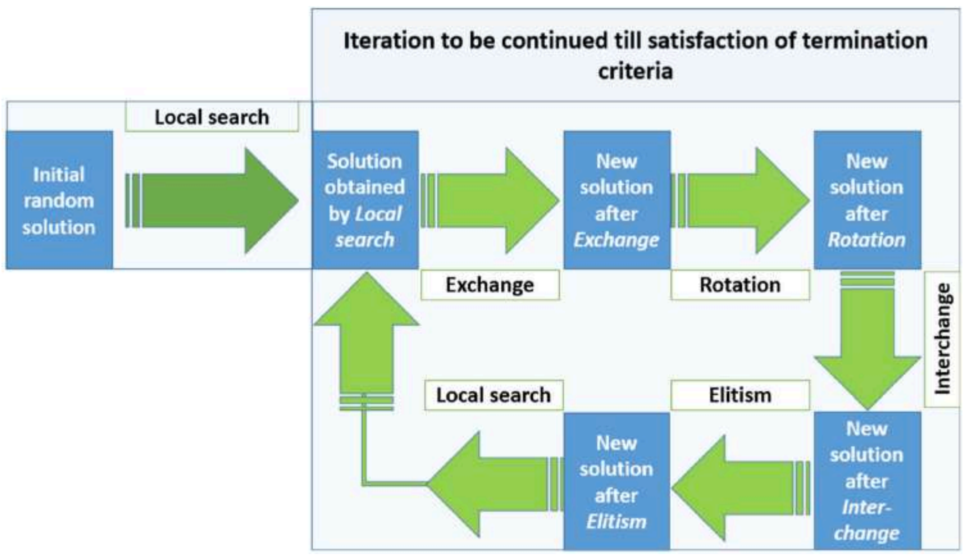


Figure 5.3. Flowchart shows the modified genetic algorithms based search model.

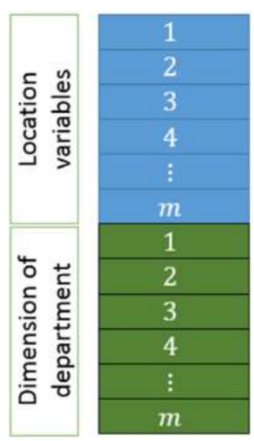


Figure 5.4. Population showing locational and dimensional variables.

(a)	Location Variable	x_1	x_2	x_3	x_4	$\cdots$	x_{n-1}	x_n
		y_1	y_2	y_3	y_4	$\cdots$	y_{n-1}	y_n
(b)	Dimension of Dept.	l_1	l_2	l_3	l_4	$\cdots$	l_{n-1}	l_n
		b_1	b_2	b_3	b_4	$\cdots$	b_{n-1}	b_n

Figure 5.5. Chromosome of locational variable. **(b)** Chromosome of dimensional variable.

5.2.2. Local Search

The local search is carried out using interior point algorithm [BYR00]. This algorithm solves an approximate minimization problem of the constrained minimization sequentially. A non-linear optimization problem can be written as

$$\begin{aligned} &\text{Minimize} && f(X) \\ &\text{subject to} && g(X) \leq 0 \\ & && h(X) = 0 \end{aligned} \qquad 5.10$$

The approximate problem can be written as

$$\begin{aligned} &\text{Minimize} && f_s(X,S) = f(X) - \kappa \sum_i ln(s_i) \\ &\text{subject to} && g(X) + S = 0 \\ & && h(X) = 0 \end{aligned} \qquad 5.11$$

Restricting κ and si to be positive, the function fs act as a barrier function. The minimum of *fs* should approach the minimum of *f* as *κ* tends to zero. We have implemented this algorithm using the *fmincon* function available in Matlab.

5.2.3. Exchange

The exchange operation is performed between two solutions of the population. The solutions participated in the exchange operation are selected randomly from the population. An exchange site is selected randomly between 1 and (n−1), where n is the number of decision variables of the problem. The portion on the right-hand side of the strings is then swap to create two children strings. This operation is similar to the binary crossover operation of genetic algorithms. However, it is performed on a real string. The basic objective of exchange operator is to create two new solutions by combining two old solutions. The exchange operation has been shown in

Figure 5.6.

Figure 5.6(a) shows the two parents before the exchange. The crossover site is selected randomly and then the exchange is performed.

Figure 5.6(b) shows the two children created by the exchange operation. The idea of the exchange operation is that the combination of some portion of two solutions may create a better solution which can even be improved by using the local search technique. Figure 5.7 shows an example of exchange operator. Figure 5.7(a) shows the two parents before exchange and Figure 5.7(b) shows the two children created by the exchange operator. In this case, the exchange site was three.

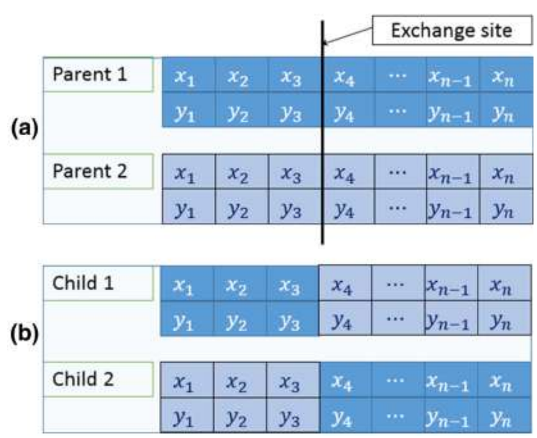


Figure 5.6. **(a)** A chromosome before exchange operation. **(b)** A chromosome after exchange operation.

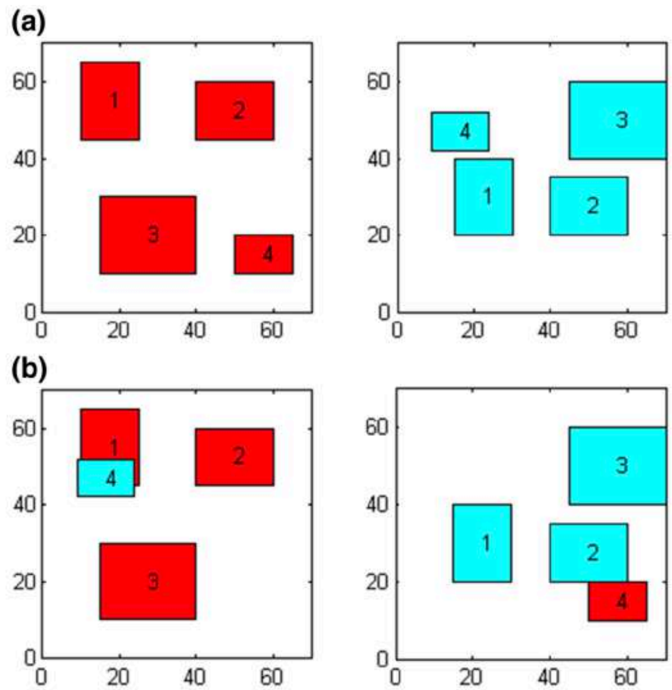


Figure 5.7. (**a**) Two parents before exchange operation. (**b**) Two parents after exchange operation.

5.2.4. Rotation

Rotation operation is performed on the dimension array. As such location array is not participating in this process. The rotation is performed about the centre of the block. Each string of the population will pass through the rotation operator. The cursor will be initially placed at the first block and a random number is generated between 0 and 1. A rotation probability is considered to control the rotation of the block. If the generated random number is less than the rotation probability, the block will be rotated. Else the block will not be rotated. The cursor is then passed to the next block. This process is then continued to all the blocks of the string. The rotation operation has been shown in

Figure 5.8.

Figure 5.8(a) shows a string before rotation. In this case, the random number generated for the third block is less than the rotation probability and hence length and breadth of the block have been interchanged.

Figure 5.8(b) shows the string after rotation. Figure 5.9 shows an example of rotation. Figure 5.9(a) shows a layout of the blocks before rotation and Figure 5.9(b) shows the layout of the blocks after applying rotation operator.

The rotation probability has to be defined by the user. In general, rotation probability has to be kept very low. If has been observed from the experiments that rotation probability of 1/n is appropriate for better convergence. It may be noted that in this study rotation has not be considered as a variable. Therefore, the optimization model has only the continuous variables which can be easily handled using classical optimization model. Incorporation of rotation variable converts the optimization problem to a mixed integer problem as rotation variable is Boolean variable. Specialized technique is necessary to solve the resulting integer problem. Apart from the Boolean nature of the rotation variable, the incorporation of rotation also increases the number of decision variables of the problem. Thus, the adopted techniques can reduce the dimension of the optimization problem substantially and will also avoid the Boolean variables.

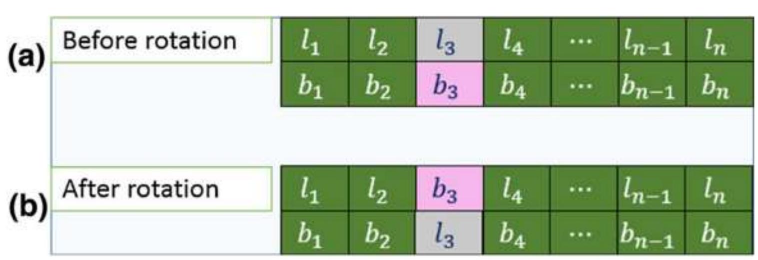


Figure 5.8. **(a)** A chromosome before applying rotation operation. **(b)** A chromosome after applying rotation operation.

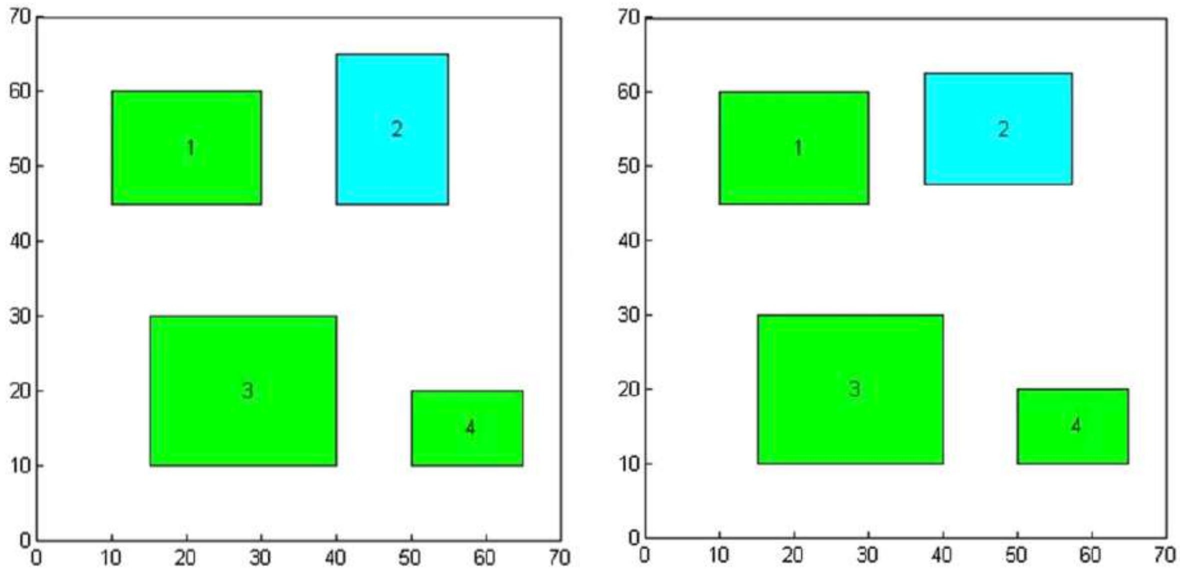


Figure 5.9. (a) A solution before applying rotation operator. (b) A solution after applying rotation operator.

5.2.5. Interchange

Sometimes it may be possible to obtain a better solution by simply interchanging the location of two blocks of a particular solution. However, this may be possible to achieve by using the interchange operation. This operator is executed by generating two random numbers between 0 and n. The position of the two blocks selected randomly is then interchanged. An interchange probability is considered to control the interchange of the blocks. If the generated random number is less than the interchange probability, the block will be interchanged. Else the block will be not be interchanged. The interchange probability is to be kept very low as we should not allow all the strings to participate in interchange. The interchange operation has been shown in Figure 5.10. Figure 5.10(a) shows the string before interchange. In this case, second and fourth blocks have been selected to participate in the interchange operation. The positions of the blocks are then interchanged which has been shown in Figure 5.10(b). Figure 5.11 shows an example of interchange operator.

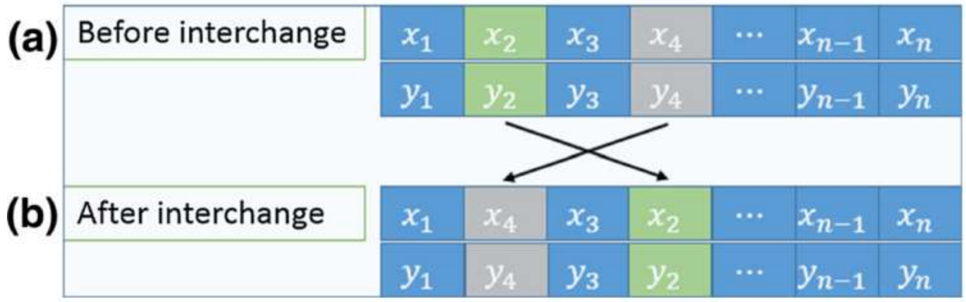


Figure 5.10. (**a**) A chromosome before applying interchange operation. (**b**) A chromosome after applying interchange operation.

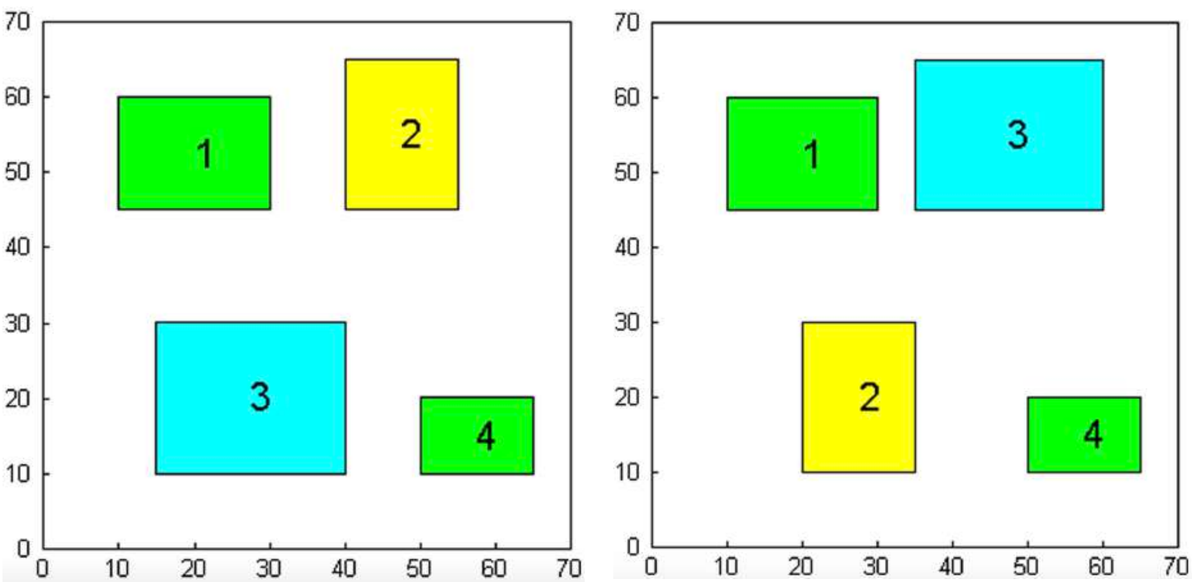


Figure 5.11. (a) A solution before applying interchange operation. (b) A solution after applying interchange operation.

5.2.6. Elitism

It is expected that implementation of exchange, rotation and interchange operators along with local search may create a better solution. However, it may not create a better solution in every iteration. As such the *elitism* operator is adopted to preserve the best individuals of the old and new population. This operator is implemented by combining the new and old population. The total population is then sorted as per the fitness value of the solutions. The best *m* (population size) solutions are then selected for the next iteration. Figure 5.12 shows the elitism operation.

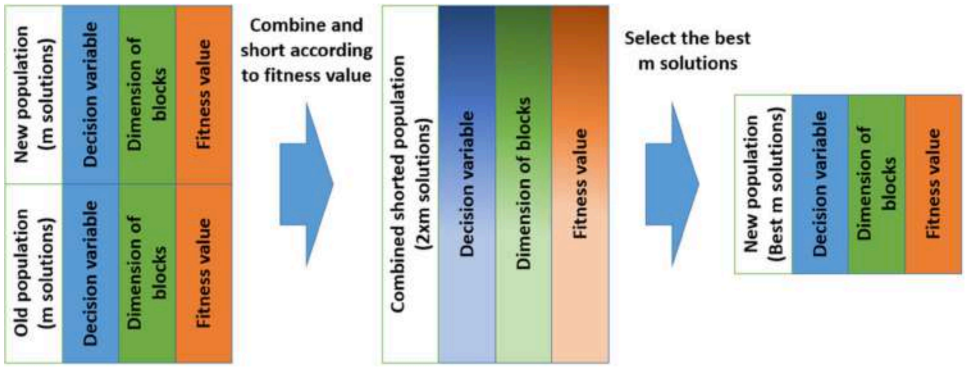


Figure 5.12. Elitism Operator

5.3. Results and Discussion

We have considered here three benchmark problems to evaluate the performance of the proposed model. Asl and Wong, 2015 have solved these three problems using modified particle swarm algorithm. They have reported that the results obtained by them are better than that of the known optimal solutions available in the literature.

5.3.1. Problem 1: 8 Blocks Problem

The problem of 8 departments by Mir and Imam, 2001, which is defined in the previous chapter, has been taken for the study. The results obtained by Asl and Wong, 2015 is 193.7488 with the simulation time 220.69 seconds and the results obtained by constructive and iterative approach is 191.5 with the bet simulation time of 264.57 seconds. In the proposed model twenty simulation runs have been carried out. The best solution obtained is 191.001. The worst and average solutions are 192.53 and 199.52. The mean value of solutions obtained by modified PSO method is 208.74. This shows that the solutions obtained by the proposed model are better than that obtained by Asl and Wong, 2015. The coordinates along with the rotation value of the departments corresponding to the best solution are listed in Table 5.6. The rotation value is 0 for all the blocks except the first block. Rotation value of '0' indicates that there is no change in orientation for these departments. The orientation of the first department has changed by 90 degrees from its original orientation. Figure 5.13(a) shows the placement of the departments, which will give total material flow cost of 191.001. Figure 5.13(b) shows an alternate solution that has material flow cost of 191.002. This shows that the problem has an alternate optimal solution apart from other local optimal solutions. Further, it may be observed that the solutions obtained by the proposed method are more compact than that of obtained by the modified PSO method.

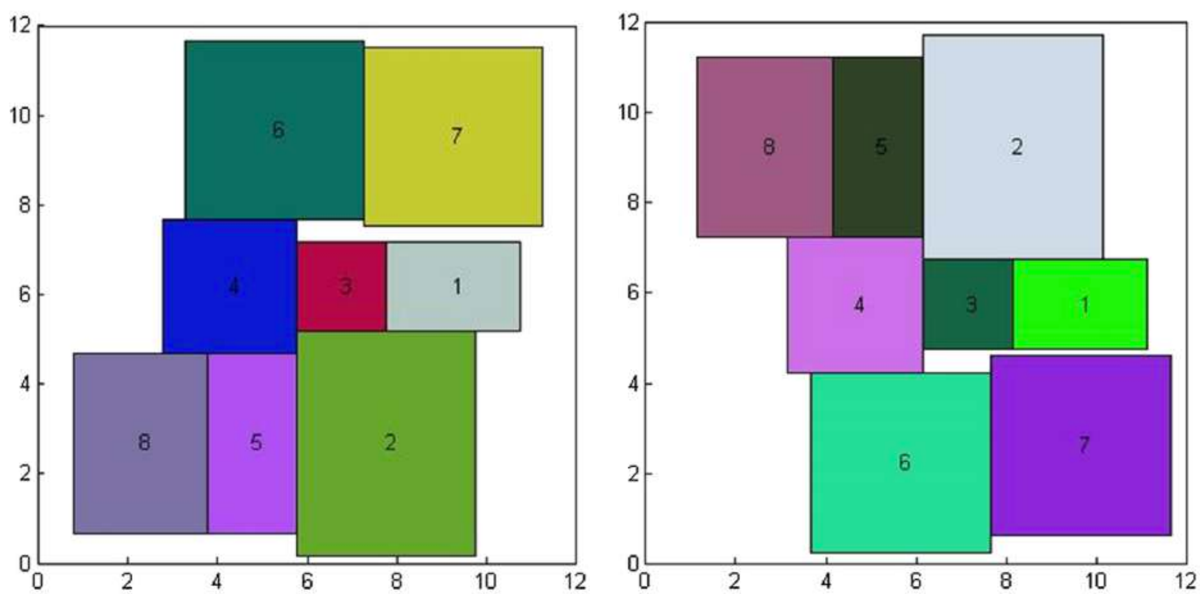


Figure 5.13. (a) An optimal solution of problem 1. (b) An alternate optimal solution of problem 1.

Table 5.6.

Solution of Problem 1

Department	1	2	3	4	5	6	7	8
xc	9.2674	7.7674	6.7674	4.2674	4.7674	5.2674	9.2674	2.2674
yc	6.1712	2.6712	6.1712	6.1711	2.6711	9.6711	9.5205	2.6711
Rotation	1	0	0	0	0	0	0	0

5.3.2. Problem 2: 11 Blocks Problem

The problem considered is complex and contains 11 blocks. The problem has been defined in the previous chapter in section 4.6.2. For this problem, we have got even better results than that of obtained by Asl and Wong, 2015. The best solution obtained by the proposed model is 1253.70. The average and worst solutions obtained are 1293.19 and 1327.80 respectively. The coordinates along with the rotation value of the departments corresponding to the best solution are listed in Table 5.7. In this case, the rotation values of all the departments are zero, which indicates that there is no change in the orientation of the departments. Figure 5.14(a) shows the placement of the departments, which will give total material flow cost of 1253.7. We have considered population size of 10. The material flow costs achieved by all the 10 solutions of the population in a run are 1253.7, 1264.6, 1269.8, 1274.6, 1274.8, 1279.5, 1280.1, 1284.8, 1285.5, and 1287.7. It can be observed that the all the solutions except the last solution are better than the solution obtained by Asl and Wong, 2015. This shows the proposed model is a robust one and has the capability to obtain all the alternate local and global optimal solutions in a single run. Figure 5.14(b) shows an alternate placement of the departments, which has material flow cost of 1269.8.

Table 5.7.

Solution of Problem 2

Department	1	2	3	4	5	6
xc	11.0888	6.7030	6.6592	3.7479	8.6651	5.4592
yc	12.5887	9.1823	7.1699	8.7021	7.1475	8.7038
Rotation	0	0	0	0	0	0

Department	7	8	9	10	11
xc	4.2884	8.5528	7.6884	12.1651	7.3367
yc	12.7038	9.1495	12.1822	7.0886	3.6369
Rotation	0	0	0	0	0

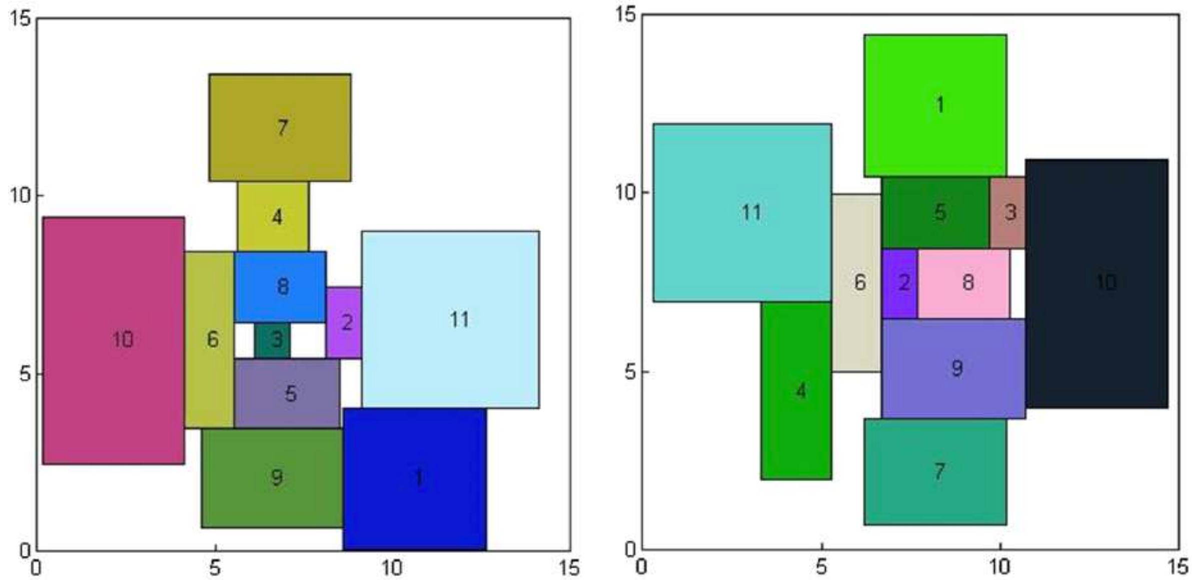


Figure 5.14. (a) An optimal solution of problem 2. (b) An alternate optimal solution of problem 2

5.3.3. Problem 3: 20 Blocks Problem

The third problem taken into consideration is of 20 departments. For initialization, the layout space was taken from the coordinate (0, 0) and (13, 13). The data considered for the required problem is written in the previous chapter 4.6. This is a difficult problem as the number of the departments is 20. We have solved this problem using the proposed model and the minimum material flow cost obtained by the model is 1171.5 which is quite better that the solution obtained by Asl and Wong, 2015. The average value is 1219.48 and worst is 1253.6. Table 5.8 shows the coordinate of the departments corresponding to the best solution, *i.e.* of 1171.5. Figure 5.15(a) and Figure 5.15(b) shows the layout of the department, which gives the material flow cost of 1171.5 and an alternate optimal solution of 1185.00 respectively.

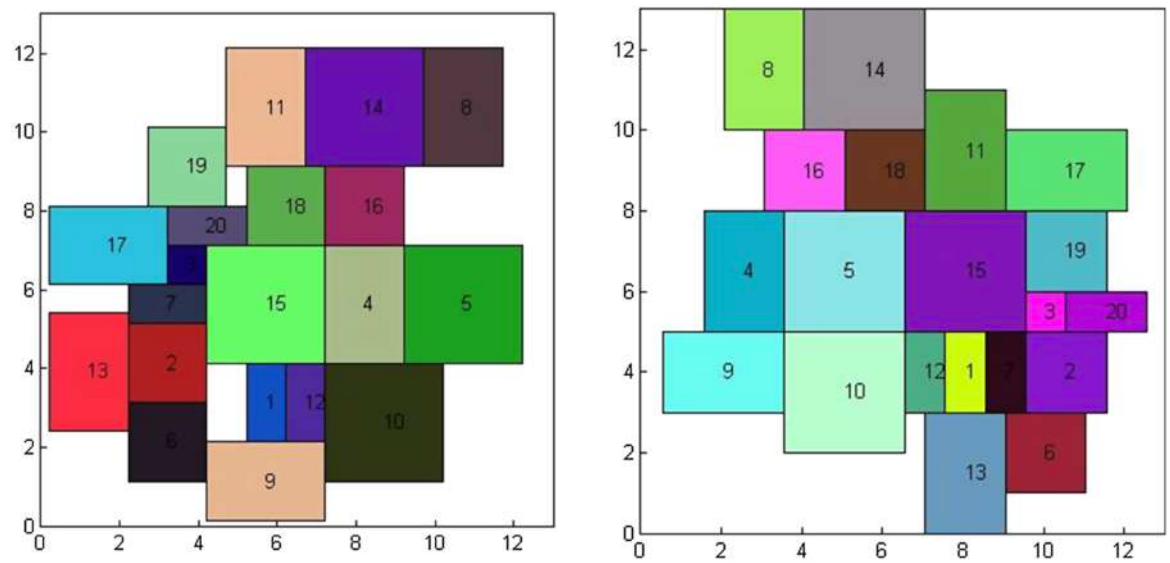


Figure 5.15. (a) An optimal solution of problem 3. (b) An alternate optimal solution of problem 3.

Table 5.8.

Solution of problem 3

Department	1	2	3	4	5	6	7	8	9	10
x_c	5.7090	3.2091	3.7091	8.2091	10.7091	3.2091	3.2091	10.7091	5.7091	8.7091
y_c	3.1338	4.1338	6.6338	5.6338	5.6338	2.1338	5.6338	10.6338	1.1338	2.6338
Rotation	0	0	0	0	0	0	0	0	0	0
Department	11	12	13	14	15	16	17	18	19	20
x_c	5.7091	6.7090	1.2091	8.2091	5.7091	8.2091	1.7091	6.2091	3.7091	4.2091
y_c	10.6338	3.1338	3.9154	10.6338	5.6338	8.1338	7.1338	8.1338	9.1338	7.6338
Rotation	0	0	1	0	0	0	0	0	0	0

5.4. General Discussion about the Proposed Method

As presented above, the efficiency of the model has been evaluated using three example problems. The model is compared mainly with the solutions obtained by Asl & Wong, 2015 as the solutions obtained by them are the best solution reported in the literature. Apart from Asl & Wong, 2015, the solutions are also compared with the results obtained by Mir and Imam, 2001; Imam and Mir, 1993; and Imam and Mir, 1989. Imam and Mir, 1989 used an analytical based search approach for optimizing topology of rectangular functional blocks of different sizes and aspect ratios. Imam and Mir modified their algorithm in 1993 and proposed a heuristic method for solving the functional blocks layout optimization problem [IMA93]. They also presented a modified simulated annealing method [MIR01]. Table 5.9 shows the comparison of the proposed model with the other methods. It may be observed from the table that the proposed model is significantly better than the other methods. In the case of problem 1 and problem 2, the proposed model achieved an improvement of 1.43% and 2.58% over the results obtained by Asl & Wong, 2015. As mentioned earlier, the problem 3 is a difficult problem due to the involvement of large numbers of departments. But for this problem also the proposed model has produced significantly improved solution over the other methods. The improvements achieved by the proposed model over Asl & Wong, 2015; Mir and Imam, 2001; Imam and Mir, 1993; and Imam and Mir, 1989 are 3.00%, 4.60%, 7.98%, and 12.74% respectively. This is quite encouraging and shows that the proposed model has the capability to handle large size layout problem efficiently and this work was also been presented and published by the author [HAS16a; HAS16b].

Overall, it can be concluded that the performance of the proposed population based local search model is quite good and can be applied to solve the large functional blocks layout optimization problem. The major achievement of this algorithm is its unique technique to handle the rotation of blocks. In general, rotation (90^o) of a block is handled by using a binary variable, *i.e.* 0 and 1. The use of binary variable converts the problem to a mixed integer problem that necessitates a specialized algorithm to solve the problem. Further, the use of rotation variable will increase the number of variables of the optimization problem by one third. For example, consider the 20 blocks example problem. The decision variables are the location of the centre of the blocks. Thus, it has 40 decision variables. The inclusion of rotation variable will increase the number of variables to 60. As such in the proposed method, the rotation is considered in a different way by maintaining an array of the dimension of the blocks; the number of decision variables is therefore only 40 for the third problems. Thus, this method reduces the number of variables by one third.

Classical gradient-based optimization algorithms are very efficient for local search solution. However, they are not capable for finding a solution when the simple interchange of the position of two blocks or rotation of a block or partial combination of two solutions gives better solution. These aspects have been handled efficiently using the exchange, interchange and rotation operators here. This is a population-based algorithm. But the experiments show that very less size of the population is just sufficient to find the global optimal solution. We have considered population size of 6 for problem 1. For problem 2 and 3, we have taken population size of 8 and 10 respectively.

Table 5.9.

Comparison of the solutions obtained by our proposed method with other method.

Methods	Problem 1				Problem 2				Problem 3			
	Best	Mean	Worst	Imp. (%)	Best	Mean	Worst	Imp. (%)	Best	Mean	Worst	Imp. (%)
Proposed method	191.00	192.53	199.52	-	1253.70	1293.19	1327.80	-	1171.50	1219.48	1253.6	-
Asl & Wong (2015)	193.74	208.74	-	1.43	1286.10	1335.63	-	2.58	1206.64	1264.21	-	3.00
Mir and Imam (2001)	-	-	-	-	-	-	-	-	1225.40	1287.29	-	4.60
Imam and Mir (1993)	-	-	-	-	-	-	-	-	1264.94	1333.81	-	7.98
Imam and Mir (1989)	-	-	-	-	-	-	-	-	1320.72	1395.64	-	12.74

5.5. Conclusion

The time taken by this method is more as compare to the other algorithm due to local search. A new technique should be developed to minimise the evaluation time. This paper proposes a new hybrid optimization technique for solving the facility layout optimization problem. Application of the gradient based local search algorithm and evolutionary algorithms based hybrid method on some test problems shows that the proposed methodology is significantly better than the solution obtained by modified PSO method. The test results show that cost improvement up to 12% can be achieved by using the proposed method. Moreover, the problem under consideration is a mixed-integer problem if the rotation of the blocks is considered. We have proposed a rotation operator that has eventually converted the mixed-integer problem to a non-integer problem. This has also reduced the number of the variables of the optimization problem by one third. The local optimal solution of the resulted problem can be solved using the gradient-based local search techniques. Although this newly developed technique is only applied to static unequal area facility layout problem, it can also be applied to dynamic unequal area facility layout problem.

CHAPTER 6:

CONCLUDING REMARKS AND FUTURE WORK

6.1. Conclusions

In the presented thesis, the literature on facility layout and the approaches to solve the existing facility layout has been reviewed. Hence, providing a basic knowledge on the problem. From the literature, it can be concluded that the FLPs are still an active area of research. This fact has encouraged the author to work with FLPs where the emphasis was given on solving the UA-FLPs. The problem has been solved using various approaches and is described in this thesis. After realizing the state of the art for various problems, we regroup the problems into cutting & packing and layout problems; we summarize and arrange the techniques and the techniques that have been developed.

For this thesis, we are interested in finding the optimal arrangement of a given number of non-overlapping unequal departments within a facility. The main goal was to list and categorize all the problems related to placement problems and to analyze the existing constructive and iterative approaches. From the analysis we intend to propose an algorithms capable of dealing with placement problem giving the best possible layout while satisfying all constraints.

In the first chapter, an overview of the thesis and the problem formulation and solving the UA-FLPs has been highlighted. Also, the description and introduction of the facility layout problem with its significance have been described. For solving the problem, the importance of constructive and iterative approach for solving the problem has been indicated.

The second chapter provides a basic knowledge about FLPs that is essential for the research. For this, the already published publication on FLPs by the researchers has been analyzed taking into account the previous characteristics and the resolution approaches. From this it was concluded that FLPs are still an active an open area of research.

The third chapter presented the basics of optimization; theory of genetic algorithm and various techniques for solving the LPs. It was concluded that there were fair amounts of work containing the optimization techniques with combine effort of local and global search method to solve the facility layout problem. The techniques used in the proposed algorithm are also presented in this chapter. Which is the concept of constructive Bottom-left fill approach [JAK96] in packing problem that has not been applied to the layout problems. Finally the critical review and the problem definition were concluded after going through all the literature.

The fourth chapter presented a hybrid approach to solve the FLPs. The objective was formulated as a combinatorial optimization problem where the material handling cost between the departments of the facility has to be minimized. The proposed modeling is not limited to the FLPs, but applies to all rectangular area packing and layout problems that can be modeled as sequential placement, where the components / compartments are positioned one after the other. However, this model cannot be applied to all layout problems as to achieve compactness all the components have to stick together to each other, which led us to develop a general resolution method to provide a generic solution approach.

In the fifth chapter, a generic method was proposed for solving FLPs. We have chosen to develop a general resolution method for its flexibility to adapt to the various problems encountered. The proposed method is a hybridization of an evolutionary algorithm with a local search algorithm. A new population based algorithm was chosen to efficiently explore the search space where the local search algorithm was used as an operator and for evaluation purpose. Unlike genetic algorithm where tournament operator handles the constraints [DEB00], in the proposed methodology the local search operator handles the constraints. To do this, it performs the minimization of the continuous objective function and also making the solution feasible that disobey the placement constraints.

6.2. Future Work

The research proves successful in terms of the results achieved, but it has opened a new argument for insight into the future line of work and promising new interesting results. Thus, these lines are:

– At present the approaches have been applied to single objective but the performance can be improved by changing the algorithm into a multi-objective for two or more objectives.

– At present in the proposed constructive approach, the standard genetic operators manage modifications of a solution. The solutions are changed without knowing whether they will improve them. The process can be improved by the use of intelligent genetic operators. Cagan et al., 1998 [CAG98] have added some selected modifications that may have the greatest influence on the objective functions. It can be considered to develop such an approach, so that the proposed modification of solutions makes maximum use of the knowledge of the problem that one may have;

– The proposed approaches have been applied to static to Facility Layout Problems (FLPs) but it can be applied to other such as, the Dynamic Facility Layout Problem or Multi-Floor Facility Layout Problem.

- The solutions can be further improved by the interaction by the user. The interactions between the designer and the optimization algorithms can be of two types. The first type concerns the human evaluation of objectives and second type concerns the modification of solutions by the designer. With described method, the optimization variables being directly the component positioning variables, the designer can easily interact with the generated solutions and propose solutions that seem promising. For this, the integration of an interactive optimization algorithm is necessary.
- At present the time taken by the hybrid local and global search approach is more. The time can be decreased by implementation of parallel genetic algorithm. The parallel processing on the randomly generated solution speeds up the whole search procedure. They also resulted in a more exhaustive search of the solution space in parallel, which rendered them as a powerful heuristic to solve NP-complete problems [GLO89b; SHA04].

REFERENCES

[AHM05] Ahmad, A. R. (2005). An intelligent expert system for decision analysis and support in multi-attribute layout optimization. *Ph.D. Thesis, University of Waterloo*, Canada, 2005.

[AIE06] Aiello, G., Enea, M., & Galante, G. (2006). A multi-objective approach to facility layout problem by genetic search algorithm and Electre method. *Robotics and Computer-Integrated Manufacturing*, 22(5), 447-455.

[AIE12] Aiello, G., La Scalia, G., & Enea, M. (2012). A multi objective genetic algorithm for the facility layout problem based upon slicing structure encoding. *Expert Systems with Applications*, 39(12), 10352-10358.

[ALB77] Albano, A. (1977). A method to improve two-dimensional layout. *Computer-Aided Design*, 9(1), 48-52.

[ARA01] Arapoglu, R. A., Norman, B. A., & Smith, A. E. (2001). Locating input and output points in facilities design-a comparison of constructive, evolutionary, and exact methods. *IEEE Transactions on Evolutionary Computation*, 5(3), 192-203.

[ARM63] Armour, G. C., & Buffa, E. S. (1963). A heuristic algorithm and simulation approach to relative location of facilities. *Management Science*, 9(2), 294-309.

[ASL15] Asl, A. D., & Wong, K. Y. (2015). Solving unequal-area static and dynamic facility layout problems using modified particle swarm optimization. *Journal of Intelligent Manufacturing*, 1-20.

[ASL16] Asl, A. D., Wong, K. Y., & Tiwari, M. K. (2016). Unequal-area stochastic facility layout problems: solutions using improved covariance matrix adaptation evolution strategy, particle swarm optimisation, and genetic algorithm. *International Journal of Production Research*, *54*(3), 799-823.

[AZA00] Azadivar, F., & Wang, J. (2000). Facility layout optimization using simulation and genetic algorithms. *International Journal of Production Research*, 38(17), 4369-4383.

[BAZ75] Bazaraa, M. S. (1975). Computerized layout design: a branch and bound approach. *AIIE Transactions*, 7(4), 432-438.

[BEK09] Bekrar, A., & Kacem, I. (2009). An exact method for the 2D guillotine strip packing problem. *Advances in Operations Research*, 2009.

[BÉN10] Bénabes, J., Bennis, F., Poirson, E., & Ravaut, Y. (2010). Interactive optimization strategies for layout problems. *International Journal on Interactive Design and Manufacturing*, 4(3), 181-190.

[BER87] Berkey, J. O., & Wang, P. Y. (1987). Two-dimensional finite bin-packing algorithms. *Journal of the Operational Research Society*, 38(5), 423-429.

[BOY12] Boyar, J., Dósa, G., & Epstein, L. (2012). On the absolute approximation ratio for first fit and related results. *Discrete Applied Mathematics*, 160(13), 1914-1923.

[BOZ96] Bozer, Y. A., & Suk-Chul, R. (1996). A branch and bound method for solving the bidirectional circular layout problem. *Applied Mathematical Modelling*, 20(5), 342-351.

[BOZ97] Bozer, Y. A., & Meller, R. D. (1997). A reexamination of the distance-based facility layout problem. *IIE transactions*, 29(7), 549-560.

[BOZ15] Bozorgi, N., Abedzadeh, M., & Zeinali, M. (2015). Tabu search heuristic for efficiency of dynamic facility layout problem. *The International Journal of Advanced Manufacturing Technology*, 77(1-4), 689-703.

[BRI06] Brintrup, A. M., Takagi, H., & Ramsden, J. (2006, April). Evaluation of sequential, multi-objective, and parallel interactive genetic algorithms for multi-objective floor plan optimisation. *In Workshops on Applications of Evolutionary Computation* (pp. 586-598). Springer Berlin Heidelberg.

[BUK14] 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*, *52*(24), 7339-7359.

[BUR04] Burke, E. K., Kendall, G., & Whitwell, G. (2004). A new placement heuristic for the orthogonal stock-cutting problem. *Operations Research*, 52(4), 655-671.

[BUS14] Buscher, U., Mayer, B., & Ehrig, T. (2014). A genetic algorithm for the unequal area facility layout problem. *In Operations Research Proceedings 2012*, 109-114. Springer International Publishing.

[BYR00] Byrd, R. H., Gilbert, J. C., & Nocedal, J. (2000). A trust region method based on interior point techniques for nonlinear programming. *Mathematical Programming*, *89*(1), 149-185.

[CAG98] Cagan, J., Degentesh, D., & Yin, S. (1998). A simulated annealing-based algorithm using hierarchical models for general three-dimensional component layout. *Computer-Aided Design*, 30(10), 781-790.

[CHA83] Chazelle, B. (1983). The bottom-left bin-packing heuristic: An efficient implementation. *IEEE Transactions on Computers*, 32(8), 697-707.

[CHE07] Chen, M., & Huang, W. (2007). A two-level search algorithm for 2D rectangular packing problem. *Computers & Industrial Engineering*, 53(1), 123-136.

[CHE13] Chen, G. Y. H. (2013). A new data structure of solution representation in hybrid ant colony optimization for large dynamic facility layout problems. *International Journal of Production Economics*, 142(2), 362-371.

[CHU82] Chung, F. R., Garey, M. R., & Johnson, D. S. (1982). On packing two-dimensional bins. *SIAM Journal on Algebraic Discrete Methods*, 3(1), 66-76.

[COR99] Corne, D., Dorigo, M., Glover, F., Dasgupta, D., Moscato, P., Poli, R., & Price, K. V. (1999). New ideas in optimization. *McGraw-Hill Ltd.,* UK.

[DAR59] Darwin, C. (1859). On the Origin of Species by Means of Natural Selection, or Preservation of Favoured Races in the Struggle for Life: Murray. *Evolution*.

[DAT11] Datta, D., Amaral, A. R., & Figueira, J. R. (2011). Single row facility layout problem using a permutation-based genetic algorithm. *European Journal of Operational Research*, 213(2), 388-394.

[DEB00] Deb, K. (2000). An efficient constraint handling method for genetic algorithms. *Computer methods in applied mechanics and engineering*, *186*(2), 311-338.

[DEB01] Deb, K. (2001). Nonlinear goal programming using multi-objective genetic algorithms. *Journal of the Operational Research Society*, 52(3), 291-302.

[DEB04] Deb, K., Jain, P., Gupta, N. K., & Maji, H. K. (2004). Multiobjective placement of electronic components using evolutionary algorithms. *IEEE Transactions on Components and Packaging Technologies*, 27(3), 480-492.

[DOR99] Dorigo, M., & Di Caro, G. (1999). Ant colony optimization: a new meta-heuristic. *IEEE Proceedings of the 1999 Congress on Evolutionary Computation, 1999*, 2, 1470-1477.

[DÓS13] Dósa, G., & Sgall, J. (2013). First fit bin packing: A tight analysis. *In 30th Symposium on Theoretical Aspects of Computer Science*, 20, 538–549.

[DÓS14] Dósa, G., & Sgall, J. (2014, July). Optimal analysis of best fit bin packing. *In 41st International Colloquium on Automata, Languages and Programming*, 429–441, 2014.

[DOW92] Dowsland, K. A., & Dowsland, W. B. (1992). Packing problems. *European Journal of Operational Research*, 56(1), 2-14.

[DOW02] Dowsland, K. A., Vaid, S., & Dowsland, W. B. (2002). An algorithm for polygon placement using a bottom-left strategy. *European Journal of Operational Research*, 141(2), 371-381.

[DRE04] Drezner, Z., Hahn, P. M., & Taillard, É. D. A study of quadratic assignment problem instances that are difficult for meta-heuristic methods. *Annals of Operations Research, special issue devoted to the State-of-the-Art in Integer Programming* (edited by: Guignard-Spielberg, M. and Spielberg, K.).

[DRI07] Drira, A., Pierreval, H., & Hajri-Gabouj, S. (2007). Facility layout problems: A survey. *Annual Reviews in Control*, 31(2), 255.

[DYC81] Dyckhoff, H. (1990). A typology of cutting and packing problems. *European Journal of Operational Research*, *44*(2), 145-159.

[DYC90] Dyckhoff, H. (1990). A typology of cutting and packing problems. *European Journal of Operational Research*, *44*(2), 145-159.

[EBE95] Eberhart, R., & Kennedy, J. (1995, October). A new optimizer using particle swarm theory. *Proceedings of the Sixth International Symposium on Micro Machine and Human Science, (Nagyo Japan),* IEEE Service Center, Piscataway, NJ, 39-43.

[EGE03] Egeblad, J. (2003, July). Placement techniques for VLSI layout using sequence-pair legalization. *Memoire de DEA, Department of Computer Science, University of Copenhagen*, Denmark, 2003.

[ELS77] Elshafei, A. N. (1977). Hospital layout as a quadratic assignment problem. *Journal of the Operational Research Society*, 28(1), 167-179.

[GAR79] Gardner, M. (1979). Some packing problems that cannot be solved by sitting on the suitcase. *Scientific American*, *241*(4), 22-26.

[GAR81] Garey, M. R., & Johnson, D. S. (1981). Approximation algorithms for bin packing problems: A survey. In *Analysis and design of algorithms in combinatorial optimization* (pp. 147-172). Springer Vienna.

[GAR13] García-Hernández, L., Pierreval, H., Salas-Morera, L., & Arauzo-Azofra, A. (2013). Handling qualitative aspects in unequal area facility layout problem: An interactive genetic algorithm. *Applied Soft Computing*, 13(4), 1718-1727.

[GAR15] García-Hernández, L., Palomo-Romero, J. M., Salas-Morera, L., Arauzo-Azofra, A., & Pierreval, H. (2015). A novel hybrid evolutionary approach for capturing decision maker knowledge into the unequal area facility layout problem. *Expert Systems with Applications*, *42*(10), 4697-4708.

[GIL61] Gilmore, P. C., & Gomory, R. E. (1961). A linear programming approach to the cutting-stock problem. *Operations Research*, 9(6), 849-859.

[GIL62] Gilmore, P. C. (1962). Optimal and suboptimal algorithms for the quadratic assignment problem. *Journal of the Society for Industrial and Applied Mathematics*, 10(2), 305-313.

[GIL63] Gilmore, P. C., & Gomory, R. E. (1963). A linear programming approach to the cutting stock problem—Part II. *Operations research*, 11(6), 863-888.

[GIL65] Gilmore, P. C., & Gomory, R. E. (1965). Multistage cutting stock problems of two and more dimensions. *Operations research*, 13(1), 94-120.

[GLO89A] Glover, F. (1989). Tabu search—part I. ORSA Journal on Computing, 1(3), 190-206.

[GLO89B] Glover, F., & Greenberg, H. J. (1989). New approaches for heuristic search: A bilateral linkage with artificial intelligence. European Journal of Operational Research, 39(2), 119-130.

[GLO97] Glover, F., & Laguna, M. (1997). General purpose heuristics for integer programming—Part I. Journal of Heuristics, 2(4), 343-358.

[GOL89] Goldberg, D., Deb, K., & Korb, B. (1989). Messy genetic algorithms: Motivation, analysis, and first results. *Complex systems*, (3), 493-530.

[GON15] Gonçalves, J. F., & Resende, M. G. (2015). A biased random-key genetic algorithm for the unequal area facility layout problem. European Journal of Operational Research, 246(1), 86-107.

[GRI04] Grignon, P. M., & Fadel, G. M. (2004). A GA based configuration design optimization method. *Transacti*ons-American Society of Mechanical Engineers Journal of Mechanical Design, 126(1), 6-15.

[GUO99] Guo, P. N., Cheng, C. K., & Yoshimura, T. (1999, June). An O-tree representation of non-slicing floorplan and its applications. In Proceedings of the 36th annual ACM/IEEE Design Automation Conference (pp. 268-273). ACM.

[HAS86] Hassan, M. M., Hogg, G. L., & Smith, D. R. (1986). SHAPE: a construction algorithm for area placement evaluation. International Journal of Production Research, 24(5), 1283-1295.

[HAS87] Hassan, M. M., & Hogg, G. L. (1987). A review of graph theory application to the facilities layout problem. Omega, 15(4), 291-300.

[HAS16a] Hasda, R. K., Bhattacharjya, R. K., & Bennis, F. (2016). Evolutionary and Local Search Based Hybrid Optimization Model to Solve Facility Layout Problem for Unequal Compartment, 5th International Conference on Engineering Optimization - Iguassu Falls, Brazil, 19-23 June 2016.

[HAS16b] Hasda, R. K., Bhattacharjya, R. K., & Bennis, F. (2016). Modified genetic algorithms for solving facility layout problems. *International Journal on Interactive Design and Manufacturing (IJIDeM)*, 1-13.

[HE12] He, Z., Yang, Y., Niu, W., Wang, Y., Zhang, H., Liu, Y., & Wu, Z. (2012, October). Component layout optimization of thermal engine used by underwater gliders. *In Oceans*, 2012 (pp. 1-7). IEEE.

[HEA99] Healy, P., Creavin, M., & Kuusik, A. (1999). An optimal algorithm for rectangle placement. *Operations Research Letters*, 24(1), 73-80.

[HER91] Heragu, S. S., & Kusiak, A. (1991). Efficient models for the facility layout problem. *European Journal of Operational Research*, 53(1), 1-13.

[HER97] Heragu, S. S. (1997). Facilities design. Boston: BWS.

[HIF03] Hifi, M., & M'Hallah, R. (2003). A hybrid algorithm for the two-dimensional layout problem: the cases of regular and irregular shapes. *International Transactions in Operational Research*, 10(3), 195-216.

[HOL75] Holland, J. H. (1975). Adaptation in natural and artificial systems. An introductory analysis with application to biology, control, and artificial intelligence. *Ann Arbor, MI: University of Michigan Press*.

[HOL92] Holland, J. H. (1992). Genetic algorithms. *Scientific american*, *267*(1), 66-73.

[HOP99] Hopper, E., & Turton, B. C. H. (1999). A genetic algorithm for a 2D industrial packing problem. *Computers & Industrial Engineering*, 37(1-2), 375-378.

[HOP01] Hopper, E. & Turton, B. C. H. (2001). An empirical investigation of meta-heuristic and heuristic algorithms for a 2D packing problem. *European Journal of Operational Research*, 128(1), 34-57.

[HUA16] Huang, C., & Wong, C. K. (2016). Discretized Cell Modeling for Optimal Facility Layout Plans of Unequal and Irregular Facilities. *Journal of Construction Engineering and Management*, *143*(1), 04016082.

[IMA89] Imam, M. H., & Mir, M. (1989). Nonlinear programming approach to automated topology optimization. *Computer-Aided Design*, 21(2), 107-115.

[IMA93] Imam, M. H., & Mir, M. (1993). Automated layout of facilities of unequal areas. *Computers & industrial engineering*, *24*(3), 355-366.

[IMA08] Imamichi, T., Arahori, Y., Gim, J., Hong, S. H., & Nagamochi, H. (2008). *Removing overlaps in label layouts using multi-sphere scheme*. Tech. Rep. 2008-006, Department of Applied Mathematics and Physics, Kyoto University and School of Information Technologies, University of Sydney.

[IZA16] Izadinia, N., & Eshghi, K. (2016). A robust mathematical model and ACO solution for multi-floor discrete layout problem with uncertain locations and demands. *Computers & Industrial Engineering*, *96*, 237-248.

[JAA09] Jaafari, A. A., Krishnan, K. K., Doulabi, S. H. H., & Davoudpour, H. (2009, October 20-22). A multi-objective formulation for facility layout problem. *In World Congress on Engineering and Computer Science, (Vol. 2)*, San Francisco, USA.

[JAC09] Jacquenot, G., Bennis, F., Maisonneuve, J. J., & Wenger, P. (2009, November). 2d multi-objective placement algorithm for free-form components. In *Proceedings of the ASME International Design Engineering Technical Conferences & Computers and Information in Engineering Conference, San Diego, CA, Aug*.

[JAI98] Jain, S., & Gea, H. C. (1998). Two-dimensional packing problems using genetic algorithms. *Engineering with Computers*, 14(3), 206-213.

[JAK96] Jakobs, S. (1996). On genetic algorithms for the packing of polygons. *European journal of operational research*, *88*(1), 165-181.

[JAN10] Jannat, S., Khaled, A. A., & Paul, S. K. (2010, January). Optimal solution for multi-objective facility layout problem using genetic algorithm. *In Proceedings of the 2010 International Conference on Industrial Engineering and Operations Management*. Dhaka, Bangladesh.

[KEN97] Kennedy, J., & Eberhart, R. C. (1997, October). A discrete binary version of the particle swarm algorithm. IEEE International Conference on Systems, Man, and Cybernetics, 1997. Computational Cybernetics and Simulation., 1997 (Vol. 5, pp. 4104-4108).

[KIE14] Kierkosz, I., & Luczak, M. (2014). A hybrid evolutionary algorithm for the two-dimensional packing problem. *Central European Journal of Operations Research*, 22(4), 729-753.

[KIM05] Kim, J. G., & Goetschalckx, M. (2005, May). A mixed integer programming model for modifying a block layout to facilitate smooth material flows. *In International Conference on Computational Science and Its Applications* (pp. 322-332). Springer Berlin Heidelberg.

[KIR83] Kirkpatrick, S., Gelatt, C. D., & Vecchi, M. P. (1983). Optimization by simulated annealing. *Science*, 220(4598), 671-680.

[KNO00] Knowles, J. D., & Corne, D. W. (2000). Approximating the nondominated front using the Pareto archived evolution strategy. *Evolutionary computation*, *8*(2), 149-172.

[KOO57] Koopmans, T. C., & Beckmann, M. (1957). Assignment problems and the location of economic activities. *Econometrica: Journal of the Econometric Society*, 53-76.

[KOT13] Kothari, R., & Ghosh, D. (2013). Tabu search for the single row facility layout problem using exhaustive 2-opt and insertion neighborhoods. *European Journal of Operational Research*, 224(1), 93-100.

[KOU92] Kouvelis, P., Chiang, W. C., & Fitzsimmons, J. (1992). Simulated annealing for machine layout problems in the presence of zoning constraints. *European Journal of Operational Research*, *57*(2), 203-223.

[KUH55] Kuhn, H. W. (1955). The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2(1-2), 83-97.

[KUL07] Kulturel-Konak, S. (2007). Approaches to uncertainties in facility layout problems: Perspectives at the beginning of the 21st Century. *Journal of Intelligent Manufacturing*, *18*(2), 273.

[KUL13] Kulturel-Konak, S., & Konak, A. (2013). Linear programming based genetic algorithm for the unequal area facility layout problem. *International Journal of Production Research*, *51*(14), 4302-4324.

[KUL15] Kulturel-Konak, S., & Konak, A. (2015). A large-scale hybrid simulated annealing algorithm for cyclic facility layout problems. *Engineering Optimization*, *47*(7), 963-978.

[KUS87] Kusiak, A., & Heragu, S. S. (1987). The facility layout problem. *European Journal of Operational Research*, 29(3), 229-251.

[LAW63] Lawler, E. L. (1963). The quadratic assignment problem. *Management Science*, 9(4), 586-599.

[LEE05] Lee, K. Y., Roh, M. I., & Jeong, H. S. (2005). An improved genetic algorithm for multi-floor facility layout problems having inner structure walls and passages. *Computers & Operations Research*, 32(4), 879-899.

[LIG00] Liggett, R. S. (2000). Automated facilities layout: past, present and future. *Automation in Construction*, 9(2), 197-215.

[LIU99] Liu, D., & Teng, H. (1999). An improved BL-algorithm for genetic algorithm of the orthogonal packing of rectangles. *European Journal of Operational Research*, 112(2), 413-420.

[LIU07] Liu, Q., & Meller, R. D. (2007). A sequence-pair representation and MIP-model-based heuristic for the facility layout problem with rectangular departments. *IIE transactions*, *39*(4), 377-394.

[LIU08] Liu, Z. W., & Teng, H. F. (2008). Human–computer cooperative layout design method and its application. *Computers & Industrial Engineering*, 55(4), 735-757.

[LOD02a] Lodi, A., Martello, S., & Vigo, D. (2002). Recent advances on two-dimensional bin packing problems. *Discrete Applied Mathematics*, 123(1), 379-396.

[LOD02b] Lodi, A., Martello, S., & Vigo, D. (2002). Heuristic algorithms for the three-dimensional bin packing problem. *European Journal of Operational Research*, 141(2), 410-420.

[MAR00] Martello, S., Pisinger, D., & Vigo, D. (2000). The three-dimensional bin packing problem. *Operations Research*, 48(2), 256-267.

[MAR03] Martello, S., Monaci, M., & Vigo, D. (2003). An exact approach to the strip-packing problem. *INFORMS Journal on Computing*, 15(3), 310-319.

[MAT13a] Matai, R., Singh, S. P., & Mittal, M. L. (2013). A new heuristic approach for solving facility layout problem. *International Journal of Advanced Operations Management*, *5*(2), 137-158.

[MAT13b] Matai, R., Singh, S. P., & Mittal, M. L. (2013). A non-greedy systematic neighbourhood search heuristic for solving facility layout problem. *The International Journal of Advanced Manufacturing Technology*, *68*(5-8), 1665-1675.

[MAT15] Matai, R. (2015). Solving multi objective facility layout problem by modified simulated annealing. *Applied Mathematics and Computation*, *261*, 302-311.

[MAZ99] Mazumder, P., & Rudnick, E. M. (1999). Genetic algorithms for VLSI design, layout & test automation. Prentice Hall PTR.

[MCK06] McKendall, A. R., & Shang, J. (2006). Hybrid ant systems for the dynamic facility layout problem. *Computers & Operations Research*, 33(3), 790-803.

[MCK10] McKendall, A. R., & Hakobyan, A. (2010). Heuristics for the dynamic facility layout problem with unequal-area departments. *European Journal of Operational Research*, 201(1), 171-182.

[MIA08] Miao, Y., Fadel, G. M., & Gantovnik, V. B. (2008). Vehicle configuration design with a packing genetic algorithm. *International Journal of Heavy Vehicle Systems*, 15(2-4), 433-448.

[MIC02] Michalek, J., & Papalambros, P. (2002). Interactive design optimization of architectural layouts. *Engineering Optimization*, 34(5), 485-501.

[MIE00] Miettinen, K., & Mäkelä, M. M. (2000). Interactive multiobjective optimization system www-nimbus on the Internet. *Computers & Operations Research*, 27(7), 709-723.

[MIR01] Mir, M., & Imam, M. H. (2001). A hybrid optimization approach for layout design of unequal-area facilities. *Computers & Industrial Engineering*, 39(1), 49-63.

[MON91] Montreuil, B. (1991). A modelling framework for integrating layout design and flow network design. *In Material Handling'90 (pp. 95-115)*. Springer Berlin Heidelberg.

[MUR95] Murata, H., Fujiyoshi, K., Nakatake, S., & Kajitani, Y. (1996). VLSI module placement based on rectangle-packing by the sequence-pair. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 15(12), 1518-1524.

[MUR96] Murata, H., Fujiyoshi, K., Nakatake, S., & Kajitani, Y. (1996). VLSI module placement based on rectangle-packing by the sequence-pair. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 15(12), 1518-1524.

[NAS13] Hosseini-Nasab, H., & Emami, L. (2013). A hybrid particle swarm optimisation for dynamic facility layout problem. *International Journal of Production Research*, 51(14), 4325-4335.

[NEG14] Neghabi, H., Eshghi, K., & Salmani, M. H. (2014). A new model for robust facility layout problem. *Information Sciences*, *278*, 498-509.

[NEG15] Neghabi, H., & Ghassemi Tari, F. (2015). An optimal approach for maximizing the number of adjacencies in multi floor layout problem. *International Journal of Production Research*, *53*(11), 3462-3474.

[OHM10] Ohmori, S., Yoshimoto, K., & Ogawa, K. (2010, May). Solving facility layout problem via particle swarm optimization. *IEEE Third International Joint Conference on Computational Science and Optimization*, (Vol. 1, pp. 409-413).

[ONO99] Ono, T., & Ikeda, T. (1999, July). Optimizing two-dimensional guillotine cut by genetic algorithms. *In Proc. of the Ninth AJOU-FIT-NUST Joint Seminar* (pp. 40-47).

[PAE17] Paes, F. G., Pessoa, A. A., & Vidal, T. (2017). A hybrid genetic algorithm with decomposition phases for the unequal area facility layout problem. *European Journal of Operational Research*, *256*(3), 742-756.

[PIS07] Pisinger, D. (2007). Denser packings obtained in O (n log log n) time. *INFORMS Journal on Computing*, 19(3), 395-405.

[POU14] Pourvaziri, H., & Naderi, B. (2014). A hybrid multi-population genetic algorithm for the dynamic facility layout problem. *Applied Soft Computing*, *24*, 457-469.

[PRI06] Price, K., Storn, R. M., & Lampinen, J. A. (2006). *Differential evolution: a practical approach to global optimization*. Springer Science & Business Media.

[REC65] Rechenberg, I. (1965). Cybernetic solution path of an experimental Problem",(Royal aircraft establishment translation no. 1122, BF toms, trans.). *Farnsborough Hants: Ministery of Aviation, Royal Aircraft Establishment*, *1122*.

[RIP10] Ripon, K. S. N., Glette, K., Høvin, M., & Torresen, J. (2010, November). A genetic algorithm to find pareto-optimal solutions for the dynamic facility layout problem with multiple objectives. *In International Conference on Neural Information Processing* (pp. 642-651). Springer Berlin Heidelberg.

[ROU87] Roucairol, C. (1987). A parallel branch and bound algorithm for the quadratic assignment problem. *Discrete Applied Mathematics*, *18*(2), 211-225.

[SAC98] Sachdev, S., Paredis, C. J., Gupta, S. K., & Talukdar, S. N. (1998, September). '3D Spatial Layouts Using A-Teams. *In Proceedings of ASME 24th Design Automation Conference* (pp. 1-11). Atlanta, Georgia, USA.

[SAL15] Salmani, M. H., Eshghi, K., & Neghabi, H. (2015). A bi-objective MIP model for facility layout problem in uncertain environment. *The International Journal of Advanced Manufacturing Technology*, *81*(9-12), 1563-1575.

[SAN13] Sangchooli, A., & Akbari Jokar, M. R. (2013). Obtaining an Initial Solution for Facility Layout Problem. *Journal of Industrial Mathematics*, 1-10.

[SAR15] Saraswat, A., Venkatadri, U., & Castillo, I. (2015). A framework for multi-objective facility layout design. *Computers & Industrial Engineering*, *90*, 167-176.

[SEP70] Seppänen, J., & Moore, J. M. (1970). Facilities planning with graph theory. *Management Science*, 17(4), B-242.

[SHA04] Shayan, E., & Chittilappilly, A. (2004). Genetic algorithm for facilities layout problems based on slicing tree structure. International Journal of Production Research, 42(19), 4055-4067.

[SIN10] Singh, S. P., & Singh, V. K. (2010). An improved heuristic approach for multi-objective facility layout problem. *International Journal of Production Research*, *48*(4), 1171-1194.

[SOL08] Solimanpur, M., & Jafari, A. (2008). Optimal solution for the two-dimensional facility layout problem using a branch-and-bound algorithm. *Computers & Industrial Engineering*, 55(3), 606-619.

[SZY95] Szykman S and Cagan J. (1995). A simulated annealing-based approach to three-dimensional component packing. *Journal of Mechanical Design*, 117(2A), 308-314.

[SZY97] Szykman S. and Cagan J. (1997). Constrained Three-Dimensional Component Layout Using Simulated Annealing. *Journal of Mechanical Design*, 119 (1), 28 -35.

[TAP01] Tappeta, R. V., & Renaud, J. E. (2001). Interactive multiobjective optimization design strategy for decision based design. *Transactions-American Society of Mechanical Engineers Journal of Mechanical Design*, 123(2), 205-215.

[TAR15] Tari, F. G., & Neghabi, H. (2015). A new linear adjacency approach for facility layout problem with unequal area departments. *Journal of Manufacturing Systems*, *37*, 93-103.

[TAS15] Tasadduq, I. A., Imam, M. H., & Ahmad, A. (2015). A hybrid algorithm for optimising facility layout. *South African Journal of Industrial Engineering*, 26(1), 120-134.

[TEO08] Teo, Y. T., & Ponnambalam, S. G. (2008, August). A hybrid ACO/PSO heuristic to solve single row layout problem. *IEEE International Conference on Automation Science and Engineering*. (pp. 597-602).

[TOM10] Tompkins, J. A., White, J. A., Bozer, Y. A., & Tanchoco, J. M. A. (2010). Facilities planning. John Wiley & Sons.

[WAN15] Wang, S., Zuo, X., Liu, X., Zhao, X., & Li, J. (2015). Solving dynamic double row layout problem via combining simulated annealing and mathematical programming. *Applied Soft Computing*, 37, 303-310.

[XIA10] Xia, B., & Tan, Z. (2010). Tighter bounds of the First Fit algorithm for the bin-packing problem. *Discrete Applied Mathematics*, 158(15), 1668-1675.

[XIA16] Xiao, Y. J., Zheng, Y., Zhang, L. M., & Kuo, Y. H. (2016). A combined zone-LP and simulated annealing algorithm for unequal-area facility layout problem. *Advances in Production Engineering & Management*, *11*(4), 259.

[XIE16] Xie, Y., Zhou, S., Xiao, Y., & Chang, W. (2016, December). A new area linearization method for Unequal area facility layout problem. In *Industrial Engineering and Engineering Management (IEEM), 2016 IEEE International Conference on* (pp. 1289-1293). IEEE.

[YIN00] Yin, S. and Cagan, J. (2000). An extended pattern search algorithm for three-dimensional component layout. BOOK REVIEW, 102.

[YOU03] Youssef, H., Sait, S. M., & Ali, H. (2003). Fuzzy simulated evolution algorithm for VLSI cell placement. *Computers & Industrial Engineering*, 44(2), 227-247.

[ZHA14a] Zhao, F., Li, G., Hu, H., Du, J., Guo, C., & Li, T. (2014, June). A novel improved hybrid particle swarm optimisation based genetic algorithm for the solution to layout problems. *IEEE 11th World Congress on Intelligent Control and Automation*, (pp. 5041-5046).

[ZHA14b] Zhang, Q., Jiang, C., Zhang, J., & Wei, Y. (2014). Application of genetic algorithm in functional area layout of railway logistics park. *Procedia-Social and Behavioral Sciences*, *138*, 269-278.

[ZUO14] Zuo, X., Murray, C. C., & Smith, A. E. (2014). Solving an extended double row layout problem using multiobjective tabu search and linear programming. *IEEE Transactions on Automation Science and Engineering*, 11(4), 1122-1132.

[ZHU12] Zhu, J., Zhang, W., Xia, L., Zhang, Q., & Bassir, D. (2012). Optimal packing configuration design with finite-circle method. *Journal of Intelligent & Robotic Systems*, 67(3), 185-199.

UNIVERSITE BRETAGNE LOIRE

CENTRALE NANTES

Thèse de Doctorat

Ranjan Kumar HASDA

Titre de la thèse: Contribution à l'optimisation de Problème d'Agencement d'Espaces Rectangulaires Inégales

Title of thesis: Contribution to the optimization of Unequal Area Rectangular Facility Layout Problem

Résumé

L'agencement d'espace est un problème courant dans la plupart des secteurs industriels. Ce problème est de nature continue et discret et il est considéré comme un problème NP-difficile. Les méthodes d'optimisation traditionnelles, plus appropriées pour une recherche locale sont difficilement utilisables aux problèmes d'agencement. Afin de contourner ces limitations inhérentes aux méthodes classiques, nous proposons deux algorithmes adaptés aux problèmes d'agencement statique de composants de différentes tailles. Pour les problèmes d'agencement considérés, les fonctions objectives à minimiser sont non linéaires et représentent les coûts associés aux sommes pondérées des distances entre les composants.

La première approche que nous considérons est une méthode hybride en deux étapes. La première étape consiste à construire un agencement en se basant sur la méthode dite "bas-gauche" comme une solution locale. Ensuite, la solution obtenue est améliorée en appliquant un algorithme génétique modifié. Les opérateurs de croisement et de mutation sont alors adaptés pour prendre en compte les spécificités du problème d'agencement.

La deuxième approche est une combinaison entre une recherche locale et globale. Dans ce cas, l'algorithme génétique est également modifié par l'introduction d'un opérateur spécialisé pour le traitement des rotations des composants. Il permet notamment d'éviter le couplage entre les variables réelles et entières et permet également de réduire considérablement le nombre de variables du problème d'optimisation.

Les performances des deux approches sont testées et comparées avec les exemples de référence extraits des publications traitant du problème d'optimisation d'agencement. Nous démontrons que les deux approches que nous proposons obtiennent de meilleures performances que les approches existantes.

Mots clés
Agencement d'Espace, Optimisation, Algorithme Génétique, Approche Hybride : Construction-Amélioration.

Abstract

A facility layout design is one of the most commonly faced problems in the manufacturing sectors. The problem is mixed-integer in nature and usually an NP-hard problem, which makes it difficult to solve using classical optimization techniques, which are better for local search. To overcome these limitations, two algorithms have been proposed for solving static facility layout problems with the unequal size compartments. The objective function of the problems considered is nonlinear in which the sum of the material handling cost has been minimized.

In the first approach, a hybrid constructive and improvement model has been proposed where an advanced bottom-left fill technique was used as constructive approach. The constructive model proposed also acts as a local search method based on greedy algorithm. For improvement approach a hybrid genetic algorithm has been proposed, where the crossover and mutation operator are specially designed to handle the solution representation which itself is used as constructive model.

In the second approach, a combined local and global search model was proposed where a rotation operator was used to avoid mixed-integer formulation of the problem. Use of rotation operator has also reduced the number of variables significantly. Apart from the conventional evolutionary operators this model has also used exchange and rotation operators.

The performances of both model are tested over a previously solved problem selected from the literature. The evaluation of the results shows that the performances of the proposed models are better than many existing algorithms and has the potential for field applications.

Key Words
Facility Layout, Optimization, Genetic Algorithm, Hybrid Construction and Improvement Approach