Development of a Multi-Objective Optimization Model for Unequal Area Facility Layout Problem

TANVEER HOSSAIN BHUIYAN

DEPARTMENT OF INDUSTRIAL AND PRODUCTION ENGINEERING
BANGLADESH UNIVERSITY OF ENGINEERING & TECHNOLOGY
DHAKA-1000, BANGLADESH

May, 2014

Development of a Multi-Objective Optimization Model for Unequal Area Facility Layout Problem

BY
TANVEER HOSSAIN BHUIYAN

A thesis submitted to the Department of Industrial & Production Engineering, Bangladesh University of Engineering & technology, in partial fulfillment of the requirements for the degree of Master of Science in Industrial & Production Engineering

DEPARTMENT OF INDUSTRIAL AND PRODUCTION ENGINEERING
BANGLADESH UNIVERSITY OF ENGINEERING & TECHNOLOGY
DHAKA-1000, BANGLADESH

May, 2014

CERTIFICATE OF APPROVAL

The thesis titled "**Development of a Multi-Objective Optimization Model for Unequal Area Facility Layout Problem"** submitted by Tanveer Hossain Bhuiyan**,** Student no: 0412082018 P has been accepted as satisfactory in partial fulfillment of the requirements for the degree of Master of Science in Industrial & Production Engineering on May 25, 2014.

BOARD OF EXAMINERS

____

Dr. Abdullahil Azeem — Chairman (Supervisor)
Professor
Department of IPE, BUET, Dhaka.

____

Head — Member (Ex-Officio)
Department of IPE, BUET, Dhaka.

____

Dr. Nafis Ahmad — Member
Professor
Department of IPE, BUET, Dhaka.

____

Dr. Ferdous Sarwar — Member
Assistant Professor
Department of IPE, BUET, Dhaka.

____

Dr. Md. Ariful Islam — Member (External)
Professor
Department of IPE,
Shahjalal University of Science and Technology, Sylhet.

CANDIDATE'S DECLARATION

It is hereby declared that this thesis or any part of it has not been submitted elsewhere for the award of any degree or diploma.

Tanveer Hossain Bhuiyan

To the Almighty

To my family

ACKNOWLEDGEMENT

All credits go to Allah, the most benevolent and the Almighty, for his boundless grace in successful completion of this thesis.

At the very beginning the author expresses his sincere gratitude and profound indebtness to his thesis supervisor Dr. Abdullahil Azeem, Professor, Department of Industrial & Production Engineering, BUET, Dhaka-1000, under whose continuous supervision this thesis was carried out. His affectionate guidance, valuable suggestions and inspirations throughout this work made this study possible.

The author also expresses his gratitude to Sarah Harun, Software Engineer and Delivery Analyst, GTO, IMS Health Dhaka for her constant co-operation and assistance in developing the algorithm and computer coding.

The author is also indebted to Dr. Ferdous Sarwar, Assistant Professor, Department of IPE, BUET and Dr. Mohammad Farhad Uddin, Associate Professor, Department of Mathematics, BUET for their valuable instructions in performing this thesis work.

Finally, the author would like to thank all of his colleagues and friends for their co-operation and motivation to complete the thesis timely. And the author would also like to extend his thanks to his parents whose continuous inspiration, sacrifice and support encouraged him to complete the thesis successfully.

ABSTRACT

Facility layout problem is one of the most fundamental optimization problems encountered in today's world since layout affects the overall operating efficiency of an organization. From the beginning of the facility layout literature, it has been a common trend to assume that the departments of a facility are of equal areas. But in reality departments are mostly unequal areas. Again, the earlier facility layout models considered only material transportation cost in designing optimal facility layout. Again, few recent works considered closeness preferences between the departments along with material handling cost in unequal-area facility layout model development. But one most important factor for customer satisfaction is makespan of the layout which was not taken in consideration. In this thesis work, three most important criteria in designing optimal facility layout has been considered which are material transportation cost, closeness rating score and makespan of the total layout. This multi-objective facility layout model has been developed for an unequal area facility layout problem. The goal of this thesis work is to determine the optimal arrangement as well as the optimal dimension of the departments by minimizing the material transportation cost, minimizing the makespan of the overall system and maximizing the total closeness rating score simultaneously. The constraint equations have been developed for non-overlapping of the departments, bounding the departments within the facility, department area constraints and aspect ratio constraints. Two suitable optimization algorithms: spatial branch and bound algorithm (sBB) and slicing tree structure embedded random weighted genetic algorithm (RWGA) are employed to optimize the constrained multi-objective mixed-integer nonlinear model. Two numerical examples and their solutions are then provided to have a better understanding about the demonstration of the proposed model. Finally sensitivity analysis is also carried out to have better insights about the model developed. It is found that, solutions found from both methods are pareto optimal and the proposed RWGA performs better in providing wide range of trade-off solutions. This model helps in developing an efficient layout for a facility by making a trade-off among material handling cost, closeness rating score and makespan.

TABLE OF CONTENT

LIST OF FIGURES

LIST OF TABLES

NOMENCLATURE

C_{ij}	Cost of transporting material per unit distance between department I and j
R_{ij}	Closeness relationship value between department i and j
t_{ij}	Material handling time per unit distance between department i and j
x_i	X-location of the centroid of department i
y_i	Y-location of the centroid of department i
l_i	Length of department i
w_i	Width of department i
a_i	Area of department i
α_i	Aspect ratio of department i
d_{ij}	Distance between the centroids of department i and department j
p_{ij}	Binary variable representing the adjacency between the departments i and j along x-axis
q_{ij}	Binary variable representing the adjacency between the departments i and j along y-axis
z_{ij}	Binary variable controlling the adjacency in both direction simultaneously
L	Length of the plant floor
W	Width of the plant floor
l_{min}	Minimum side length for the departments
x_{ij}	Distance between departments i and j along y-axis
y_{ij}	Distance between departments i and j along y-axis
$left_{ij}$	Representing department i to the left of department j
$below_{ij}$	Representing department i to the below of department j
M	Large positive number
S_{ijm}	Relationship value between departments i and j for objective m
T_{ijm}	Normalized relationship value between departments i and j for objective m

ABBREVIATIONS

FLP	Facility layout problem
UA-FLP	Unequal area facility layout problem
QAP	Quadratic assignment problem
MIP	Mixed-Integer programming
MINLP	Mixed-Integer nonlinear programming
NLT	Nonlinear technique
VLSI	Very large scale integration
MPEC	Mathematical program with equilibrium constraints
MOOP	Multi-objective optimization problem
MOGA	Multi-objective genetic algorithm
RWGA	Random weighted genetic algorithm
SDP	Semidefinite programming
S-BB	Spatial Branch-and-Bound
GA	Genetic algorithm
MDS	Multidimensional scaling
RWGA	Random weight genetic algorithm
STS	Slicing tree structure
FBS	Flexible bay structure

CHAPTER II

LITERATURE REVIEW

The facility layout problems (FLPs) have been the subject of interest for many years. Even though facility layout problems have received considerable attention over the years, it was not until the emergence of the interest in operations research and management science that the subject received renewed attention in a number of disciplines. The facility layout problem is concerned with finding the most efficient arrangement of m indivisible departments within a facility. The output of a facility layout problem is a block layout which specifies the relative location of each department. The mathematical modeling of facility layout problem is complex, because there exists several criteria that must be taken into consideration when formulating and solving the model. In case of multi-objective facility layout model, there may be some criteria which are conflicting, perhaps non-commensurate. This imposes pressure upon the researchers to implement an appropriate and realistic mathematical modeling of a facility layout problem.

Some research papers have been studied to understand the background of the study. Many factors have been considered to develop the previous facility layout models. Different methodologies have been followed to develop and solve the models. In this section, some research papers have been studied to understand the factors and methodologies considered by researchers.

The facility layout problem has traditionally been formulated as a quadratic assignment problem (QAP) or using a graph oriented approach (e.g Kusiak and Heragu, 1987; Francis, McGinnis and White, 1992; Schockaert, Smart, and Twaroch, 2011). This formulation assigns n (equal-sized) facilities to n mutually exclusive sites (locations). The distance between various locations is measured by a rectilinear distance. Therefore, the QAP is a special case of the facility layout problem because it assumes all facilities have equal areas, the distance from one site to another can be predetermined, and that all locations are fixed and known a priori. Therefore, such QAP-type models, however, are not applicable for FLPs with unequal-sized departments (Bozer and Meller, 1997). In graph oriented approaches the dimensions of facilities are not incorporated during optimization.

They have to be considered afterwards when the layout is constructed according to the optimal graph (e.g Osman, 2006). The main drawback of these approaches is that geometric constraints, e.g. unequal sizes of facilities, cannot be considered sufficiently, though in reality, departments are mostly of unequal sized. Bazaraa (1975) stated a generalized quadratic assignment problem which incorporates facilities with unequal areas. A single facility may be represented by a given number of multiple blocks in this model. But this results in a large number of blocks and problems assuring connectivity and given shapes of the facilities.

Since the nineties, the research in this area focused mainly on the unequal area rectangular FLP. The concept of unequal-area facility layout problem (UA-FLP) was first introduced by Armour and Buffa (1963) but the model was not shape constrained, i.e. no minimum side length for each department was specified. Again, they determined only the relative location of departments within the facility by developing a computerized relative location of facilities technique (CRAFT). CRAFT begins by determining the centroid of each department in the initial layout. It then performs two way or three way exchanges of the centroids of nonfixed departments that are also equal in area or adjacent in the current layout. For each exchange, CRAFT will calculate an estimated cost reduction and it chooses the exchange with the largest estimated reduction. But, constraining the feasible department exchanges to those departments that are adjacent or equal in area is likely to affect the quality of the solution.

Later, van Camp et al. (1991) developed a nonlinear model of UA-FLP where a minimum side length was specified for each department. In this paper, a nonlinear programming (NLP) technique was developed in finding good solutions to the facility layout problem by minimizing the material handling cost. The developed nonlinear technique (NLT) transforms the constrained nonlinear model into an unconstrained form by an exterior point quadratic penalty function method. This method works by including a penalty in the objective function for every constraint that is violated and then minimizing the transformed objective function. The penalty is iteratively increased and the objective function minimized again until, in the limit, a feasible point is generated.

Since, the constraints of the model are nonconvex, the optimization procedure is guaranteed to find only a local minimum to the problem.

A mixed integer programming (MIP) model for FLP was introduced by Montreuil (1990) having linear area constraints and binary variables. The binary relative location decision variables were utilized to ensure that the departments do not overlap. This model is one of the first Mixed-Integer Programming facility layout (MIP-FLP) models for the continuous representation-based FLP. This model is commonly referred to as FLP1.

There are a variety of FLP representation methods, but most of them fall into two main categories: discrete representation and continuous representation. With a discrete representation, the facility is represented by an underlying grid structure with fixed dimensions and all departments are composed of an integer number of grids. By representing the FLP in a discrete fashion, the FLP is simplified, but at the penalty of eliminating many solutions from consideration. In a continuous representation, department dimensions are not restricted to an underlying grid structure, but rather, represented continuously. Continuous representation is more accurate and realistic than discrete representation, and thus, is capable of finding the "real optimal" final layout solution. However, this continuous representation also increases the complexity of the FLP (Liu and Meller, 2007).

Montreuil (1990) used a distance based objective which was material handling cost in developing the layout. In this model the nonlinear area constraint was replaced by a linear bounded perimeter constraint to avoid the nonlinearity in model formulation. This bounded perimeter constraint was used as a surrogate area constraint since the actual one was nonlinear. However, using a bounded perimeter constraint instead of an exact area constraint can lead to errors in the final area of each department. Goetschalackx (1998) explores the impact of shape constraints on formulation runtime.

A modified MIP-FLP model based on FLP1 was presented by Meller et al.(1999) which was able to improve the model accuracy and approach efficiency. This model is commonly referred to as FLP2. They reformulated FLP1 of Montreuil (1990) by redefining the binary variables and tightening the department area constraints. Based on the acyclic subgraph structure underlying in this model, they proposed some general classes of valid inequalities. Using these inequalities in a branch-and-bound algorithm, they had been able to moderately increase the range of solvable problems. Also this modified MIP-FLP model was able to eliminate some infeasible solutions from the solution space and to improve the algorithm's efficiency due to these valid inequalities.

In order to further improve the performance of the MIP-FLP model and algorithm, a series of enhancements were presented in Sherali et al.(2003). Those new enhancements are based on FLP2, including a novel polyhedral outer approximation scheme for the nonlinear area constraints, some symmetry-avoiding valid inequalities, several surrogate constraints to prevent department overlapping, and a well-designed branching variable selection priority scheme. The computational results presented in Sherali et al. (2003) showed that the accuracy of the final solutions was increased, but more telling, some difficult test cases with eight and nine departments were solved for the first time. Castillo and Westerlund (2005) provided an €-accurate scheme for controlling the department area. They proposed a mixed-integer linear programming model for the block layout design problem with unequal areas that satisfies the area requirements with a given accuracy. The basic aspect of the model consists of an €-accurate representation of the underlying nonconvex and hyperbolic area restrictions using cutting planes. Such representation of the area restrictions allow to solve several challenging test problems to optimality with a guarantee that the final area of each department was within an €% error of the required area.

Even with all these improvements to the MIP-FLP, there were still some difficulties. The major difficulties that arise in solving the MIP-FLP are due to the disjunctive constraints and also the large number of binary integer variables that prevent departmental overlap. Associated with these binary variables, there are numerous infeasible settings, which waste a great deal of computational effort. Hence, Banerjee et al.(1992), Montreuil et al. (1993), Lacksonen (1994, 1997) and Langevin et al.(1994) attempted to solve MIP-FLP models by heuristically fixing a subset of those binary integer variables and then solving the resulting simplified model.

Banerjee et al.(1992) and Montreuil et al.(1993) applied qualitative layout anomalies and design skeletons to the Montreuil-Venkatadri-Ratliff MIP-FLP model (Montreuil et al., 1993). Lacksonen (1994) proposed an approach that combined the Quadratic Assignment Problem (QAP) model with the Montreuil (1990) model. Langevin et al.(1994) proposed a heuristic approach based on the Montreuil (1990) model to solve the spine layout problem, with a main aisle being used for material handling and all departments being located along both sides of an aisle. This approach used a heuristically fixed ordered list as the initial input and is unable to consider all possible solutions. It is also specifically designed for application to the spine layout problem. Lacksonen (1997) proposed a pre-processing heuristic to fix a subset of the total binary variables according to a regression formula based on the area of each department and the material flows associated with each department.

All these heuristics in the researches discussed above were either designed for a specific FLP topology or cannot consider all possible all-rectangular department solutions due to certain pre-processing or restrictions. Few efforts have been made to design a heuristic for the generic continuous-representation-based FLP that is capable of considering all possible all-rectangular department solutions.

On the other side, a great deal of research has been conducted in the Very Large Scale Integration (VLSI) layout design domain, which is similar to the FLP. In VLSI design, different modules are placed onto a chip without overlapping, whereas in the FLP different departments are to be placed into a facility without overlapping. The objective of VLSI design is to minimize the area of the chip while including all modules. In the FLP, the objective is to minimize the material handling costs for a given facility. The sequence pair representation was first presented to solve the VLSI design problem (Sha and Dutton, 1985; Murata et al.1995; Murata and Kuh, 1998a; 1998b). A sequence pair is a pair of module (department) sequences that is used to represent the relative location relationship of the modules (departments) in the VLSI design problem (FLP).

Liu and Meller (2007) introduced this sequence-pair representation in MIP-FLP that was originated in the (VLSI) design literature. They showed that, each sequence pair is feasible in terms of representing the departments' relative locations to one another and preventing departments from overlapping. By incorporating sequence-pair representation, they developed an MIP-FLP-based approach that is not limited to a specific FLP topology. However, their approach was limited to rectangular department shapes, positive inter-departmental relationships, and they did not consider relationships with the perimeter of the facility or columns in the facility. A genetic-algorithm-based heuristic was proposed that combines the sequence-pair representation with the MIP-FLP model. The Genetic Algorithm (GA) was employed to search the solution space and the sequence-pair representation ensured the binary feasibility. The heuristic considered only feasible binary variable settings when searching amongst the all-rectangular-department solution space, and thus, can efficiently solve larger continuous-representation-based FLPs. Meller et. al (2007) also a presented a new formulation for the facility layout problem based on the sequence-pair representation, which was used successfully in VLSI design. By tightening the structure of the problem with this formulation, we have extended the solvable solution space from problems with nine departments to problems with eleven departments.

Solimanpur and Jafari (2008) proposed a mixed-integer nonlinear mathematical programming model for determining the optimum layout in a two-dimensional area. A technique was used to linearize the formulated nonlinear model. An algorithm based on branch-and-bound approach was proposed to obtain the optimal solution of the proposed mathematical programming model. They concluded that the proposed branch-and-bound approach performed inefficient for large-sized problems. Therefore, for large-sized problems, the proposed mathematical programming model should be solved through meta-heuristics like genetic algorithms, tabu search, ant colony optimization etc. Taghavi and Murat (2011) also developed a nonlinear mixed integer model for the integrated facility layout design and flow assignment problem. Since this complex model could not be efficiently solved using classical methods for large problems, therefore, they proposed a novel integrated heuristic procedure based on the alternating heuristic, a perturbation algorithm and sequential location heuristic. Since the alternating heuristic between facility layout design and product-machine assignment sub-problems terminated with local optima, they developed a perturbation algorithm based on assignment decisions. The iterative heuristic solution procedure proposed in this research was employed for solving the large instances of the integrated layout design and machine assignment problem presented in Solimanpur and Jafari (2008).

Anjos and Vanneli (2006) presented a new framework for efficiently finding competitive solutions for the facility layout problem. This framework was based on the combination of two new mathematical programming models. The first model was a relaxation of the layout problem and was intended to find good starting points for the iterative algorithm used to solve the second model. The second model was an exact formulation of the facility layout problem as a non-convex mathematical program with equilibrium constraints (MPEC). Aspect ratio constraints, which are frequently used in facility layout methods to restrict the occurrence of overly long and narrow departments in the computed layouts, were easily incorporated into this framework.

Jankovits et al. (2011) developed a nonlinear model of UA-FLP where they considered the material transportation cost as the objective in determining the optimal layout. Unlike Montreuil (1990) they considered the nonlinear area constraint. Aspect ratio constraints, which is frequently used in facility layout models to restrict the occurrence of overly long and narrow departments in the computed layouts also considered here. Because, long and narrow departments are not realistic as this interrupts the placement of machines and other entities within the department. However, As the maximum allowable aspect ratio becomes smaller the problem becomes more constrained, and feasible solutions become harder to find.

Jankovits et al. (2011) presented a convex-optimization-based framework for efficiently finding competitive solutions for this UA-FLP. The framework is based on the combination of two mathematical programming models. The first model was a convex relaxation of the layout problem that establishes the relative position of the departments within the facility. In the first model, they used circles to approximate the initial positions of departments within the facility. Their first model yields the following improvements: The first improvement is that in the new model the objective function does not improve as the circles start overlapping and the distance between the circle centres becomes less than the distance between the circle centres. A second improvement is the inclusion of some information about aspect ratios. Thirdly, a systematic approach to making parameter choices is introduced.

Using the fixed-outline of the facility and the locations of circles from the first stage model, the second stage model uses semidefinite programming to provide the precise location and rectangular dimensions of the departments while minimizing the layout costs. Semidefinite programming (SDP) refers to the class of optimization problems where a linear function of a symmetric matrix variable X is optimized subject to linear constraints on the elements of X and the additional constraint that X must be positive semidefinite. SDP has been successfully applied in the development of approximation algorithms for several classes of hard combinatorial optimization problems. A mixed integer SDP model was recently proposed in (Takouda al, 2005) to find global lower bounds for the floor planning problem in physical circuit design, a problem closely related to the FLP. Using this SDP Jankovits et al. (2011) formulated the FLP as a convex optimization problem and then solved it to determine the final layout.

Since the UA-FLP is one of the most complex optimization problems and thus falls on the class of NP-hard or NP-complete problems, therefore, the application of exact methods to large instances of the problem is too time consuming; therefore, heuristic methods have been developed to obtain a near optimal solution of the problem. Many researchers worked on this area.

Tate and Smith (1995) applied genetic optimization with an adaptive penalty function to the shape-constrained unequal-area facility layout problem. They implemented a genetic search for unequal-area facility layout, and show how optimal solutions are affected by constraints on permitted department shapes, as specified by a maximum allowable aspect ratio for each department. To affect a search, departmental shapes must be restricted to some subset of possible shapes. The flexible bay structure developed by Tong (1991) was used in this research. The pre-specified rectangular area is divided in one direction into bays of varying width. Each bay is then divided into rectangular departments of equal width but different length. The bays are flexible in that their widths will vary with their number and contents. They chose to encode flexible bay solutions on two distinct chromosomes. The first chromosome carries a permutation of the integers 1 through n ,where n is the number of departments. This sequence represents the sequence of departments, bay by bay, read from top to bottom, left to right. The second chromosome contains an encoding of the number of bays, and where in the sequence the breaks between bays occur. For breeding, they used a variant of uniform crossover (Radcliffe, 1991). Each location in the offspring's sequence is occupied by the department in the corresponding location from one or the other parent with equal probability, so that all common locations in the parents are carried over to the child. Conflicts are then resolved to ensure that each department occurs exactly once in the offspring's encoding. The number and location of bay breaks in the solution is taken without change from one parent or the other, with equal probability.

Tate and Smith (1995) also showed how an adaptive penalty function can be used to find good feasible solutions to even the most highly constrained problems. The adaptive penalty function was developed for highly constrained genetic search (Smith and Tate, 1993). The penalty function uses observed population data during evolution to adjust the severity of the penalty being applied to infeasible solutions. They observed that the degree of infeasibility of any one department is less important to the search process than the number of departments that are infeasible. For example, a solution in which more than half of all departments are slightly infeasible in shape might require extensive modifications to yield a feasible solution, whereas a solution with one extremely infeasible department might be made feasible simply by shifting that department into an adjoining bay. The object of the penalty function is to find feasible solutions; infeasible solutions are attractive only to the extent that they are likely to breed or mutate good feasible solutions. Accordingly, they scaled the penalty function so that a solution with one infeasible department and the best known objective function value would be considered equally promising with the best known feasible solution. They emphasized that the purpose of the penalty function is to guide the search to include near feasible solutions, not to replace the constraints of the original problem formulation. The value of the penalty alters over evolution. For problems with difficult shape constraints, finding any feasible solution can be nearly as difficult as finding near-optimal solutions, heuristics that always preserve feasibility at intermediate points in the search will be unable to operate effectively.

A penalty in the objective function was also employed by Scholz et al. (2009) to avoid the infeasible layouts. They presented a slicing tree based tabu search heuristic for the rectangular, continual plane facility layout problem. The possibility to specify various requirements regarding (rectangular) shape and dimensions of each individual facility was also integrated by using bounding curves. Therefore, it was possible to solve problems containing facilities of fixed and facilities of flexible shapes at the same time. The proposed procedure calculates the layout corresponding to a given slicing tree on the basis of bounding curves. These layouts are slicing structures which are able to contain empty spaces to guarantee that stringent shape restrictions of facilities are kept. Due to these features this approach is better suited for practical use. The tabu search heuristic based on slicing trees and bounding curves minimized the flow costs as well as the size of the resulting layout. A slicing structure results from dividing an initial rectangle either in horizontal or vertical direction completely from one side to the other (so-called guillotine cut) and recursively going on with the newly generated rectangles. A slicing tree is a binary tree which was used to represent such a slicing structure. On the other hand, floor space requirements and geometric characteristics of facilities was expressed by bounding curves. A bounding curve was the borderline between feasible and infeasible dimensions for sites of facilities. So , a slicing tree gave information about the relative location of facilities to each other, but it did not include exact positions and dimensions of facilities. Therefore, they presented a layout generating procedure based on bounding curves.

Facilities can be characterized as four basic types to meet the practical requirements which are the following:

Type 1: Facilities with given area and fully free aspect ratio.

Type 2: Facilities with fixed dimensions.

Type 3: Facilities with given area and restricted range of feasible aspect ratios.

Type 4: Facilities which can have one out of several given implementations.

Scholz et al. (2009) showed that, each of these four types can be represented by a specific bounding curve.

In the following year, Scholz et al. (2010) proposed an extension to the Slicing Tree and Tabu Search based heuristic for practical applications of the facility layout problem. This research considered a very general case of the facility layout problem, which allows incorporating various aspects appearing in real life applications. These aspects include loose requirements on facilities' footprints, each of which only needs to be of rectangular shape and can optionally be restricted concerning the surface area or the aspect ratio. Compared to former approaches other generalizations of practical relevance are multiple, not necessarily rectangular workshops, exclusion zones in workshops, predefined positions of facilities, the consideration of aisles, and the adherence of further restrictions such as the enforced placement of certain facilities next to an exterior wall or a minimum distance between certain pairs of facilities. Although different objectives could be applied, they especially focused on the one in practice, the minimization of transportation costs.

An Ant System (AS) was proposed by Komarudin and Wong (2010) for solving Unequal Area Facility Layout Problems. Until then a formal Ant Colony Optimization (ACO) based metaheuristic has not been applied for solving UA-FLPs. ACO is a discrete optimization technique which imitates the foraging behavior of an ant colony. Particularly, it resembles the behavior of an ant colony in finding the shortest path to reach its food source. After the initialization of parameters and input of problem data, every ACO iteration consists of ant solutions construction, local search procedures (optional), and pheromone information update. The proposed algorithm employed the pheromone information and heuristic information for constructing ant solutions. Like Tate and Smith (1995) an adaptive penalty function was incorporated in the objective function of this research to guide the search process towards feasible solution regions. Specifically, an ant solution was given a penalty value which was proportional to the number of infeasible departments that it contained. A department was considered as infeasible if it violates the department constraints.

As a discrete optimization algorithm, the proposed algorithm used slicing tree representation to easily represent the problems without too restricting the solution space. In this proposed algorithm, slicing tree is implemented using a solution representation which can be divided into three parts: (1) department sequence, (2) slicing sequence, and (3) slicing orientation. The first two parts are represented by integer numbers whereas the last part is represented by binary numbers. The department sequence is the ordering of n departments (represented by integer numbers), which will be transformed into a slicing tree form. The slicing sequence is the ordering of n-1 integer numbers which slices the department sequence. The slicing orientation is represented by n-1 binary numbers, whereby 0 represents a horizontal cut and 1 represents a vertical cut. This type of solution representation separates the department sequence from the slicing sequence and slicing orientation.

Komarudin and Wong (2010) also employed nine types of local search to enhance the search performance of AS. The algorithm randomly selected one of them in an implementation, and recursively repeats it until a stopping criterion is met. As in this case, the stopping criteria for the local search were specified as: (1) the maximum number of steps pre-specified by users, and (2) the number of steps where the local search does not improve the solution quality. Whenever one of these criteria is met, the local search will be terminated. The local search procedures can be classified into two categories. Both were used to provide a robust search for the proposed algorithm. The first category was a neighborhood search of the slicing tree form or structure and the second category was a neighborhood search of the ant solution representation. All the procedures were used because of the large solution space of certain problem instances. All of them had the same probability to occur since the solution space of one problem instance may differ from those of other problem instances and thus, it was hard to predict the most effective local search procedure.

In the same year, Wong and Komarudin (2010) proposed an Ant System (AS) algorithm for solving Unequal Area Facility Layout Problems using Flexible Bay Structure (FBS) representation. They have made an improvement to the FBS representation when solving problems with empty spaces. The two improvements for FBS representation were: (1) extending the feasible solution space by using empty space to fulfill department constraints and (2) improving the objective function by recursively filling the bay with empty space. The authors refer to this representation as modified-FBS (mFBS). The transformation from FBS representation into mFBS solution was based on two procedures:

1. The bay height based on the total area of departments which were located in a bay was determined. Whenever departments that violate department constraints were found, the bay height will be increased by adding an available empty space into the bay to satisfy the constraints. Whenever an empty space is added, it will be proportionally placed in the left and right side of the bay.

2. After all department constraints in each bay were satisfied and if there was still an unoccupied empty space, the middle bay will be selected and the available empty space will be added until the maximum bay height was fulfilled. Similarly, the added empty space will be proportionally placed in the left and right side of the bay. It will be examined whether this improves the objective function value or not. If there was an improvement, the modification will be accepted. Otherwise, the original layout will be restored. This procedure will be repeated for all bays by selecting the nearest bay from the center point of the facility.

Chang and Lin (2012) proposed an ant colony system (ACS) algorithm with local search for solving the unequal area facility layout problem. They also used flexible bay structure (FBS) representation. ACS differs AS by introducing the state transition rule and local pheromone update as well as by using the best-so-far solution update rule (Dorigo and Gambardella, 1997). In general, ACS produces better efficiency than AS. The proposed algorithm employed the pheromone information and heuristic information for constructing ant solutions. It was noted that the two parts of an ant solution were not constructed concurrently. First, the department sequence codes were constructed based on pheromone information and heuristic information. The bay break codes were generated based on the proposed space filling heuristic. They have also imposed a penalty cost for the infeasible layouts in addition to material handling cost in the objective function. Again, in order to enhance the search performance, three types of local search were applied to the best solution of the iteration to further improve it.

A Multidimensional scaling (MDS) was used by Chen et al. (2002) as a dimension reduction tool which arranged facilities in a two-dimensional space while preserving the adjacency relationship between facilities. Multidimensional scaling (MDS) is a useful mathematical tool that enables the analysis of data in areas where organized concepts and underlying dimensions are not well developed. The output of MDS is a scatter diagram and is in turn used as the input or location references for developing into the final block layout. This research employed a non-metric MDS approach for developing scatter diagrams. With a scatter diagram yielding the relative locations of the blocks, the corresponding layout remains to be achieved by satisfying additional constraints such as areas and aspect ratios on the blocks. An algorithm was therefore established to generate a feasible layout based on the scatter diagram. In this research the bay structures of layout were considered where the given floor space was first partitioned horizontally or vertically into bays, which were subsequently partitioned into the blocks. Rotating the scatter diagram about the origin results in different layouts in the bay structure. A simulated annealing approach was adopted to rotate the scatter diagram so that the total cost of traveling between facilities and shape violation in the final layout was minimized. Safizadeh and McKenna (1996) also used MDS, but this work was concerned with equal departmental areas.

A hybrid optimization approach for unequal-area facility layout design was proposed by Mir and Imam (2001). Simulated Annealing (SA) was used to optimize a randomly generated initial placement on an extended plane considering the unequal-area facilities enclosed in magnified envelope blocks. An analytical method was then applied to obtain the optimum placement of each envelop block in the direction of steepest descent. Stepwise reduction of the sizes of the envelop allows controlled convergence in a multiphase optimization process. They considered the usual objective –minimizing material handling cost in their facility layout design. In this research, Mir and Imam (2001) actually modified the univariate search technique with controlled convergence (Imam and Mir, 1993) by making some improvement. This search technique starts the optimization with all blocks randomly placed on an extended plane. The convergence was controlled by carrying out the optimization using magnified envelop blocks which were gradually reduced in sizes until their dimensions become equal to those of the actual facilities. A disadvantage of the univariate search was that it restricted the search for the optimum position of a block in orthogonal directions only. Mir and Imam (2001) presented a substantial improvement of this univariate search technique by applying two major modifications. First, the technique was converted into a hybrid by using SA for obtaining an optimized initial placement of the unequal-area facilities. Secondly, instead of using the orthogonal directions for finding an optimal position of a block, the search was made in the directions of the steepest descent which corresponds to the maximum rate of reduction of the cost function. The first modification offsets the initial solution bias and the second helped in finding a better optimum layout.

Lee et al (2005) proposed an improved genetic algorithm for solving multi-floor facility layout problems having inner structure walls and passages. The proposed algorithm modeled the multi-foor layout of facilities on gene structures. These gene structures consist of a 4ve-segmented chromosome. Improved solutions are produced by employing genetic operations known as selection, crossover, inversion, mutation, and refinement of these genes for successive generations. All relationships between the facilities, passages, and lifts are represented as an adjacency graph. The shortest path and distance between two facilities is calculated using Dijkstra's algorithm of the graph theory. Sadrzadeh (2012) presented a genetic algorithm-based meta-heuristic to solve the facility layout problem (FLP) in a manufacturing system, where the material flow pattern of the multi-line layout was considered with the multi-products. The matrix encoding technique had been used for the chromosomes under the objective of minimizing the total material handling cost. The proposed algorithm produced a table with the descending order of the data corresponding to the input values of the flow and cost data. The generated table was used to create a schematic representation of the facilities, which in turn was utilized to heuristically generate the initial population of the chromosomes and to handle the heuristic crossover and mutation operators.

Bozer and Wang (2012) developed a mixed-integer programming (MIP) model of the UA-FLP. But as obtaining an optimal solution to the MIP model is difficult, and currently only problems with a limited number of departments can only be solved to optimality, therefore, developed a heuristic procedure which uses a ‘‘graph pair’’ to determine and manipulate the relative location of the departments in the layout. The graph-pair representation technique essentially eliminates the binary variables in the MIP model, which allows the heuristic to solve a large number of linear programming models to construct and improve the layout in a comparatively short period of time. The search procedure to improve the layout was driven by a simulated annealing algorithm. The proposed heuristic used only Linear Programming (LP) to generate a layout that is optimum for the binary variables that specify the relative location of the department pairs. The key element of this heuristic is a pair of graphs that control the values of the binary variables. Each graph contains *n* nodes, where *n* is the number of departments and the edges in either graph indicate the relative locations of the departments. Actually proposed heuristic-GRAPH starts by creating an initial layout, which was obtained by solving an LP derived from a graph pair that was constructed .The LP model yields an initial layout if the solution is feasible. Otherwise, a new graph pair is constructed, and the process is repeated until an initial layout is obtained. Using the initial layout as a starting point, GRAPH conducts an SA-based search in order to identify a lower-cost layout, since they considered only material transportation cost in developing optimal layout.

Bozer and Wang (2012) also showed incremental changes to a graph pair can attempt to reduce the layout cost. They proposed two simple methods; namely, ‘‘node relabeling’’ (NR) and ‘‘edge migration’’ (EM). With NR, they randomly picked two nodes in the graph pair and relabel them; i.e., one department swaps its relative location settings with another department. However, unlike Armour and Buffa, with NR, the locations of any two departments can be exchanged, whether or not they are adjacent or equal in area. Also, NR can be used with three departments as well. With EM, a pair of departments was randomly selected, and moved the edge between them from one graph to the other. The direction of the edge in the receiving graph is determined via Dijkstra’s algorithm to prevent cycles. If either direction is feasible, the edge direction is picked randomly.

Ulutas and Konak (2012) introduced an artificial immune system (AIS) based algorithm to solve the unequal area facility layout problem (FLP) with flexible bay structure (FBS). AIS algorithms imitate the immune functions, models and principles to solve complex problems (De Castro & Timmis, 2002). Clonal selection algorithm (CSA), is one of the population based algorithms of AIS. The proposed CSA algorithm had a new encoding and a novel procedure to cope with dummy departments that were introduced to fill the empty space in the facility area. Main principles of Evolutionary Algorithms (EAs) and CSA have some common features, although they may have a different biological inspiration. Crossover and mutation are the basic tools for creating new solutions for a standard EA (i.e GA) while hypermutation and receptor editing operators are distinctive for CSA. In GA, reproductive parents are selected due to their fitness values. A crossover operator is applied, and new offspring are created. Differentiation is mostly achieved by crossover. After applying mutation operator, individuals for the next generation are selected from the whole population. On the other hand, the CSA takes into account the affinity (objective function) values during hypermutation. Mutated antibodies are immediately tested for acceptance or rejection. Hypermutation rates are inversely proportional to the affinity of antibodies. In a standard GA, the mutation rate does not change for different individuals, and is usually considered as a small constant rate for the population. The most remarkable characteristic of the CSA is to use the hypermutation rate as a self-adapting parameter which resembles with the natural immune system.

Unlike the common use of the FBS (Tate and Smith, 1995) which was the two section representation (i.e., the first section gives the department order and the second gives the bay break positions), Ulutas and Konak (2012) proposed a new encoding scheme for the FBS where a randomly generated, integer-valued layout string holds the following three information:

1. Height and width of the facility and number of departments.
2. The number of bays.
3. The sequence of departments that was randomly generated.

Therefore, a single array illustrates the whole information regarding department sequence, the number and position of bays.

All the researches discussed above considered only a single objective in the optimal layout design which was the usual objective-minimization of the material transportation cost. But unequal area FLP comprises a class of extremely difficult and widely applicable optimization problems arising in many diverse areas to meet the requirements for real-world applications. Therefore, the real world facility layout problems are multi-objective in nature. Only a few researchers worked on multi-objective unequal area facility layout problems. Ripon et al (2011) considered a qualitative qualitative which was the closeness relationship between the departments in addition to material transportation cost. This qualitative objective aims at maximizing the maximizing the closeness relationship (CR) scores between facilities based on the placement of facilities that utilize common materials, personnel, or utilities adjacent to one another, while separating facilities for the reasons of safety, noise or cleanliness etc. Since there were multiple and conflicting objectives, the researchers developed a set of pareto optimal solutions for this multi-objective optimization problem. They also presented a multi-objective genetic algorithm in obtaining the pareto optimality. Slicing tree structure representration was used for the layout problem.

The multi-objective unequal area facility layout problem of Ripon et al (2011) was again solved by Ripon et al (2013) to obtain pareto optimality by using another evolutionary approach. This research employed the variable neighborhood search (VNS) with an adaptive scheme that presents the final layouts as a set of Pareto-optimal solutions. The VNS is an explorative local search method whose basic idea is systematic change of neighborhood within a local search. Traditionally, local search was applied to the solutions of each generation of an evolutionary algorithm, and has often been criticized for wasting computation time. To address these issues, this approach was composed of the VNS with a modified 1-opt local search, an extended adaptive local search scheme for optimizing multiple objectives, and the multi-objective genetic algorithm (GA). Unlike conventional local search, this adaptive local search scheme automatically determined whether the VNS was used in a GA loop or not. They investigated the performance of the proposed approach in comparison to multi-objective GA-based approaches without local search and augmented with traditional local search and found their approach to perform better.

Aiello et al. (2012) considered four objectives in developing their facility layout design. They took into account material handling costs, aspect ratio, closeness and distance requests among the departments. This paper proposed a new multi objective genetic algorithm (MOGA) for solving unequal area facility layout problems. The genetic algorithm suggested is based upon the slicing structure where the relative locations of the facilities on the floor were represented by a location matrix encoded in two chromosomes. A block layout was constructed by partitioning the floor into a set of rectangular blocks using guillotine cuts satisfying the areas requirements of the departments. In the following year, Aiello et al. (2013) proposed a non dominated ranking Multi Objective Genetic Algorithm and electre method for the mutiobjective unequal area facility layout problem of Aiello et al. (2012). problem in two subsequent steps: in the first step, the Pareto-optimal solutions are determined by employing Multi Objective Genetic Algorithm (MOGA) implementing four separate fitness functions within a Pareto evolutionary procedure, following the general structure of Non-dominated Ranking Genetic Algorithm (NRGA) and the subsequent selection of the optimal solution is carried out by means of the multi-criteria decision-making procedure Electre. This procedure allows the decision maker to express his preferences on the basis of the knowledge of candidate solution set. Quantitative and qualitative objectives were considered referring to the slicing-tree layout representation scheme. Electre is a multi-criteri a decision-ma king procedure that can be applied when a set of alternatives must be ranked according to a set of criteria reflecting the decision maker's preferences. Relationships between alternatives and criteria are described using attributes referred to the aspects of alternatives that are relevant according to the established criteria. In multi-criter ia decision problems, although logical and mathematical conditions required to determine an optimum do not exist, a solution representing a good compromise according to the conflicting criteria established can be individuated. They showed that the proposed method in this research yield better results for their multi-objective UA-FLP than the previous one.

None of the above mentioned papers considered the makespan of the overall system in developing their facility layout model. This thesis work developed a multi-objective unequal area facility layout model considering makespan of the overall system in addition to material transportation cost and closeness rating score. Considering makespan of the overall system in developing a facility layout model is more practical because it plays the most significant role in customer satisfaction in today's competitive market.

CHAPTER I

INTRODUCTION

Determining the physical organization of a production or service facility is defined to be the facility layout problem. The facility layout problem (FLP) is concerned with determining an efficient layout of facilities on a planar site. This problem occurs in various contexts, e.g. arranging machines in a workshop or arranging buildings on a factory premises. Minimizing the sum of material transportation costs between facilities is the most common goal in designing a facility layout. However, the general objective is to arrange the facilities in such a way that ensures a smooth work flow. Development of a facility layout depends on some important issues like-material handling activity between facilities, Proximity priorities among facilities etc. An efficient layout is crucial for virtually any organization that creates a product or delivers a service. Therefore, the facility layout design has a significant impact on the overall operating efficiency and productivity of the facility. So, it is important to optimize the facility layout problem to develop an efficient layout for the facility.

1.1 Rationale of the study

In today's world, Facility layout problem is a fundamental optimization problem encountered in many manufacturing and service organizations. Facility layout plays a key role for the companies and it is an inseparable part of the manufacturing system design process. Companies aim to develop an effective layout for their manufacturing system in order to obtain its effective utilization. A poor layout can lead to the accumulation of work-in-process inventory, overloading of material handling systems, inefficient set-ups, and longer queues (Chiang and Chiang 1998). Furthermore, the facility layout problem represents a costly, long-term investment; hence, modifications that require large expenditures cannot easily be done. Re-layout of facilities is not only time consuming but also disrupts worker activities and the flow of materials (Sule 1994). In this regard, optimization of a facility layout model and thus finding an optimum facility layout has paid extensive attention among the researchers. In the past, the FLP was mostly modeled as a quadratic assignment problem which assumes all facilities have equal areas, but in reality, facilities are usually unequal-sized. Again, most of the layout model aimed at minimizing the material transportation cost in developing the optimum layout and only very few considered the closeness relationship among the facilities and none of them considered the makes pan of the overall manufacturing system which plays the most significant role in customer satisfaction. Therefore, considering the makespan of the overall system along with material handling cost and closeness relationship between the facilities an unequal-area facility layout model is developed to obtain an efficient layout.

1.2 Objectives with Specific Aims and Possible Outcomes

The specific objectives of this research are the following:

i. To develop a constrained nonlinear model for unequal area facility layout problem considering material handling cost, closeness rating score and makespan of the overall system.
ii. To optimize the formulated multi-objective mixed-integer nonlinear model by minimizing the material handling cost and makespan of the overall system as well as maximizing the closeness rating score simultaneously using spatial branch-and-bound algorithm and a multi-objective genetic algorithm.
iii. To compare the results obtained from the spatial branch-and-bound algorithm and the developed multi-objective genetic algorithm.

This thesis work presents possible clues in the development of a multi-objective unequal-area facility layout model by providing mathematical results to help on understanding, formulation and analysis of such mathematical facility layout model.

1.3 Outline of Methodology

This research work is theoretical in nature. A mathematical facility layout model is developed considering some factors significantly affecting the layout decisions such as material handling cost, closeness rating score and makespan of the overall system. The model is composed of some mathematical equations used to determine the numerical values of some decision variables (centroid locations of each department, length width of each department). The research methodology is outlined as follows:

i. The objective functions will be developed for material handling cost, closeness rating score and makespan of the overall system.
ii. The objectives will be normalized and then combined to obtain a scalarized objective function value.
iii. Mathematical equations will be developed for non-overlapping constraints, constraints for bounding the departments within the facility; nonlinear area constraints and maximum aspect ratio constraints will be developed.
iv. A numerical example is considered to illustrate and explain the facility layout model.
v. A multi-objective genetic algorithm was developed incorporating slicing tree structure.
vi. The multi-objective facility layout model is optimized using spatial-branch and-bound algorithm and the slicing tree embedded multi-objective genetic algorithm.
vii. The results obtained from the spatial branch-and-bound algorithm is compared with that of genetic algorithm.

CHAPTER III

MODEL FORMULATION

3.1 Problem Identification

A basic assumption about a facility layout design was that the departments are all equal in area. In addition, it was assumed that the facility layout design is only affected by the cost of transporting material between the facilities. But in reality the departments of a facility are not equal in area. Again, a facility layout design does not solely affected by the layout cost; rather many other factors have significant impact on an efficient layout design. The qualitative factors such as adjacency of facilities due to utilization of common personnel, materials or other utilities as well as separation of facilities for the reasons of safety, noise, or cleanliness also affects the effective layout design. In addition, in this competitive market, customer satisfaction plays an important role in the organization's success. Makespan of the overall system plays the most significant role in customer satisfaction though it was not considered so far in facility layout design. Therefore, makespan of the overall system in addition to the qualitative factors are more realistic factors to be considered along with material transportation cost in designing an efficient layout.

3.2 Problem Definition

The purpose of this thesis work is to extend the previous research in developing the facility layout model by incorporating the concept of qualitative factors and customer satisfaction in addition to cost function. The qualitative factors are quantified by the closeness rating score between the departments and customer satisfaction is quantified by makespan of the overall system. In this thesis work, the total cost per unit distance of transporting items between departments, closeness rating between departments and total time per unit distance of transporting items between departments are known parameter. It is also assumed that, the site for the facility already exists and the area required for each department within the facility is already known. Therefore, the total area available for the facility and the area of each department within the facility are also known parameter. The facility layout model is formulated to determine the optimal layout, i.e to determine the optimal arrangement of the departments within the facility, optimal dimensions of each department in order to minimize the total cost of material transportation and makespan of the overall system as well as to maximize the total closeness rating score simultaneously. Therefore, this optimization problem is multi-objective in nature. Mathematical equations are obtained and derived for material transportation cost, closeness rating score and makespan of the overall system. After that, the objective functions are normalized and combined to have a scalarized objective function. Mathematical equations are also developed for the non-overlapping constraints, constraints for bounding the departments within the facility, nonlinear area constraints and maximum aspect ratio constraints. Since the facility layout model incorporated customer satisfaction in unequal-area facility layout design, it is more practical than other researches published in the literature. Again, so far in the literature, Mixed-integer nonlinear programming formulation was not found. Since the developed facility layout model is a constrained multi-objective mixed-integer nonlinear (MINLP) model, a spatial branch and bound algorithm (sBB) has been employed to solve this model. Again, a multi-objective genetic algorithm based on slicing tree structure has also been employed to solve this model since it is best suited meta-heuristics for this type of problem than any other meta-heuristics.

3.3 Assumptions of the Study

Some assumptions are made in performing this thesis work which are the following:

I. The total area of the facility is known and fixed.
II. Department areas are fixed and known.
III. Departments may have unequal areas.
IV. Departments are either square or rectangular in shape.
V. Departments are of variable dimensioned. i.e., their length and widths are variable.
VI. Departments may have free orientations. i.e., departments may be either horizontally or vertically oriented. If the longer side of the department is

parallel to the x-axis, the department is horizontally oriented, if the longer side is parallel to y-axis, the department is vertically oriented.

VII. For a particular type of customer order the sequence of operation is fixed.

VIII. Material flows between the centres of the departments.

IX. Material handling cost, closeness rating and material handling time are known and deterministic.

X. Material handling cost and makespan are distance based objectives and their values also depend on the mode of transportation.

XI. The distances are calculated from the centre of one department to another.

3.4 Mathematical Modeling

In this research, a new mathematical model for unequal area facility layout problem is proposed which incorporates makespan of the overall system as part of an integrated model. The model is developed to determine the optimal arrangements as well as the optimal dimensions of the departments within the facility by simultaneously optimizing the objectives. The objectives of this multi-objective facility layout model are-

I. Minimizing material handling cost

II. Maximizing closeness rating score

III. Minimizing makespan of the overall system.

Before proceeding to the mathematical model, some parameters and variables of the model are introduced in the following:

Parameters

C_{ij} Cost of transporting materials per unit distance from department i to department j

R_{ij} Closeness rating value between department i and department j

t_{ij} Time required to transport materials per unit distance from department i to department j

L	Length of the plant floor
W	Width of the plant floor
a_i	Area of department i

Variables

(x_i, y_i)	Centroid of department i
l_i	Length of department i
w_i	Width of department i
p_{ij}	Binary variable representing the adjacency between the departments i and j
d_{ij}	Distance between the centroids of department i and department j.

In this model, the rectilinear distance is used, because in case of square or rectangular shaped departments, with walls and corridors a rectilinear assumption is more realistic. Therefore,

$d_{ij} = d_{ij}^x + d_{ij}^y$ (d_{ij}^x and d_{ij}^y are horizontal and vertical distances between the centroids of the departments i and j)

Here, $d_{ij}^x = | x_i - x_j|$ and $d_{ij}^y = | y_i - y_j|$

So, $d_{ij} = | x_i - x_j| + | y_i - y_j|$

Now using this parameter and variable notations, the developed facility layout model is presented as follows:

3.4.1 Material Handling Cost

In the literature, the efficiency of a facility layout is typically measured in terms of material handling cost (MHC). The most usual goal of previous facility layout model was to minimize the total material transportation cost between the departments which depends on the mode of transportation. Tompkins (2003) stated that 10-30 % of MHC can be reduced by having an efficient facility layout. Therefore, Minimization of material handling cost is one of the objectives of this multi-objective facility layout model. The MHC between two department i and j is computed by multiplying the cost of transporting materials per unit distance from department i to department j (C_{ij}) and distance between the centroid of department i and department j (d_{ij}). Therefore, for *n* number of departments within the facility, the total material handling cost can be expressed as-

$$MHC = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} C_{ij} \times d_{ij}$$

Since, rectilinear distance is used, therefore, the expression of material handling cost becomes-

$$MHC = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} C_{ij} \{ |x_i - x_j| + |y_i - y_j| \} \qquad (3.1)$$

3.4.2 Closeness Rating Score

Though the material handling cost was used to represent the efficiency of a facility layout in most of the papers in the literature, it does not consider the qualitative factors between the departments which also affect the efficiency of a layout. It is very usual in a facility that, some departments utilize common materials, require common personnel etc. in that case, these departments should be adjacent to each other. On the other hand, there may be some reasons such as safety, noise or cleanliness that enforce the separation of some departments. So, it is necessary to quantify the closeness requirement of two departments and a closeness rating value (R_{ij}) is assigned between the departments i and j depending on their closeness preferences. For separation requirement, a negative value of R_{ij} is assigned. Therefore, efforts have been made to maximize the total closeness rating score (CRS). The following equation represents the total closeness rating score-

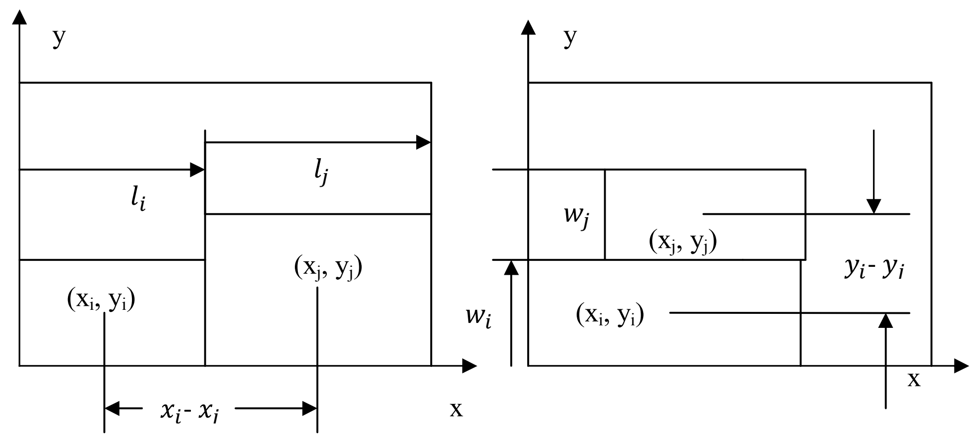
$$CRS = \sum_{i=1}^{n-1} \sum_{j=i+i}^{n} R_{ij} \times p_{ij} \tag{3.2}$$

p_{ij} is a binary variable whose value depends on the adjacency of the departments i and j.

$$p_{ij} = \begin{cases} 1, & if\ the\ department\ i\ and\ j\ have\ a\ common\ boundary \\ 0, & otherwise \end{cases}$$

So, p_{ij}=1 if $|x_i - x_j| = 0.5\ (l_i + l_j) or\ |y_i - y_j| = 0.5(w_i + w_j)$ (3.3)

These adjacency situations are illustrated in the following Fig. 3.1. Figure 3.1(a) shows the adjacency in x-direction and Fig. 3.1(b) shows adjacency in y-direction.



(a) Adjacency in x-direction

(b) Adjacency in y-direction

Figure 3.1: Adjacency between the departments

3.4.3 Makespan of the Overall System

In this competitive world, customer satisfaction is greatly affected by on time delivery of a lot which in turn depends on the makespan of the system. So, minimization of makespan has a key role on customer satisfaction. Therefore, incorporating minimization of makespan as one of the objectives in developing the facility layout model is more practical.

In this research, the goal is to minimize the makespan of the facility by minimizing the total material transportation time which is obtained by rearranging the departments. The material handling time per unit distance between two departments depends on the mode of transportation. The material transportation time between two department i and j can be computed by multiplying the time required in transporting materials per unit distance from department i to department j (t_{ij}) and distance between the centroid of department i and department j (d_{ij}). Therefore, for *n* number of departments within the facility, the total material transportation time or in other words makespan can be expressed as-

Makespan = $\sum_{i=1}^{n-1}\sum_{j=i+1}^{n} t_{ij} \times d_{ij}$

For rectilinear distance, the above expression can be written in the form –

$$\text{Makespan} = \sum_{i=1}^{n-1}\sum_{j=i+1}^{n} t_{ij}\{|x_i - x_j| + |y_i - y_j|\} \quad (3.4)$$

3.4.4 Facility's Dimensional Constraints

In this model it is assumed that, the dimension of the total facility is known. In solving the UA-FLP the total area of the facility is to be partitioned into departmental sub-regions to locate the departments within the facility. This situation imposes a constraint in the model which is, all the departments must be within the facility area. Therefore, to prevent the departments from exceeding the facility's dimension in the positive x (+ve) direction, the right wall of each department should be to the left of the facility's right wall. This situation is represented in the following Fig. 3.2(a). This constraint can be expressed mathematically as-

$$(x_i + 0.5l_i) \leq L \quad \forall\, i \quad (3.5)$$

Again, to prevent the departments from exceeding the facility's dimension in the negative x (-ve) direction, the left wall of each department should be to the right of the facility's left wall. This situation is represented in the following Fig. 3.2(b). This constraint can be expressed mathematically as-

$$(x_i - 0.5l_i) \geq 0 \quad \forall\, i \quad (3.6)$$

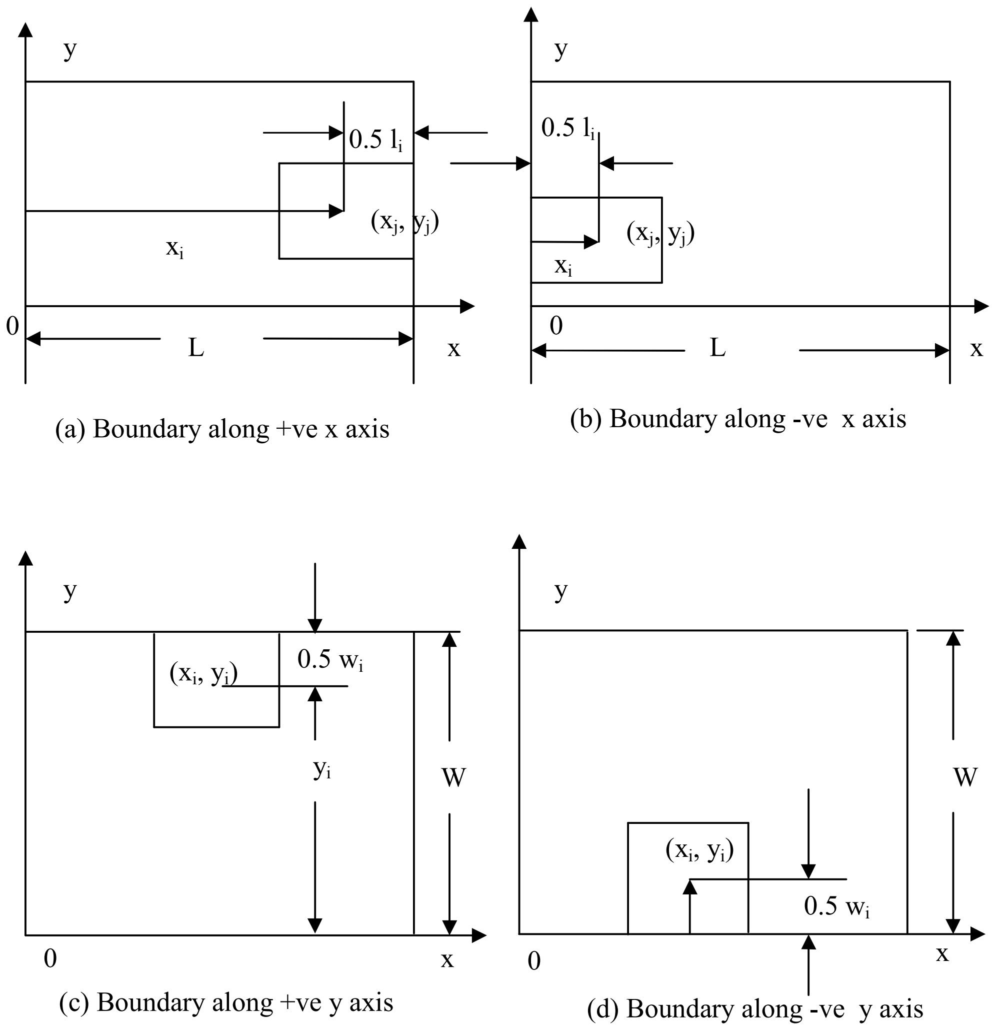


Figure 3.2: Bounding the departments within the facility.

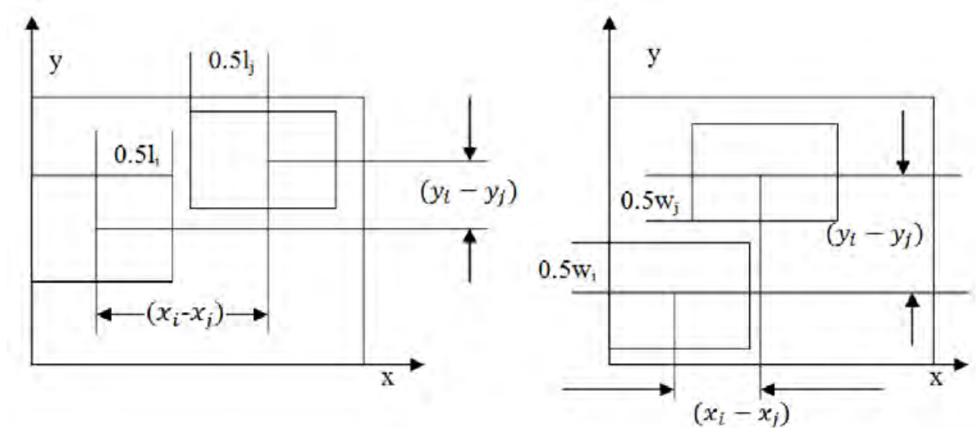
Similarly, to prevent the departments from exceeding the facility's dimension in the positive y (+ve) direction, the top wall of each department should be to the bottom of the facility's top wall. In case of the negative y (-ve) direction, the bottom wall of each department should be to the top of the facility's bottom wall. These conditions are represented in Fig. 3.2(c) and Fig. 3.2(d). These two constraints are expressed by the mathematical equations (3.7) and (3.8) respectively.

$$(y_i + 0.5w_i) \leq \mathrm{W} \quad \forall\, i \tag{3.7}$$

$$(y_i - 0.5w_i) \geq 0 \quad \forall\, i \tag{3.8}$$

3.4.5 Non-overlapping Constraints

The goal of this unequal area facility layout model is to locate n number of departments within a facility area. Unlike the QAP of facility layout model, the locations of the departments are not predefined. Therefore, there is a possibility of the department to overlap with each other. This justifies the incorporation of non-overlapping constraints in this facility layout model. Overlapping can occur either in x-direction or in y-direction or both.



(a) Non-overlapping in x direction (b) Non-overlapping in y direction

Figure 3.3: Non-overlapping of the departments

Actually, these constraints state that, if the distance between the centres of the two departments i and j in the y-direction (y_i-y_j) is less than half of the sum of their widths (0.5(w_i+w_j)), then the departments i and j cannot overlap in the x direction (Fig. 3.3(a)). On the other side, if the distance between the centres in the x-direction (x_i-x_j) is less than half of the sum of their lengths (0.5(l_i+l_j)), then the departments i and j cannot overlap in the y direction (Fig. 3.3(b)). These constraints can be expressed mathematically as-

$$\text{If } |y_i - y_j| < \frac{1}{2}(w_i + w_j) \text{ then } |x_i - x_j| - \frac{1}{2}(l_i + l_j) \geq 0 \quad \forall\, i,j \tag{3.9}$$

$$\text{If } |x_i - x_j| < \frac{1}{2}(l_i + l_j) \text{ then } |y_i - y_j| - \frac{1}{2}(w_i + w_j) \geq 0 \quad \forall\, i,j \tag{3.10}$$

These equations are similar to that of Van Camp et. al (1991).

3.4.6 Aspect Ratio Constraints

The ratio of the maximum side length to the minimum side length is called the aspect ratio (α). A limit to the maximum aspect ratio is imposed to avoid long and narrow departments where the centroids are very close to one another. Such long and narrow departments are realistic as it interrupts the arrangement of machines and other resources within the department. However, as the aspect ratio becomes smaller the problem becomes more constrained and feasible solutions become harder to find. This aspect ratio constraint can be expressed as-

$$\frac{\max\{l_i, w_i\}}{\min\{l_i, w_i\}} \leq \alpha_i \quad \forall\, i \tag{3.11}$$

Where, α_i is the aspect ratio for department i. In this model, the value of α for the departments will be varied within a relevant range to show the impact of aspect ratio on the solution.

3.4.7 Nonlinear Area Constraints

It was previously assumed that, each departments have a given fixed area but are variable dimensioned. Therefore, the solution to this model will provide the optimal length and width of each department. Therefore, the product of the length (l_i) and width (w_i) of each department should be equal to the given area (a_i) of the department i. This nonlinear area constraint can be expressed mathematically as-

$$l_i w_i = a_i \quad \forall\, i \tag{3.12}$$

3.4.8 Multi-Objective Optimization

Since the facility layout model developed in this thesis has multiple objectives, it falls on the class of multi-objective optimization. The multi-objective optimization problem (MOOP) is an area of multiple criteria decision making, that is concerned with mathematical optimization problems involving more than one objective function to be optimized simultaneously.

Real-life scientific and engineering problems typically require the search for satisfactory solution for several objectives simultaneously. It is also common that conflicts exist among the objectives. In the presence of such multiple and conflicting objectives, the resulting optimization problem gives rise to a set of optimal solutions, instead of one absolute optimal solution. Multiple optimal solutions exist because no single solution can be optimal for multiple conflicting objectives. These multiple solutions, namely the Pareto-optimal solutions, are optimal in the wider sense that no other solutions in the search space are superior when all the objectives are considered. Since none of these solutions can be said to be an 'absolute optimum', it is reasonable for the users to find as many different Pareto-optimal solutions as possible. The set of such solutions is called Pareto front. Among these solutions, the designer is free to select any solution that offers the most profitable trade-off among the objectives. The following Fig. 3.4 presents a Pareto front for two objectives, which are subject to minimization.

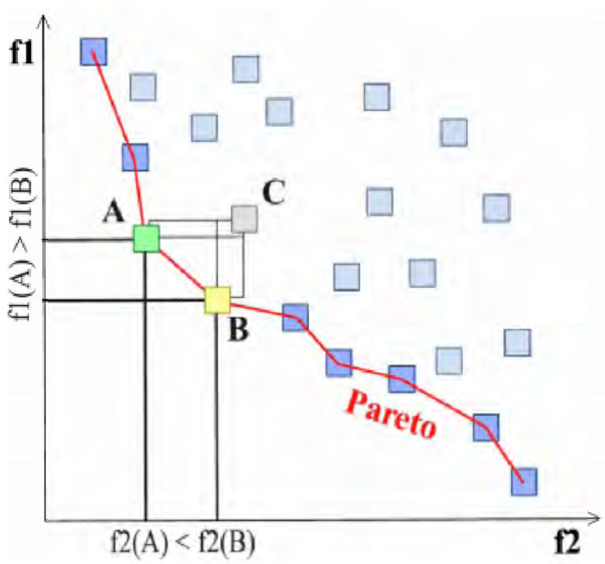


Figure 3.4: Pareto front (Ripon et al., 2011)

Therefore, since practical FLPs are multi-objective by nature, they require the decision makers to consider a number of criteria before arriving at any conclusion. A solution that is optimal with respect to a certain given criterion might be a poor candidate for some other criterion. Hence, it is desirable to generate multiple optimal layouts considering multiple objectives according to the requirements of the production order or customer demand. Then, the production manager can selectively choose the most demanding one among all of the generated layouts for specific order or customer demands.

The developed facility layout model has three objectives and conflicts exist among the objectives. Therefore, this optimization model has multiple optimal solutions. So, a multi-objective optimization approach is required in solving this optimization model. In this thesis work, weighted sum approach has been used to solve the multi-objective optimization problem. The weighted sum approach is the most widely used solution approach for multi-objective optimization problems. In the classical weighted sum approach a weight w_i is assigned to each objective function $z_i(x)$ so that the multi-objective problem is converted to a single objective problem with a scalar objective function as follows:

$$\text{Min } Z = w_1 Z_1(x) + w_2 Z_2(x) + \cdots + w_k Z_k(x) \quad (3.13)$$

Where, $\sum w_i = 1$

This approach is also called the priori approach since the user is expected to provide the weights. Solving the problem with a given weight vector $w = \{w_1, w_2, \dots w_k\}$ yields a single solution. To obtain multiple optimal solutions the problem is to be solved multiple times with different weight combinations.

However, the classical weighted sum approach has some deficiencies in combining the multiple objective functions. They are the following:

1. All factors may not be represented on the same scale: for example, values for material handling cost may range from zero to a large numerical value, while closeness rating values may range from -1 to 5. As a result, the closeness ratings would be dominated by material handling cost and has little impact on the final layout.
2. Measurement units used for the objectives may be incomparable: the closeness rating represents an order preference indicating the necessity that given facilities is to be located close together. The total closeness rating score is only an ordinal value; on the other hand, the material handling is measured according to cost. Aggregation of these two values with different measurement units in an algebraic operation is unsuitable.

To avoid these deficiencies Harmonosky and Tothero (1992) suggested an approach that normalizes all objectives, before combining them. Once all objectives have been quantified, the data is normalized. To normalize an objective, each relationship value is divided by the sum of all relationship values for that objective. The following Eq. (3.14) is used to do this.

$$T_{ijm} = S_{ijm} / \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} S_{ijm} \tag{3.14}$$

Where, S_{ijm} represents the relationship value between departments i and j for objective m and T_{ijm} represents the normalized relationship value between i and j for objective m.

After normalization, the Eq. (3.13) is changed to the following Eq. (3.15)

$$\text{Min } Z = w_1 Z'_1(x) + w_2 Z'_2(x) + \cdots + w_k Z'_k(x) \tag{3.15}$$

As discussed earlier, the developed multi-objective facility layout model in this thesis work has three objectives which are minimizing material handling cost, maximizing closeness rating score and minimizing makespan of the overall system. The weighted sum approach is used to solve this multi-objective model. Therefore, after normalizing the objective functions following the approach that has been discussed above, the objective functions have been combined to obtain a scalar objective function. So, normalizing and combining Eqs. (3.1), (3.2) and (3.4) yields the scalarized objective function in Eq. (3.16).

$$\begin{aligned} Minimize\ Z = {} & w_1 \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} C_{ij}\{|x_i - x_j| + |y_i - y_j|\} - w_2 \sum_{i=1}^{n-1} \sum_{j=i+i}^{n} R_{ij} \times p_{ij} + \\ & w_3 \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} t_{ij}\{|x_i - x_j| + |y_i - y_j|\} \end{aligned} \tag{3.16}$$

Where, $\sum_{i=1}^{3} w_i = 1$

A negative sign is assigned to the closeness rating equation in the weighted objective function, because maximizing a function is equivalent to minimize the negative of this function. Multiple pareto optimal solutions can be obtained by solving Eq. (3.16) using different weight combinations.

Now, combining all the mathematical equations the integrated facility layout model can be expressed as the following:

$$Minimize\ Z = w_1 \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} C_{ij} \{|x_i - x_j| + |y_i - y_j|\} - w_2 \sum_{i=1}^{n-1} \sum_{j=i+i}^{n} R_{ij} \times p_{ij} + w_3 \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} t_{ij} \{|x_i - x_j| + |y_i - y_j|\}$$

Subject to,

$$(x_i + 0.5 l_i) \leq \text{L} \quad \forall\, i$$

$$(x_i - 0.5 l_i) \geq 0 \quad \forall\, i$$

$$(y_i + 0.5 w_i) \leq \text{W} \quad \forall\, i$$

$$(y_i - 0.5 w_i) \geq 0 \quad \forall\, i$$

$$\text{If } |y_i - y_j| < \tfrac{1}{2}(w_i + w_j) \text{ then } |x_i - x_j| - \tfrac{1}{2}(l_i + l_j) \geq 0 \quad \forall\, i, j$$

$$\text{If } |x_i - x_j| < \tfrac{1}{2}(l_i + l_j) \text{ then } |y_i - y_j| - \tfrac{1}{2}(w_i + w_j) \geq 0 \quad \forall\, i, j$$

$$\frac{\max\{l_i, w_i\}}{\min\{l_i, w_i\}} \leq \alpha_i \quad \forall\, i$$

$$l_i w_i = a_i \quad \forall\, i$$

The developed mathematical model is a constrained multi-objective mixed integer nonlinear programming (MINLP) model. Optimization of MINLPs is precisely so difficult, because they combine all the difficulties of both of their subclasses: the combinatorial nature of mixed integer programs (MIP) and the difficulty in solving nonconvex (and even convex) nonlinear programs (NLP). Since, the subclasses MIP and NLP are among the class of theoretically difficult problems (NP-complete), so it is not surprising that solving MINLP can be a challenging and daring venture. Therefore, recently MINLPs have been experiencing tremendous growth and a flourish of research activity. Only a few exact algorithms have been developed in solving the MINLP. Since, solving such complex optimization problem requires more computational time and effort and sometimes it fails to reach optimality for large sized problems, therefore some researchers approached this type of problem using meta-heuristics. In this research, the MINLP model was first optimized using a spatial branch and bound algorithm as it is one of the established exact method in solving such problems. After that, a multi-objective genetic algorithm based on slicing tree structure has also been employed to optimize the model. A comparison is also made between the results of spatial branch-and-bound algorithm and genetic algorithm.

CHAPTER V

LAYOUT MODEL USING GENETIC ALGORITHM

Various methodologies and procedures have been proposed to solve unequal-area facility layout problems, which could be classified into: (1) exact procedures, (2) heuristics and improvement procedures and (3) meta-heuristic. UA-FLPs are computationally difficult, in an *n* facility problem we would have to evaluate *n!* different layouts. Due to the combinatorial nature of the problem, exact (optimal) algorithms have been successfully applied only to small problems (< 11 facilities), but they require high computational efforts and extensive memory capabilities. Again, despite the guarantee of global optimality by the exact algorithm, their efficiency as well as accuracy decreases as the number of departments increases. As a result, metaheuristic algorithms have got the attention in recent years to solve FLPs. This is due to their ability to generate feasible solutions in the least possible computational time.

The developed unequal-area facility layout model is a multi-objective constrained MINLP problem which falls on the class of the most difficult optimization problem. In the previous chapter, this model is optimized using a reformulation-based branch and bound algorithm which is a heuristic embedded exact algorithm. But, it has been observed that, it requires more computational time in reaching optimality. Therefore, in this chapter, a meta-heuristic method named Random weighted genetic algorithm (RWGA) is used to approach the developed model.

The primary difficulties associated with UA-FLP have to do with the vast number of possible physical layouts, and with the existence of many locally optimal layouts that are poor compared with the global optimum layout. For such a problem, one might expect parallel search methods to perform better than strictly serial searches, and randomized search methods to perform better than greedy or enumerative searches. Since Genetic algorithms combine both of these attributes in a parallel, stochastic heuristic, therefore a multi-objective genetic algorithm (GA) based on the slicing tree structure (STS) is used to optimize the developed model.

GA is a well known search algorithm and has been widely used in different field of study. STS is also an effective tool for floor plan design commonly used in VLSI design. This representation was first introduced to the facility layout literature by Tam (1992). Another alternative of STS is the Flexible Bay Structure (FBS). FBS is defined as a continuous layout representation allowing the departments to be located only in parallel bays with varying widths. As stated in Konak et al. (2004), there is no limit on the number, width and content of bays; that's why this representation is called flexible bay. But while using FBS, every department in a bay must be of same width/height. It imposes a restriction on the shape of a department and hampers flexibility. Use of STS provides more flexibility than FBS. Moreover Ulutas and Konak, (2012) showed that for some cases result can be worse than slicing tree. In the following section definition, basic idea, components of GA and a brief discussion on STS will be provided.

5.1 Genetic Algorithm

GA is an iterative procedure maintaining a population of structures that are candidate solutions to specific domain challenges. During each temporal increment (called a generation), the structure in the current population are rated for their effectiveness as domain solutions, and on the basis of these evaluations, a new population of candidate solution is formed using specific genetic operators such as reproduction, crossover and mutation.

The genetic algorithm technique was first invented by Holland, 1975 and has been successfully applied to numerous large search space problems by Davis (1987); Forrest (1993); Goldberg (1989). It is a search algorithm based on the mechanics of the natural selection process (biological evolution). The most basic concept is that the strong tend to adapt and survive while the weak tend to die out. That is, optimization is based on evolution and the "Survival of the fittest" concept. GA has the ability to create an initial population of feasible solutions, and then recombine them in a way to guide their search to only the most promising areas of the state space. Each feasible solution is encoded as a chromosome (string) also called genotype and each chromosome is given a measure of fitness via a fitness (evaluation or objective) function. The fitness of a chromosome determines its ability to survive and produce offspring. A finite population of chromosome is maintained. GA uses probabilistic rules to evolve a population from one generation to the next. They have a high built in degree of randomness to escape from local optima and inferior regions of the solution space. Through parallel processing on a population of randomly generated chromosomes, it speeds up the whole search procedure. There is an abundance of applications of GAs in almost any field, including an extensive use in solving the FLP. In short it is a robust search technique and produce "close" to optimal results in a "reasonable" amount of time.

5.1.1 Pseudo-code and Flow Chart for Generic GA

The procedure of a generic GA is shown as follows:

```
Begin

    INITIALIZE population with random candidate solutions;
    EVALUATE each candidate;

    Repeat until TERMINATION-CONDITION is satisfied

        1. SELECT parents;
        2. CROSSOVER between parents
        3. MUTATE the resulting children
        4. EVALUATE children;
        5. SELECT individuals for next generation

End
```

The procedure of generic GA can be represented by the following Fig. 5.1

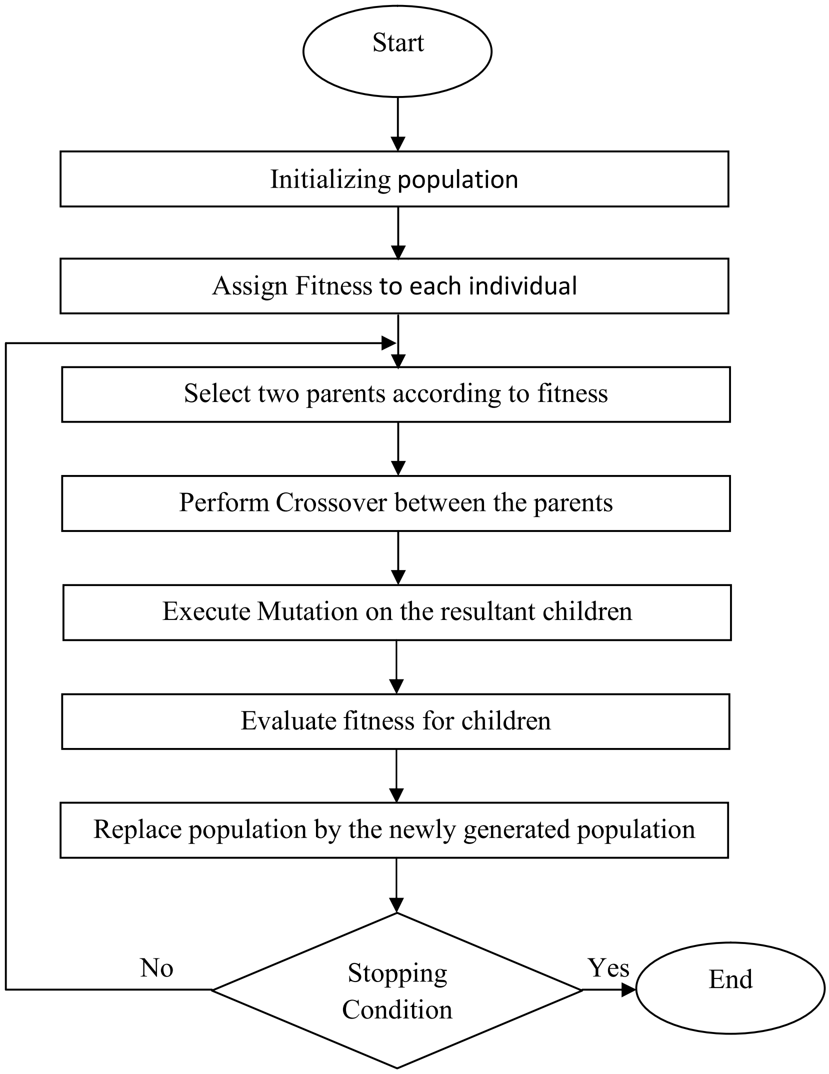


Figure 5.1: Flow chart of genetic algorithm

5.1.2 Basic Components of GA

As described in Davis (1987), a genetic algorithm has five components:

1. A means of encoding solutions to the problem as a chromosome.
2. A function that evaluates the "fitness" of a solution.
3. A means of obtaining an initial population of solutions.
4. Reproduction operators for the encoded solutions.
5. Appropriate settings for the genetic algorithm control parameters.

5.1.2.1 Chromosome Encoding

The first, and perhaps the most important, step in applying a genetic algorithm to a problem is to choose a way to represent a solution to the problem as a finite-length string over a finite alphabet. These strings are referred to as chromosomes. The values on the chromosome may be arranged and interpreted as needed. They may represent Boolean values, integers, or even discretized real numbers. Complex chromosome can have combination of two or more type. Some encodings, which have been already used successfully, have been introduced here.

Binary - Binary encoding is the most common, mainly because first works about GA used this type of encoding. In binary encoding every chromosome is a string of bits, 0 or 1.

Example: Chromosome A: 101100101100

Permutation - Permutation encoding can be used in ordering problems, such as travelling salesman problem or task ordering problem. In permutation encoding, every chromosome is a string of numbers, which represents number in a sequence.

Example: Chromosome A: 1 5 3 2 6 4 7 9 8

Value - Direct value encoding can be used in problems, where some complicated values are used. In value encoding, every chromosome is a string of some values. Values can be anything connected to problem e.g. real numbers or chars to some complicated objects.

Example: Chromosome A: 1.2324 5.3243 0.4556 2.3293 2.4545

Chromosome B: (back), (back), (right), (forward), (left)

Tree - Tree encoding is used mainly for evolving programs or expressions, for genetic programming. In tree encoding every chromosome is a tree of some objects, such as functions or commands in programming language.

Example:

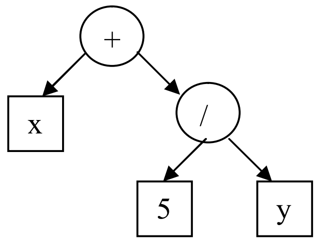


Figure 5.2: Tree encoding for equation: (+ x (/ 5 y))

The choice of how to encode solutions on a chromosome is of primary importance to the success of the genetic algorithm approach to a problem. The encoding of information on the chromosome should be right for the problem rather than specific to the problem. The encoding should be able to represent all the relevant parameters of the problem and should avoid other parameters. Using parameters that are not directly relevant will cause the genetic algorithm to be subject to changes in the problem that would not otherwise affect it, thereby making it no more useful than a specialized heuristic. Some knowledge of the search space is, of course, unavoidable according to Rawlins, (1991).

5.1.2.2 Fitness Function

A function is needed that will interpret the chromosome and produce an evaluation of the chromosome's fitness. This function must be defined over the set of possible chromosomes and is assumed to return some positive value representing the fitness. The definition of this function is crucial because it must accurately measure the desirability of the features described by the chromosome. In addition, the function must make this evaluation in a very efficient manner because of the large number of times the function will be called during the execution of the genetic algorithm. For example, with a population of 100 chromosomes that runs for 1000 generations, there could be as many as 100,000 calls to this evaluation function during execution.

5.1.2.3 Choosing an Initial Population

In a "pure" genetic algorithm, the initial population is chosen randomly, with the goal of selecting chromosomes from all over the search space. Whatever genetic material is in the initial population will be the only material, except for the rare changes due to mutation, available to the genetic algorithm during its search. One might employ a heuristic to choose the initial population in an attempt to introduce the “right” genetic building blocks into the population. However, this can lead to problems of premature convergence to a local optimum since genetic algorithms are “notoriously opportunistic” Grefenstette (1987).

5.1.2.4 Reproduction Operators

According to GA outline, Parent Selection Mechanism is a prerequisite for reproduction operations: Crossover and Mutation.

Parent Selection Mechanism: The role of parent selection is to distinguish among individuals based on their quality to allow the better individuals to become parents of the next generation. Parent selection is probabilistic. Thus high quality individuals get a higher chance to become parents than those with low quality. Nevertheless, low quality individuals are often given a small but positive chance; otherwise the whole search could become too greedy and get stuck in a local optimum. Some of the selection methods are described below:

I. *Roulette Wheel Selection* - Parents are selected according to their fitness. The better the chromosomes, the more chances to be selected. Imagine a roulette wheel where every chromosome in a population has its place big accordingly to its fitness function. This can be simulated by following algorithm.

Roulette Wheel Selection procedure:

1. *[Sum]* Calculate sum of all chromosome’s fitness in population - sum *S*.
2. *[Select]* Generate random number from interval *(0,S)* - *r*.
3. *[Loop]* Go through the population and sum fitness from *0* - sum *S*. When the sum *S* is greater than *r*, stop and return the chromosome where you are.

Step 1 is performed only once for each population.

II. *Rank Selection* - Rank selection first ranks the population and then every chromosome receives fitness from this ranking. The worst will have fitness *1*, second worst *2* etc. and the best will have fitness *N* (number of chromosomes in population).

III. *Steady-State Selection* - Main idea of this selection is that big part of chromosomes should survive to next generation. In every generation a few (good - with high fitness) chromosomes are selected for creating a new offspring. Then some (bad – having low fitness) chromosomes are removed and the new offspring is placed in their place. The rest of population survives to new generation.

Crossover Operator: This operator merges information from two parents' genotype into one or two offspring genotypes. Crossover is a stochastic operator: the choice of what parts of each parent are combined and the way these parts are combined depend on random drawings. The principle behind crossover is simple: by mating two individuals with different but desirable features, an offspring can be produced which combines both of these features. These are different kinds of crossover:

I. *One-point crossover* - One crossover point is selected. String from beginning of chromosome to the crossover point is copied from one parent; the rest is copied from the second parent.

Example: **11001**011+11011**111** = **11001111** & 11011011

Parent1 + Parent 2 = Child1 & Child 2

II. *Two point crossover* - Two crossover points are selected. String from beginning of chromosome to the first crossover point is copied from one parent, the part from the first to the second crossover point is copied from the second parent and the rest is copied from the first parent.

Example: **11**0010**11** + 11**0111**11 = **11011111** & 11001011

Parent1 + Parent 2 = Child1 & Child 2

III. *Uniform crossover* – Genes are randomly copied from the first or from the second parent.

Example: 1**10**010**11** + **1**10**111**01 = **11011111** & 11011111 (random)

Parent1 + Parent 2 = Child1 & Child 2

Mutation Operator: A unary variation operator is called mutation. It is applied to one genotype and delivers a modified mutant. In general, mutation is supposed to cause a random unbiased change. Mutation has a theoretical role; it can guarantee that the space is connected.

Example: 1**1**001001 = 1**0**001001 (2^{nd} bit is inverted).

5.1.2.5 Genetic Algorithm Control Parameters

There are other parameters that govern the genetic algorithm search process. Some of these are:

I. *Population size* - Determines how many chromosomes, and therefore how much genetic material, are available for use during the search. If there is too little, the search has no chance to adequately cover the space. If there is too much, the genetic algorithm wastes time evaluating chromosomes.

II. *Generations* - Specifies how many times the population will be replaced through reproduction.

III. *Crossover Rate* - Specifies the probability of crossover (mating) occurring between two chromosomes.

IV. *Mutation Rate* - Specifies the probability that a value in the chromosome of a newly created offspring will be randomly changed.

V. *Termination Condition* - GA is stochastic process and mostly there are no guarantees to reach a global optimum. Commonly used conditions for terminations are the following:
   - A solution is found that satisfies minimum criteria.
   - Fixed number of generations reached.
   - For a given number of generations, there is no improvement in fitness.

5.1.3 Constraints Handling in GA

There are many ways to handle constraint in a GA. At the high conceptual level it can be distinguished in two cases: Indirect constraint handling and direct constraint handling. Indirect constraint handling means to incorporate them in the fitness function *f(x)* such that *f(x)* optimal implies that the constraints are satisfied. Direct constraint handling means that the constraint stays as they are and the GA is 'adapted' to enforce them. Direct and Indirect both can be used together in a single application.

5.1.3.1 Direct Constraint Handling

Treating constraint directly implies that violating them is not reflected in the fitness function, thus there is no bias towards chromosome satisfying them. Therefore the population will become less and less feasible with respect to these constraints. This means feasibility of the chromosomes have to be monitored and maintained. The basic problem in this case is that the regular operators are blind to constraints, mutating one or crossing over two feasible chromosomes can result in infeasible offspring. Typical approaches to handle constraints directly are the following:

I. Eliminating infeasible solution
II. Repairing infeasible solution
III. Preserving feasibility by special operators
IV. Decoding, i.e., the search space.

Eliminating infeasible solution is very inefficient, and therefore hardly applicable. Repairing infeasible candidates requires a repair procedure for the chromosome. Preserving feasibility can be NP-Complete. Finally decoding can be seen as shifting to a search space that is different than the original problem formulation.

5.1.3.2 Indirect Constraint Handling

In the case of indirect constraint handling, the optimization objectives replacing the constraints are viewed as penalties for constraint violation hence to be minimized. In general penalties are given for violated constraints. Advantages of indirect constraint handling are:

- Reproduction of the problem to simple optimization.
- Possibility of embedding user preferences by means of weights.

Disadvantages of indirect constraint handling are:

- Loss of information packing everything in a single number.
- Does not work well with sparse problem.

5.1.4 Reasons to choose GA

Genetic algorithm is a parallel, stochastic search process. It is widely used in many applications due to the following reasons:

1. The search is highly parallel, with each population member defining many different possible search directions. Potentially, GA search could be implemented extremely efficiently on massively parallel hardware.
2. No special information about the solution surface such as gradient or local curvature need to be identified. The objective function need not to be smooth, continuous or unimodal.
3. Genetic algorithms have proved to be fairly robust under varying parameter settings and problem particulars. As long as solutions with similar encodings do not have highly variant objective function values, genetic algorithms usually find near optimal solutions.
4. Being a population-based approach, GAs are well suited to solve multi-objective optimization problems. A generic single-objective GA can be modified to find a set of multiple non-dominated solutions in a single run.

Since the optimization of multi-objective unequal-area FLPs have to do with the vast number of possible physical layouts, and with the existence of many locally optimal layouts as well as should capture the pareto front, therefore GA is used in this thesis work.

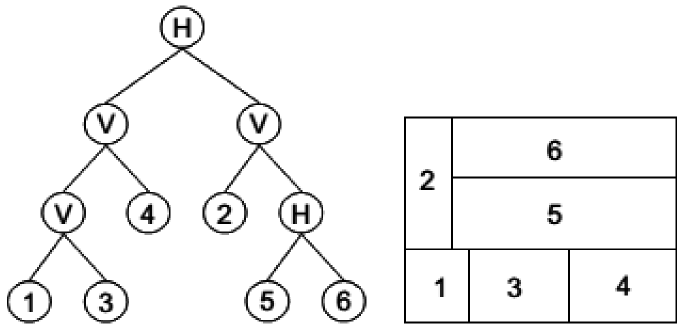
5.2 Slicing Tree and Slicing Structure

A slicing structure results from dividing an initial rectangle either in horizontal or vertical direction completely from one side to the other (so-called guillotine cut) and recursively going on with the newly generated rectangles (Scholz et al., 2010). An equivalent representation of a slicing structure is a slicing tree. A slicing tree is a binary tree which is used to represent such a slicing structure.

For a layout problem of n departments, the slicing tree consists of *n* leaf nodes and *n-1* internal node. Each leaf node symbolizes a site for a department, and each internal node contains information about the direction of a guillotine cut (H:

horizontal, V: vertical). Each internal node corresponds to the area in the layout of the rectangle enfolding both of its children.

Since the procedure must allocate enough space to each department, it is assumed that the total usable area is at least as large as the total area of the departments combined. To enforce the area requirements, the location where a rectangular partition is cut, called the cut point, must be decided so that the split partitions receive the area desired. The cut point is determined in a top-down and left-right fashion starting from the root of the tree and passing down to the children nodes recursively. If the dimensions of a node (i.e., a partition) and the areas of its children are known, the cut point can be located in a straightforward manner, provided the partition does not contain any occupied areas. A slicing tree and its corresponding slicing structure are shown in Fig. 5.3.



(a) Slicing tree (b) Slicing structure

Figure 5.3: Slicing tree and slicing structure

In the above figure, the slicing direction of the root node is horizontally. Therefore, the first cut is dividing the layout completely from left to right. In the layout the departments of the right sub-tree are placed above those of the left one. The right sub-tree start with a vertical cut, and consequently department 2 is arranged on its left hand side and departments 5 and 6 on the right hand side. With the left sub-tree it is the same, but there are two consecutive cuts in the same direction, so departments 1, 3, and 4 are placed side by side.

Every slicing structure can be represented by a slicing tree and vice versa, but there can be multiple slicing trees corresponding to the same slicing structure. For example, the layout from Fig. 5.3(b) is also represented by the slicing tree shown in Fig. 5.4, in which the sequence of consecutive cuts in the same direction has been changed. Wong and Liu (1986); Wong and Liu (1989) addressed a slicing tree a skewed slicing tree if no internal node and its right child possess the same orientation. The following Fig. 5.4 and Fig.5.5 represents a non-skewed slicing tree and skewed slicing trees respectively.

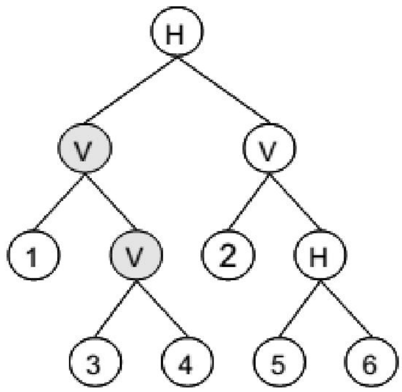


Figure 5.4: Non-skewed slicing tree

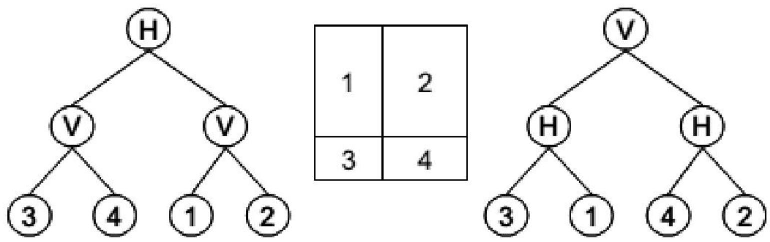


Figure 5.5: Slicing structure and corresponding (skewed) slicing trees

But even when using only skewed slicing trees, some layouts are represented by multiple slicing trees. This occurs if vertical and horizontal cuts generate a 'cross', like in Fig. 5.5. This may be some kind of special case, depending on the exact department dimensions, but should be kept in mind even if Wong and Liu (1986), Wong and Liu (1989) stated a one-to-one correspondence between slicing structures and skewed slicing trees.

Slicing tree approaches are restricted in such a way that only slicing structures can be represented. An example of a non-slicing structure is the wheel structure shown in Fig. 5.6.

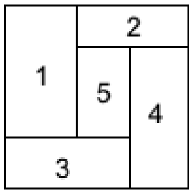


Figure 5.6: Wheel structure

This slicing structure restriction seems to be a hard restriction but it provides some benefits when applied to unequal-area facility layout problems. This slicing structure helps in handling the usual constraints for us:

- Departments must be within the total facility premises.
- Departments must not overlap with each other.
- Department's area constraints.

The STS has gained popularity in developing genetic algorithms for facility layout problems. Existing implementations based on STS mostly require repairing procedures (section 5.1.3.1) to ensure that the chromosomes represent feasible layouts after application of genetic operators (e.g., Ripon et al., 2011).

5.3 Model Optimization using GA based on STS

Since the developed facility layout model is multi-objective optimization problem, therefore a multi-objective genetic algorithm known as Random Weighted Genetic Algorithm (RWGA) is used to optimize the model. The RWGA was proposed by Murata and Ishibuchi (1995); Murata et al., (1996). Based on a weighted sum of multiple objective functions where a normalized weight vector w_i is randomly generated for each solution x_i during the selection phase at each generation. This approach has the advantage over the weighted sum approach which is this approach aims to stipulate multiple search directions in a single run without using additional parameters. Therefore, it is much easier to find to find a set of multiple non-dominated solutions in a single run which cannot be obtained in a weighted sum approach.

Again, in thesis work slicing tree representation is used for the facility layout problem. This slicing tree structure (STS) has been embedded in the RWGA. Thus, a customized RWGA based on STS has been developed and employed to optimize the developed facility layout model. The general procedure of the customized RWGA based on STS employed in optimizing the developed model is given as follows:

Step 1: Generate a random population.

Step 2: Assign a fitness value to each $x \in P_t$ solution by performing the following steps:

Step 2.1: Generate a random number u_k in [0,1] for each objective *k, k=1,2,3,...K*.

Step 2.2: Calculate the random weight of each objective k as $w_k = u_k / \sum_{i=1}^{k} u_i$

Step 2.3: Calculate the fitness of the solution as $f(x) = \sum_{k=1}^{K} w_k \, z_k(x)$. Save pareto-optimal solution.

Step 3: Select parents using Roulette wheel approach

Step 4: Apply Crossover and Mutation on the selected parents (use Crossover rate and Mutation rate)

Step 5: If necessary, apply Refactoring to the newly generated child. Replace old population by this newly generated one.

Step 6: If termination condition is not satisfied go to step 2. Otherwise, return the Pareto-front and the chromosome containing the optimized solution found so far.

The complete solution approach can be described in brief by the following flow chart shown in Fig. 5.7

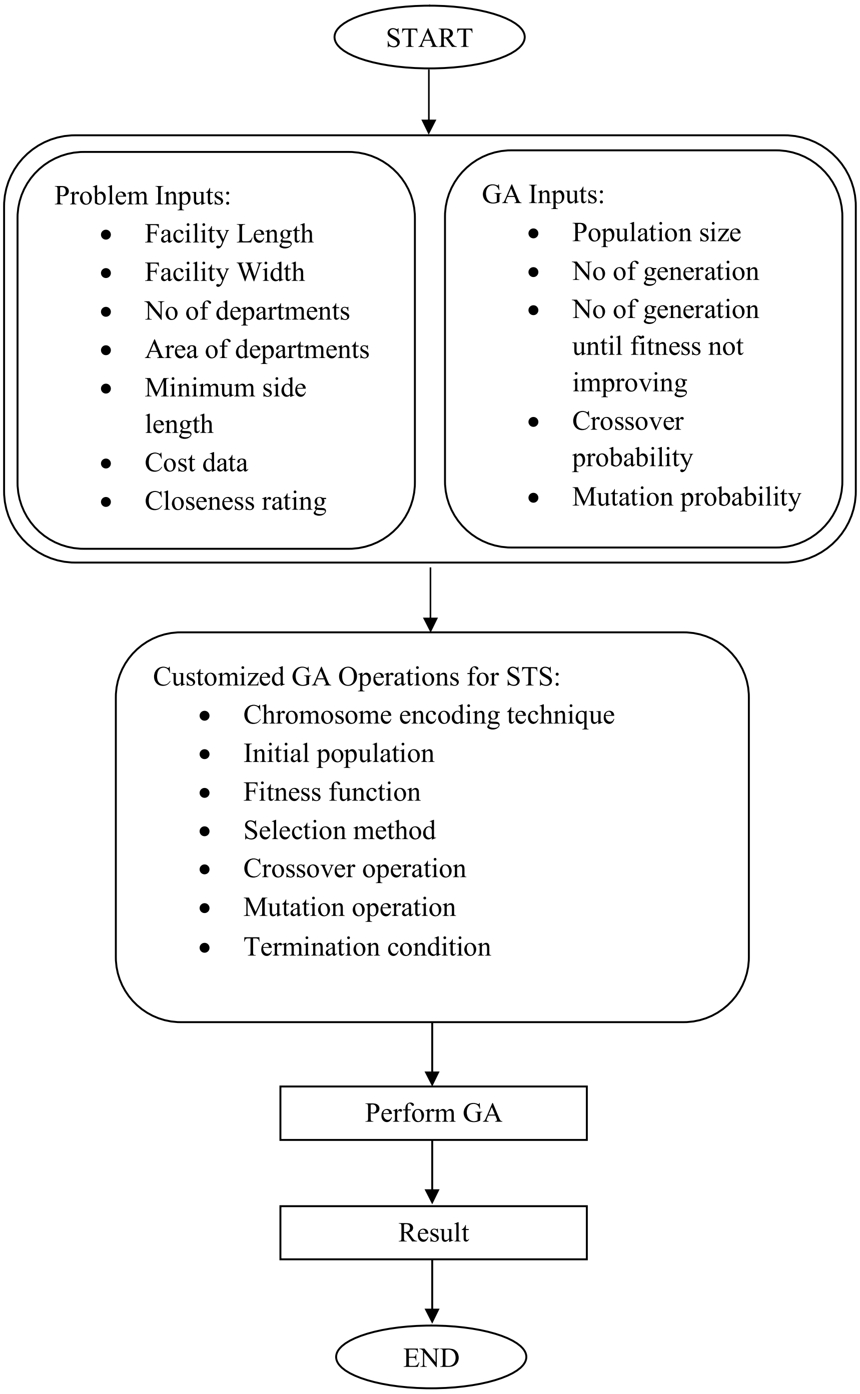


Figure 5.7: Flow chart of GA optimization procedure

Solving with GA based on STS requires some customized chromosome structure, fitness measurement mechanism, selection procedure, reproduction operations and a repairing procedure. The brief description of the solution process is outlined below:

5.3.1 Chromosome Encoding

The encoding method in the natural system is regarded as chromosomes. In the artificial system, it is a string of genes, coded by fixed length binary values (0, 1) or alphabetical characters (A, B, C) or numeric numbers (1, 2, 3). In this section, a chromosome representation suitable for STS of UA-FLP is used. The chromosomes are encoded as (d_1 d_2 d_3…… d_N)(so_1 so_2 so_3…... so_{N-1}) (ns_1 ns_2 ns_3…... ns_{N-1}), where N is the number of departments; and d, so, and ns represents departments sequence, slicing orientation and node selector respectively. In general, the slicing tree representation recursively divides the total floor area (horizontally or vertically), in proportion to the areas of the departments. The first part of the chromosome is represented by integer numbers, whereas the last two parts are represented by either 1 or 0. The department sequence will be transformed into a slicing tree. The slicing orientation 0 represents a horizontal cut and 1 represents a vertical cut. In the node selector, 0 represents an internal node and 1 represents a leaf (department). A chromosome for a 7 department problem is presented in Fig. 5.8. Figure 5.9 presents the corresponding solution representation, transformation into slicing tree, and the layout solution for this chromosome.

2	4	7	5	6	1	3	**0**	**1**	**1**	**1**	**0**	**0**	*0*	*0*	*0*	*1*	*0*	*1*	*1*	*1*	*0*	*0*	*1*	*1*	*1*

Figure 5.8: Chromosome representation for 7- department problem

Department sequence

2	4	7	5	6	1	3

Slicing orientation

0	**1**	**1**	**1**	**0**	**0**

Horizontal - 0

Vertical - 1

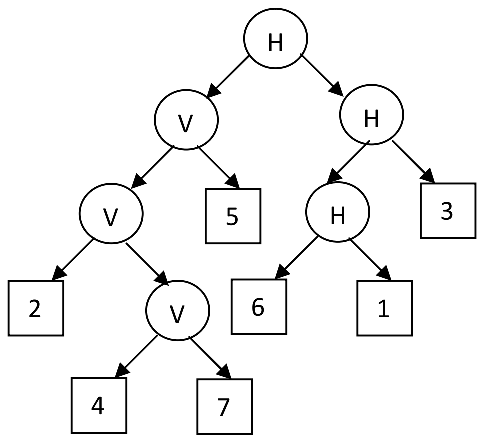
Node selector

0	*0*	*0*	*1*	*0*	*1*	*1*	*1*	*0*	*0*	*1*	*1*	*1*

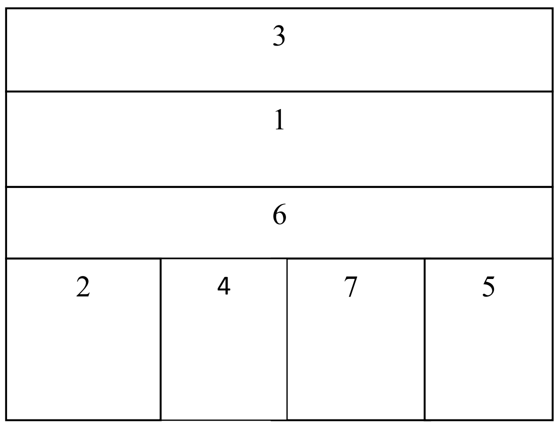
Internal Node (V or H) - 0

Leaf (Departments) - 1

(a) Solution representation



(b) Slicing tree



(c) Facility layout

Figure 5.9: Solution representation, slicing tree and facility layout for the 7-department problem.

5.3.2 Fitness Function

Since, the RWGA employed in optimizing the developed multi-objective model is based on the weighted sum of the multiple objectives functions; therefore, the fitness of each solution is calculated as the weighted sum of the objectives. The fitness function is can be expressed as:

$$\text{Min } Z = w_1 Z'_1(x) + w_2 Z'_2(x) + \cdots + w_k Z'_k(x)$$

Where, $Z'_i(x)$ is the normalized objective function $Z_i(x)$ and w_i is randomly generated normalized weight for each chromosome where $\sum w_i = 1$

The actual fitness function for the developed model is Eq. (3.16) as shown below:

$$Minimize\ Z = w_1 \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} C_{ij}\{|x_i - x_j| + |y_i - y_j|\} - w_2 \sum_{i=1}^{n-1} \sum_{j=i+i}^{n} R_{ij} \times p_{ij} + w_3 \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} t_{ij}\{|x_i - x_j| + |y_i - y_j|\}$$

5.3.3 Initial Population

Anderson and Ferris (1994) mentioned that the performance of GA is better in random start than any preselected starting population. That's why, to determine an initial population, *R* random chromosomes are generated with valid slicing tree and slicing structure. Every slicing tree corresponds to a layout with non-overlapping departments which meet their geometric requirements.

5.3.4 Parent Selection

Roulette Wheel Selection is employed here for parent selection. Pseudo code and a short description on Roulette Wheel Selection have been described in section 5.1.2.4. In short this approach ensures that chromosome with high fitness value can participate more in reproduction system.

5.3.5 Crossover Procedure

In this approach, 3-point crossover has been applied for performing the crossover operation. For keeping the chromosome valid after the operation, these 3 points are chosen randomly from each segment of a chromosome. However, for the first and last segments (department sequence and node selector), some repair works are required after the crossover to remove any duplication or absence of departments. In the repair work for the first segment, list of distinct departments available in each parent are detected. By using list of all departments (from 1 to N), absent department's list are then extracted. After that a merging between these two lists is done. Repair work for the last segment is simpler. Number of '1' (leaf) present in this segment has to be equal to N (number of departments). All these procedure have to be done for each damaged child. Figure 5.10 depicts the crossover operation.

P1:	2	4	7	5	6	1	3	**0**	**1**	**1**	**1**	**0**	**0**	*0*	*0*	*0*	*1*	*0*	*1*	*1*	*1*	*0*	*0*	*1*	*1*	*1*
P2:	1	6	5	2	4	7	3	**1**	**0**	**1**	**1**	**0**	**1**	*0*	*0*	*0*	*1*	*1*	*1*	*0*	*0*	*1*	*1*	*0*	*1*	*1*

(a) Parent chromosomes before crossover.

C1:	2	4	7	5	4	7	3	**0**	**1**	**1**	**1**	**0**	**1**	*0*	*0*	*0*	*1*	*0*	*1*	*0*	*0*	*1*	*1*	*0*	*1*	*1*
C2:	1	6	5	2	6	1	3	**1**	**0**	**1**	**1**	**0**	**0**	*0*	*0*	*0*	*1*	*1*	*1*	*1*	*1*	*0*	*0*	*1*	*1*	*1*

(b) Child chromosomes after crossover without repair.

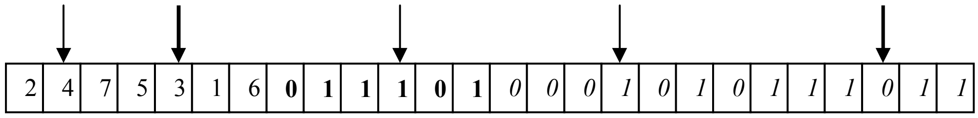
C1:	2	4	7	5	3	1	6	**0**	**1**	**1**	**1**	**0**	**1**	*0*	*0*	*0*	*1*	*0*	*1*	*0*	*1*	*1*	*1*	*0*	*1*	*1*
C2:	1	6	5	2	3	4	7	**1**	**0**	**1**	**1**	**0**	**0**	*0*	*0*	*0*	*1*	*1*	*1*	*0*	*1*	*0*	*0*	*1*	*1*	*1*

(c) Child chromosome after repair.

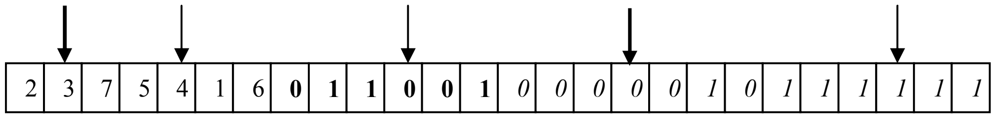
Figure 5.10: Crossover operation

5.3.6 Mutation Procedure

To apply mutation, swap mutation with the restriction that both genes will be chosen from the same segment has been implemented. As a result, no repair work is necessary for mutation. This choice of swapping genes will be random for every chromosome of the population pool. Simple inversion is applied on the middle segment's gene. Figure 5.11 gives an example for the mutation.



(a) Before mutation



(b) After mutation

Figure 5.11: Mutation operation

5.3.7 Termination Condition

In this algorithm two different stopping criterions have been set. They are as follows:

1. Number of generation
2. Number of generation fitness not improving

This program will terminate if any of these conditions whichever occurs earlier.

CHAPTER IV

LAYOUT MODEL USING BRANCH-AND-BOUND ALGORITHM

The unequal-area facility layout problems are among the class of hardest optimization problems. Again, when the mathematical model of such a problem becomes more complex, reaching a solution becomes more difficult. The optimization problems can be solved using either exact algorithms or meta-heuristics algorithms. Since it is computationally easier and not much time consuming, often researchers approach the problems with meta-heuristics. But they do not always guarantee the global optimality. On the other hand, exact deterministic methods have the advantage that they provide a rigorous guarantee of global optimality of any solution produced within a specified €-tolerance (Smith and Pantelides, 1999). Again, optimization problems can be of two types: constrained and unconstrained. The constrained optimization problems require that the parameters and variables must satisfy some conditions. Therefore, these types of problems are much harder to solve compared to the constrained optimization problems. This thesis work deals with a constrained optimization problem.

A relatively new and effective technique for optimizing the convex or non-convex mixed integer nonlinear programming problems is reformulation based spatial branch and bound algorithm and it guarantee global optimal solution for both convex and non-convex MINLPs. However, so far in the literature, it has not used in solving facility layout problems. Therefore, in this thesis work, a reformulation based branch and bound algorithm has been employed to solve the developed constrained mixed-integer nonlinear facility layout model. In the following sections the basic optimization procedures, core concepts and application of spatial branch and bound algorithm to this facility layout model has been discussed.

4.1 Reformulation-Based Spatial Branch-and-Bound Algorithm

A reformulation-based spatial branch and bound algorithm is an efficient method in determining the optimal solution of a mixed integer nonlinear programming (MINLP) problems. This algorithm is a hybrid of the two algorithms: A symbolic reformulation algorithm and a spatial branch and bound algorithm. Though the spatial branch and bound algorithm can solely yield the optimization of MINLPs, it was combined with a reformulation algorithm. Because there are some shortcomings of spatial branch and bound algorithm in some cases. One of the criticisms of spatial branch-and-bound algorithms for global optimization is that they tend to be applicable to limited classes of problems. This arises from the need to construct the convex relaxations of the objective function and constraints. Tight convex relaxations had already been proposed for several special algebraic forms, such as bilinear and linear fractional terms (McCormick, 1976; Quesada & Grossmann, 1995). Simple exponentiation and convex/concave univariate function terms (Smith, 1996) and logarithmic mean temperature difference terms (Zamora & Grossmann, 1997b). However, not all general process engineering problems are naturally expressed using solely this set of nonlinear terms. The research by Floudas and co-workers has led to techniques which allow the formation of the convex relaxation for any continuous twice differentiable functions (Androulakis et. al.,1995; Adjiman and Floudas, 1996), with extensions for problems containing binary variables (Adjiman et al., 1997). However, these relaxations are not generally as tight as the specific ones mentioned above. Smith and Pantelides (1999) proposed a symbolic reformulation based spatial branch and bound algorithm.

4.1.1 Symbolic Reformulation Algorithm

This symbolic reformulation algorithm was proposed by Smith and Pantelides (1999). This algorithm can be used to reformulate the original MINLP problem to an equivalent problem which contains only linear constraints and special nonlinear term definitions. This algorithm is based on the simple observation that any algebraic expression is made up of binary operators corresponding to the five basic operations of arithmetic (addition, subtraction, multiplication, division and exponentiation) and unary operators corresponding to a relatively small number of transcendental functions of a single variable (e.g. logarithms, exponentials etc.). Consequently, if it is possible to construct a convex relaxation of simple terms corresponding to these binary or unary operations, then it must, in principle, also be possible to construct convex relaxations of any algebraic expression.

The author illustrated the reformulation by an example- they considered the mass balance equation on a component A taking part in a second order reaction 2A→B in a well-stirred non-isothermal system. Typically, this would be of the form:

$$F_{in}C_{A,in} - F_{out}C_A - 2Ae^{-E/RT}C_A^2\,V = 0 \tag{4.1}$$

Where F_{out}, C_A and T are variables while the rest of the symbols denote constants. This equation can be reformulated in terms of the linear constraint:

$$F_{in}C_{A,in} - w_1 - 2AVw_5 = 0 \tag{4.2}$$

and the additional nonlinear relations:

$$w_1 = F_{out}C_A \tag{4.3}$$

$$w_2 = \frac{-E}{RT} \tag{4.4}$$

$$w_3 = e^{w_2} \tag{4.5}$$

$$w_4 = C_A^2 \tag{4.6}$$

$$w_5 = w_3w_4 \tag{4.7}$$

It was noted that this has involved the introduction of five auxiliary variables w_1, i= 1,…,5 which are related to each other and to the original variables via the nonlinear relations Eqs.(4.3)-(4.7). All of the latter belong to special algebraic forms for which convex relaxations are available.

The above reformulation was easily obtained by inspection. In fact, as shown by Smith and Pantelides (1996), it is possible to completely generalize and automate the reformulation procedure by employing the standard binary tree representation of algebraic expressions. This type of representation for the expression on the left hand side of Eq. (4.1) is shown in Fig. 4.1.

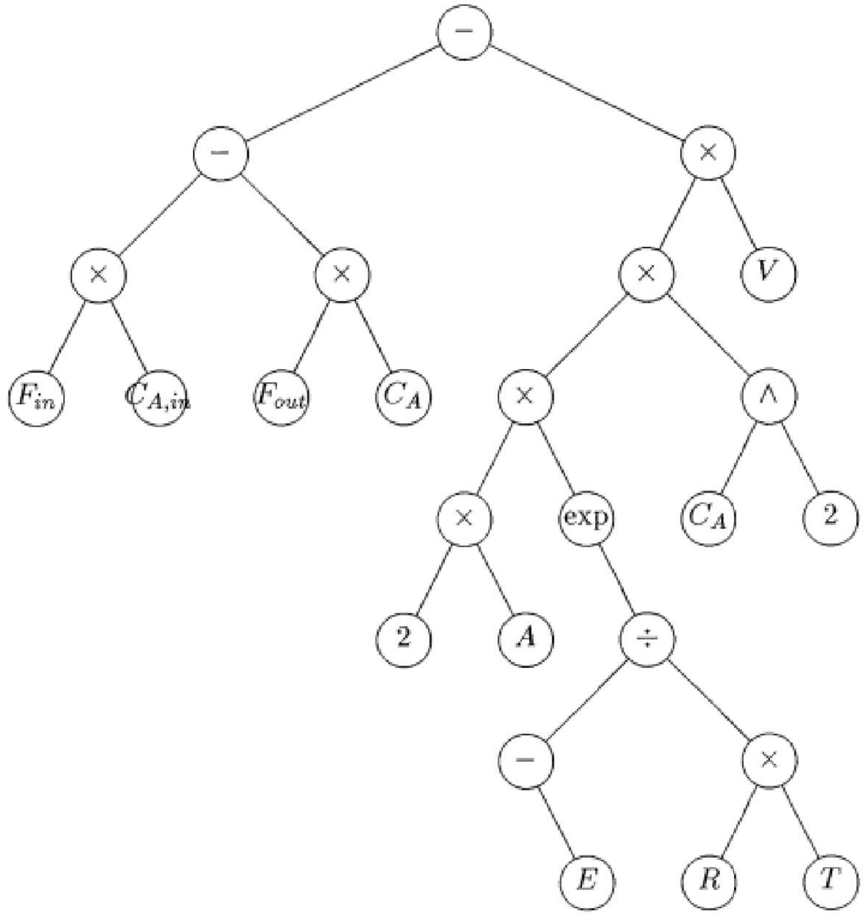


Figure 4.1: Binary tree representation of algebraic expressions

The procedure produces the reformulation that involves the introduction of the minimum number of new variables. For instance, it can be seen that Eqs. (4.3), (4.4), (4.5), (4.6) and (4.7) involve five new variables despite the fact that the binary tree of Fig. 4.1 comprises 13 operator nodes. Moreover, the procedure detects identical nonlinear expressions occurring in different constraints in the problem and introduces the necessary variables once only; for instance, the last term of Eq. (4.1) is likely to occur repeatedly in the mass balance equations for all components, as well as in the energy balance. However, the procedure would introduce the auxiliary variables $w_2 \ldots w_5$ required for the reformulation of this term once only.

The author stated that, using the above reformulation procedure, any process optimization problem of the following MINLP form can be converted to the standard form.

$$\text{Min } f(x, y)$$

Subject to,

$$g(x, y) = 0$$

$$h(x, y) \leq 0$$

and

$$x^l \leq x \leq x^u$$

$$y \in [y^1, \ldots, y^u]$$

where x is a vector of bounded continuous variables and y is a vector of integer variables allowed to take values between the integer bounds y^l and y^u. No restrictions are imposed on the functional form of the objective function $f(x, y)$ or the constraints $g(x, y)$ and $h(x, y)$.

Using the proposed reformulation procedure, the above MINLP form can be converted to the following standard form:

$$\text{Min } w_{obj}$$

Subject to,

$$Aw = b$$

$$w^l \leq w \leq w^u$$

$$y \in [y^1, \ldots, y^u]$$

$$w_k \equiv w_i w_j \quad \forall \, (i, j, K) \in T_{bt}$$

$$w_k \equiv \frac{w_i}{w_j} \, \forall \, (i, j, K) \in T_{Ift}$$

$$w_k \equiv w_i^n \, \forall \, (i, j, n) \in T_{et}$$

$$w_k \equiv fn(w_i) \forall \, (i, k) \in T_{uft}$$

where the vector of variables w comprises the original continuous variables x and discrete variables y, as well as slack variables that have been introduced to convert the inequality constraints in the original form to equalities and any auxiliary continuous variables introduced during the reformulation. A and b are respectively a matrix and vector of real coefficients. In the latter, all the nonlinearities (and potential nonconvexities) were described by the sets corresponding to bilinear product, linear fractional, simple exponentiation and univariate function terms respectively.

The above reformulation is exact, i.e. MINLP problem in the original form and after reformulation are completely equivalent. Therefore, applying a spatial branch-and-bound to the standard form will locate a global optimum which is also the global optimum for the original MINLP problem.

4.1.2 Spatial Branch-and-Bound Algorithm

The spatial branch and bound algorithm used in this thesis work was proposed by Smith and Pantelides (1999). The main steps are outlined below:

Step 1: Initialize search

Set $f^u = \infty$ and list of search region (L) to the single domain $R = [w^l, w^u]$.

Step 2: Initial bounds tightening

Tighten the variable bounds in region $R = [w^l, w^u]$. If the lower bounds for one or more variables exceed the corresponding upper bounds, go to step 9.

Step 3: Choose a subregion

If L=Ø, go to step 9, otherwise choose a subregion R from the list of regions L such that: $R = \arg\min f^{R,l}$.

Step 4: Bounds tightening in R

Tighten the bounds for subregion R. If the lower bounds for one or more variables exceed the corresponding upper bounds, go to step 8.

Step 5: Objective function lower bound in R

Generate the convex relaxation of the reformulated problem in R. Solve this convex relaxation to determine the objective function lower bound $f^{R,l}$. If the convex relaxation is infeasible or if $f^{R,l} \geq f^{u}$, go to step 8.

Step 6: Objective function upper bound in *R*

Solve (locally) a restriction of the reformulated problem in R to determine the objective function upper bound $f^{R,u}$. If a local minimum cannot be determined or if $f^{R,u} > f^{u}$, go to step 7. Otherwise, set $f^{u} = f^{R,u}$ and delete all subregions from the list L.

Step 7: Partition region *R*

Apply a branching rule to subregion R to choose a branch variable and its corresponding value about which to branch. Create two new subregions R^{L} and R^{U} by partitioning *R* at the branch point. Add these to the list L.

Step 8: Delete region *R*

Delete subregion *R* from the list of regions L. Go to step 3.

Step 9: Terminate search

If $f^{u} = \infty$, the problem is globally infeasible. Otherwise, the global optimum objective function value is f^{u}.

The above algorithm maintains a list L of regions (hyper-rectangles) which are to be searched for the global optimum. It also keeps a record of the incumbent best upper bound f^{u} for the global optimum objective function value. The main iteration loop (step 3 – step 8) processes a single region R from the list and calculates the corresponding lower and upper bounds on the objective function value (steps 5 and 6, respectively). On the basis of these bounds, it either discards the region (step 8) or partitions it into two child regions (step 7) which are then placed on the list for further examination. This steps repeats until convergence to a global optimal solution is reached.

4.2 Model Optimization using Spatial Branch-and-Bound Algorithm

The spatial branch and bound algorithm is an efficient method of finding the global optima for both convex and non-convex MINLP problems. However, to solve the facility layout model developed in this thesis, AMPL/COUENNE solver is used, since the COUENNE works on the reformulation-based spatial branch and bound algorithm discussed in the previous section.

However, solving mathematical model using spatial branch and bound algorithm requires all functions to be smooth functions, i.e, continuously differentiable. But, in the developed model, there are some non-smooth functions. Therefore, an equivalent model was developed by modifying the non-smooth functions by their equivalent mathematical equations which are smooth functions.

4.2.1 Equivalent Facility Layout Model

The objective functions and some constraint equations of the developed model are non-smooth functions. They have been modified by introducing some additional variables to develop equivalent equations of them.

4.2.1.1 Modification of the Objective Functions

1. The first (Eq. 3.1) and third objective (Eq. 3.4) functions have an absolute operator, which makes them non-smooth function. This absolute operator can be replaced by introducing two new variables and a set of equivalent linear equations which will act as constraints. Such linearization was used by Meller et al. (1999). After this modification, the objective functions can be expressed as-

$$\text{MHC} = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} C_{ij}(x_{ij} + y_{ij}) \tag{4.8}$$

$$\text{Makespan} = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} t_{ij}\,(x_{ij} + y_{ij}) \tag{4.9}$$

This modification of the objective functions results the following linear constraints which replaces the absolute operator.

$$x_{ij} \geq x_i - x_j \quad \forall\, i, j \tag{4.10}$$

$$x_{ij} \geq x_j - x_i \quad \forall\, i, j \tag{4.11}$$

$$y_{ij} \geq y_i - y_j \quad \forall\, i, j \tag{4.12}$$

$$y_{ij} \geq y_j - y_i \quad \forall\, i, j \tag{4.13}$$

Therefore, these constraints ensure that, always the positive value of the distance between two departments will be taken in distance calculation.

2. The second objective function (Eq.3.2) involves a binary variable (p_{ij}) for determining the adjacency of the departments. The conditions for p_{ij} to have a value of 1 or 0 is an if-then-else expression which is a non-smooth function. The equivalent linear equations of an if-then expression can be developed using the method from Winston and Venkataramanan (2003), where they proposed a method of replacing the if-then expression using binary variable and a large positive number.

Therefore, two additional binary variables (q_{ij} and z_{ij}) have been introduced to replace the if-then-else expression with equivalent linear equations.

The objective function now becomes-

$$\text{CRS} = \sum_{i=1}^{n-1} \sum_{j=i+i}^{n} R_{ij} \times (p_{ij} + q_{ij}) \tag{4.14}$$

The conditions for the adjacency between the two departments i and j in x or y direction (Eq. 3.3) can now be replaced by the following equivalent linear equations-

$$x_{ij} - 0.5(l_i + l_j) = M(1 - p_{ij}) \;\; \forall\, i, j \tag{4.15}$$

$$y_{ij} - 0.5(w_i + w_j) = M(1 - q_{ij}) \;\; \forall\, i, j \tag{4.16}$$

Where, M is a sufficiently large positive number. Equation (4.15) represents the adjacency in x direction and equation (4.16) represents the adjacency in y direction. If p_{ij} has a value of 1 the departments i and j will have common boundary in x direction whether they will have common boundary in y direction if q_{ij} has a value of 1. But, two departments i and j cannot have common boundary in both x and y direction, because there was a "or" relationship between the two conditions in Eq.(3.3). So, p_{ij} and q_{ij} cannot have a value of 1 simultaneously. This requires another constraint to control the value of the two binary variables.

$$p_{ij} + q_{ij} = (1 - z_{ij}) \ \forall\, i, j \tag{4.17}$$

Here, z_{ij} is also a binary variable. The above equation prevents p_{ij} and q_{ij} having a value of 1 simultaneously. Again, it is also possible that, two departments cannot have a common boundary in either direction while satisfying other constraints. This is also satisfied by Eq.(4.17). Having a value of 1 for z_{ij} ensures that, the departments i and j cannot have a common boundary in either direction.

4.2.1.2 Modification of the Constraint Equations

Some constraint equations of the developed mathematical model contain non-smooth functions. Therefore, equivalent linear equations have been developed to replace the non-smooth functions.

1. The non-overlapping constraints (Eq. 3.9 and 3.10) contain if-then expression which are non-differentiable and also introduce non-convexities to the model. This if-then expression has been replaced by introducing binary variables. This type of formulation was used by Montreuil (1990); McKendall and Hakobyan (2010).

   Two new binary variables $left_{ij}$ and $below_{ij}$ have been introduced in the model to modify the non-overlapping constraints, where-

$$left_{ij} = \begin{cases} 1, if\ the\ department\ i\ is\ left\ of\ department\ j (i.e., x_i + 0.5l_i \le x_j - 0.5l_j) \\ 0, otherwise \end{cases}$$

$$below_{ij} = \begin{cases} 1, & if\ the\ department\ i\ below\ department\ j (i.e., y_i + 0.5w_i \le y_j - 0.5w_j) \\ 0, & otherwise \end{cases}$$

Therefore, the constraints for preventing the overlapping of departments i and j can be mathematically expressed by the following Eqs (4.18) -(4.21):

$$(x_i + 0.5l_i) - (x_j - 0.5l_j) \le M(1 - left_{ij}) \ \forall\, i, j \tag{4.18}$$

$$(x_j + 0.5l_j) - (x_i - 0.5l_i) \le M(1 - left_{ji}) \ \ \forall\, i, j \tag{4.19}$$

$$(y_i + 0.5w_i) - (y_j - 0.5w_j) \le M(1 - below_{ij}) \ \ \forall\, i, j \tag{4.20}$$

$$(y_j + 0.5w_j) - (y_i - 0.5w_i) \le M(1 - below_{ji}) \ \ \forall\, i, j \tag{4.21}$$

Again, a binary constraints need to be developed since the department i cannot be on the left and right of department j simultaneously.

$$left_{ij} + left_{ji} + below_{ij} + below_{ji} = 1 \ \forall i, j \tag{4.22}$$

Equation (4.22) ensures that, each time only one of the binary variables can have value of 1.

2. The equation (Eq. 3.11) for maximum aspect ratio constraints involves max, min operator which is also a non-smooth function. Therefore, to replace this constraint by an equivalent linear equation a limit on the minimum side length has been imposed. Since, the aspect ratio is closely related to the minimum side length of the department, aspect ratio can be controlled via the minimum side length of departments (Jankovits et al., 2011).
   Therefore, minimum side length was specified for each department which in turn will control the maximum aspect ratio. Thus the original aspect ratio constraint in Eq. (3.11) has now been replaced by the following minimum side length constraints.
   $$l_i \geq l_{min} \ \forall i \tag{4.23}$$
   $$w_i \geq l_{min} \ \forall i \tag{4.24}$$

4.2.2 Optimization of the Model using AMPL/COUENNE

After replacing the non-smooth functions with their equivalent linear equations, the equivalent facility layout model is as follows:

$$Minimize\ Z = w_1 \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} C_{ij}(x_{ij} + y_{ij}) - \sum_{i=1}^{n-1} \sum_{j=i+i}^{n} R_{ij} \times (p_{ij} + q_{ij}) + \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} t_{ij}\,(x_{ij} + y_{ij})$$

Subject to,

$$(x_i + 0.5l_i) \leq \mathrm{L} \ \forall i$$

$$(x_i - 0.5l_i) \geq 0 \ \forall i$$

$$(y_i + 0.5w_i) \leq \mathrm{W} \ \forall i$$

$$(y_i - 0.5w_i) \geq 0 \ \ \forall\, i$$

$$(x_i + 0.5l_i) - (x_j - 0.5l_j) \leq M(1 - left_{ij}) \ \ \forall\, i, j$$

$$(x_j + 0.5l_j) - (x_i - 0.5l_i) \leq M(1 - left_{ji}) \ \ \forall\, i, j$$

$$(y_i + 0.5w_i) - (y_j - 0.5w_j) \leq M(1 - below_{ij}) \ \ \forall\, i, j$$

$$(y_j + 0.5w_j) - (y_i - 0.5w_i) \leq M(1 - below_{ji}) \ \ \forall\, i, j$$

$$left_{ij} + \ left_{ji} + below_{ij} + below_{ji} = 1 \ \ \forall\, i, j$$

$$l_i \geq \ l_{i,min} \ \ \forall\, i$$

$$w_i \geq \ w_{i,min} \ \ \forall\, i$$

$$x_{ij} \geq \ x_i - x_j \ \ \forall\, i, j$$

$$x_{ij} \geq \ x_j - x_i \ \ \forall\, i, j$$

$$y_{ij} \geq y_i - y_j \ \ \forall\, i, j$$

$$y_{ij} \geq y_j - y_i \ \ \forall\, i, j$$

$$x_{ij} - 0.5(l_i + l_j) = M(1 - p_{ij}) \ \ \forall\, i, j$$

$$y_{ij} - 0.5(w_i + w_j) = M(1 - q_{ij}) \ \ \forall\, i, j$$

$$p_{ij} + \ q_{ij} = (1 - z_{ij}) \ \ \forall\, i, j$$

$$l_i w_i = a_i \ \ \ \forall\, i$$

$$x_i, y_i, x_j, y_j, l_i, w_i, l_j, w_j, x_{ij}, y_{ij} \geq 0$$

This model has been implemented in AMPL (Fourer et al., 2002) and used COUENNE (Belotti, 2009) solver to optimize the model. The COUENNE is a global optimization solver for both convex and non-convex MINLPs that is a reformulation based branch and bound algorithm which has been discussed in the previous section. After reformulation of the original problem (Sec. 4.1) this solver implements:

Linearization: This linearization step allows obtaining a Linear Programming (LP) relaxation of the reformulated problem which can be easily embedded in a branch-and-bound framework.

Bound Tightening Technique: This technique is used to infer better bounds on all variables (both original and auxiliary), in order to get a tighter lower bound.

Heuristics: A heuristic is used to obtain feasible solution, since finding feasible solution for highly constrained problem is difficult. It is a rounding heuristic that, for each fractional variable x, fixes x at its rounded down value, then runs a fast bound propagation procedure in the hope of fixing other variables or proving the problem is infeasible; then repeats the same for its rounded up value.

Branching Techniques: They are used for partitioning the set of solutions.

Therefore, using these techniques this solver converges to a global optimal solution for MINLPs.

Again, the facility layout model is multi-objective optimization model and thus need multiple pareto solution set. Since, weighted sum approach is used for solving the multi-objective model, this model is solved multiple times using different weight vector. Therefore, to generate a number of random weight vectors containing three normalized weight for the three objectives, a computer program is developed in C# which will provide normalized weight vector so that the sum of a weight vector is equal to 1.

CHAPTER VI

RESULTS AND DISCUSSIONS

In this thesis work a multi-objective unequal area facility layout model has been developed considering some practical criteria in designing facility layout such as qualitative factors as well as makespan of the overall system. The developed constrained multi-objective mixed-integer nonlinear model has been optimized to determine the optimal arrangement of the departments within the facility as well as the optimal dimensions of the departments so that material transportation cost, makespan of the overall system will be minimized and total closeness rating score will be maximized. This model is illustrated with two numerical examples and then optimized using a reformulation-based spatial branch and bound (sBB) algorithm and a random weighted genetic algorithm (RWGA) based on slicing tree structure.

6.1 Numerical Example 1

In this case, a production plant is considered with ten departments of unequal areas. These areas, a_i, in square meters is given in table 6.1. The cost per unit distance of material flow between the departments is given in table 6.2 which depends on the mode of transportation. These data have been taken from Van Camp et al. (1991). A relationship chart has also been developed to represent the closeness preferences between the departments. This closeness rating value is required for calculating the closeness rating score. Material handling time per unit distance between the departments which depends on the mode of transportation is given in table 6.7 which is required to calculate the makespan.

Since the layout was to be developed for an existing facility, the total shape of the plant was constrained to being rectangular, of dimensions 40m×32 m. Thus, a_T = 1280 sq-m. It is assumed that, for a valid layout there is a restriction on the minimum side length of each department (maximum aspect ratio of each department). In solving this model it has been specified that, no department could be narrower than 5m. Again, this minimum side length is also varied within a range to see how it affects the solution, since increasing the minimum side length decreases the aspect ratio and thus makes the problem more constrained and eventually feasible solutions are harder to find.

The following table contains the department areas:

Table 6.1: Departmental areas (sq-m) for the 10 departments

Department	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Area (a_i)	238	112	160	80	120	80	65	85	221	119

Table 6.2: Cost (USD) per unit distance of material flow

Dept.	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
1	-	0	0	0	0	218	0	0	0	0
2		-	0	0	0	148	0	0	296	0
3			-	28	70	0	0	0	0	0
4				-	0	28	70	140	0	0
5					-	0	0	210	0	0
6						-	0	0	0	0
7							-	0	0	28
8								-	0	888
9									-	59.2
10										-

To represent the qualitative factors, closeness relationships are developed among the departments. A numerical value is also assigned for each relationship which actually represents the preference of that relationship. The following table 6.3 shows the reasons for the closeness relationship and the relationship along with their numerical value is shown in table 6.4.

Table 6.3: Different reasons of closeness

Code	**Reason**
1	Ease of supervision
2	Common personnel
3	Contact necessary
4	Sharing common resource
5	Safety

Table 6.4: Closeness relationships with their numerical value

Closeness relationship	Rating	Numerical value
Absolutely necessary	A	5
Especially important	E	4
Important	I	3
Ordinary	O	1
Unimportant	U	0
Undesirable	X	-1

The relationship ranges from A to X, where A is given the highest positive value making it mostly important to be satisfied. The X relationship is given a negative value which means that two departments having X rating should not be close together. The following table presents the relationship chart for the departments.

Table 6.5: The relationship chart for the 10 department problem

Dept.	1	2	3	4	5	6	7	8	9	10
1	-	U	O	U	E	A	U	U	X	U
2		-	U	O	U	I	E	O	U	U
3			-	E	O	X	U	O	U	U
4				-	U	U	I	U	O	A
5					-	I	U	O	O	E
6						-	I	U	A	O
7							-	E	O	I
8								-	A	X
9									-	E
10										-

Now, putting the numerical value of the relationships among the departments, in the above table the closeness rating value is found which is shown in table 6.6.

Table 6.6: Closeness rating value

Dept.	1	2	3	4	5	6	7	8	9	10
1	-	0	2	0	4	5	0	0	-1	0
2		-	0	2	0	3	4	2	0	0
3			-	4	2	-1	0	2	0	0
4				-	0	0	3	0	2	5
5					-	3	0	2	2	-1
6						-	3	0	5	2
7							-	-1	2	3
8								-	5	-1
9									-	4
10										-

For calculating the makespan of the overall system, data of material handling time between the departments is needed which is given in hours as shown in table 6.7.

Table 6.7: Material handling time (Hour) per unit distance

Dept.	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
1	-	0	0	0	0	3	0	0	0	0
2		-	0	0	0	1	0	0	4	0
3			-	4	5	0	0	0	0	0
4				-	0	3	2	1	0	0
5					-	0	0	2	0	0
6						-	0	0	0	0
7							-	0	0	6
8								-	0	4
9									-	5
10										-

Once all the factors have been quantified, the data is normalized which is necessary since all the factors are not represented on the same scale. This normalization of the data has been performed according to Eq. (3.14) as discussed in chapter 3. The normalized data for table 6.2, table 6.6 and table 6.7 appears in the table 6.8, table 6.9 and table 6.10 respectively.

Table 6.8: Normalized cost per unit distance of material flow

Dept.	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
1	-	0	0	0	0	0.0998	0	0	0	0
2		-	0	0	0	0.068	0	0	0.136	0
3			-	0.0128	0.032	0	0	0	0	0
4				-	0	0.0128	0.032	0.064	0	0
5					-	0	0	0.096	0	0
6						-	0	0	0	0
7							-	0	0	0.013
8								-	0	0.407
9									-	0.027
10										-

Table 6.9: Normalized closeness rating value

Dept.	1	2	3	4	5	6	7	8	9	10
1	-	0	0.03	0	0.06	0.076	0	0	-0.015	0
2		-	0	0.03	0	0.045	0.06	0.03	0	0
3			-	0.06	0.03	-0.015	0	0.03	0	0
4				-	0	0	0.045	0	0.03	0.076
5					-	0.045	0	0.03	0.03	-0.015
6						-	0.045	0	0.076	0.03
7							-	-0.015	0.03	0.045
8								-	0.076	-0.015
9									-	0.06
10										-

Table 6.10: Normalized material handling time per unit distance

Dept.	1	2	3	4	5	6	7	8	9	10
1	-	0	0	0	0	0.075	0	0	0	0
2		-	0	0	0	0.025	0	0	0.1	0
3			-	0.1	0.125	0	0	0	0	0
4				-	0	0.075	0.05	0.025	0	0
5					-	0	0	0.05	0	0
6						-	0	0	0	0
7							-	0	0	0.15
8								-	0	0.1
9									-	0.125
10										-

6.1.1 Optimization using Spatial Branch-and-Bound Algorithm

The developed MINLP facility layout model has been implemented in AMPL and solved using COUENE solver which works on reformulation-based sBB algorithm discussed in chapter 4.

The optimal solution obtained is as follows:

Table 6.11: Results obtained from sBB

MHC (USD)	CRS	Makespan (Hour)	Combined objective function value
35133.12	27	664.28	3525.46

For this result, the optimal arrangement of the departments as well as the optimal dimensions of the departments within the facility has also been obtained which are shown in the following table:

Table 6.12: Optimal arrangement and dimensions of the departments

Department	X_Location (m)	Y_Location (m)	Length (m)	Width (m)
1	5.74879	10.35	11.50	20.7
2	4.95575	26.35	9.91	11.3
3	18.1068	6.05	13.22	12.10
4	16.151	16.40	9.31	8.59
5	29.673	6.05	9.92	12.10
6	25.458	16.40	9.31	8.59
7	37.315	6.05	5.37	12.10
8	35.0557	16.40	9.89	8.59
9	19.6903	26.35	19.56	11.3
10	34.7345	26.35	10.53	11.3

Using these centre locations and dimensions of the departments a block layout of the facility is presented in the following figure:

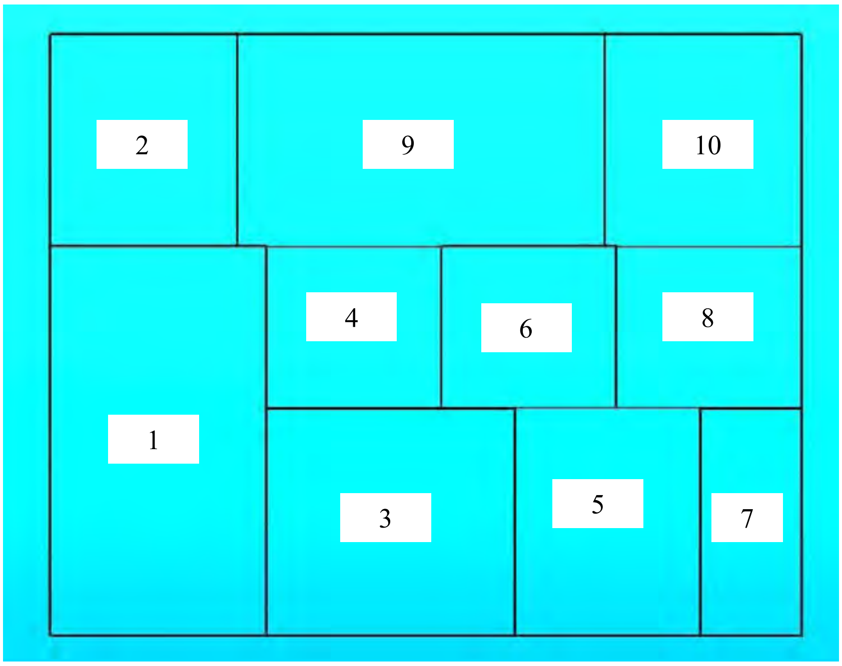


Figure 6.1: Block layout obtained from sBB

From the block layout, it can be seen that, all the constraints of the developed model has been satisfied such as no-overlapping of the departments, bounding the departments within the facility area, departmental area constraints etc.

However, a multi-objective optimization problem (MOOP) differs significantly from the single objective optimization. In a MOOP there can never exists a single absolute solution that can satisfy all the objectives to their best. In case of two or more objectives, each objective corresponds to a different optimal solution, but none of the trade-off solutions is optimal with respect to all objectives. Therefore, in a multi-objective optimization the goal is to generate as many trade-off solutions as possible.

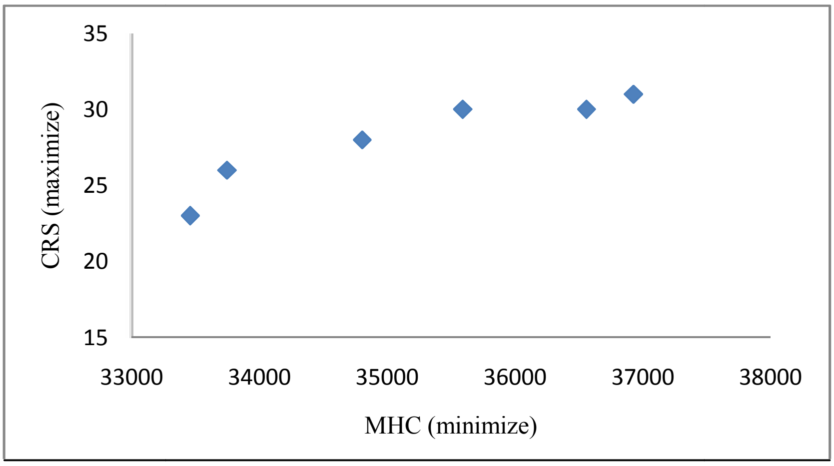
So, though the above optimal solution has the minimum total combined objective function value, it cannot be the best solution with respect to all the objectives simultaneously. Because of the nature of MOOP, this solution may be optimal with respect to one objective or the total fitness becomes minimum by best satisfying all the objectives but may be a poor candidate for a particular objective. Therefore, it is desirable to generate many optimal layouts considering all the three objectives. For this reason, the program has been run several times using different weight vector to obtain multiple trade-off solutions where one of them are not better than another. Some of the pareto optimal solutions obtained in this method are shown in the following table:

Table 6.13: Pareto optimal solutions obtained from sBB

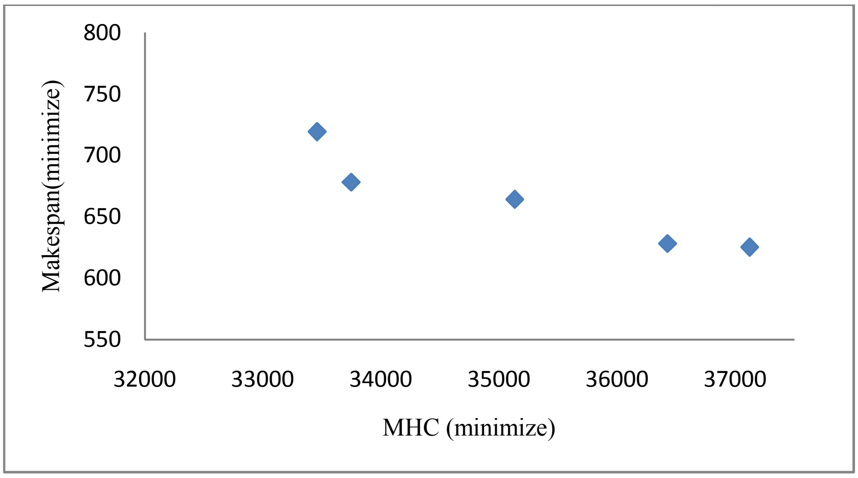
	MHC (USD)	**CRS**	**Makespan (Hour)**
Pareto solution 1	35133.14	27	664.28
Pareto solution 2	33748.18	26	678.13
Pareto solution 3	34805.91	28	685.32
Pareto solution 4	33461.86	23	719.34
Pareto solution 5	36425.12	27	628.06
Pareto solution 6	36928.56	31	724.18
Pareto solution 7	35592.34	30	731.06

From the above table, it is evident that, none of the solution is better than others with respect to all three objectives. Therefore, these pareto optimal solutions are non-dominated. Thus this solution set will be useful for the decision maker as he can selectively choose the most demanding one among all of the generated layouts for a specific order or customer demands.

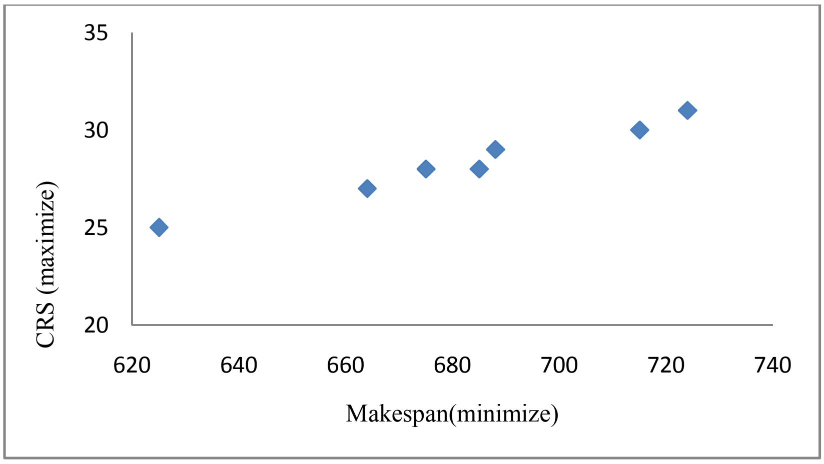
Now, to have a better insight about the pareto optimality of the solutions obtained, the pareto front for the objectives are presented in the following figures.



(a) Pareto front with MHC and CRS



(b) Pareto front with MHC and Makespan



(c) Pareto front with Makespan and CRS

Figure 6.2: Pareto fronts obtained from sBB

From the above figures, it is clearly evident that, the solutions are mostly pareto optimal because none of the objectives can be improved without sacrificing another one. Therefore, these solutions can be useful for decision maker in multi-criteria decision making.

6.1.2 Optimization using Random Weighted Genetic Algorithm

The developed facility layout model has been solved using the proposed slicing tree structure embedded random weighted genetic algorithm. This algorithm is coded with C# and compiled using Microsoft Visual Studio 2012. The computer program has been used to solve the model with different combinations GA parameters. Two generation numbers 1000 and 2000 were tested. Initial population size was varied from 10 to 50; Crossover probability was varied from 0.3 to 0.9; three mutation probabilities was tested: 0.01, 0.1, 0.2, 0.3, 0.5 and 0.7. Therefore, combining these parameters the program has been tested. Another criterion is tested which is number of generations fitness not improving. Three values of this criterion are tested which are: 200, 300 and 500.

Among the combinations of parameters tested, best result has been found for the following parameter combination:

Generation number: 2000

Initial population size: 50

Crossover probability: 0.3

Crossover type: Three point crossover

Mutation probability: 0.01

Number of generation fitness not improving: 300.

The best result obtained from RWGA is shown in the following table:

Table 6.14: Results obtained from RWGA

MHC (USD)	CRS	Makespan (Hour)	Combined objective function value
26013.54	33	656.36	2949.56

For this result, the optimal arrangement of the departments as well as the optimal dimension of the departments within the facility has also been obtained which is shown in the following table:

Table 6.15: Optimal arrangement and dimensions of the departments

Department	X_Location (m)	Y_Location (m)	Length (m)	Width (m)
1	6.05	9.83	12.09	19.68
2	22.47	25.84	9.09	12.32
3	33.51	25.84	12.98	12.33
4	32.76	16.92	14.49	5.53
5	35.76	7.08	8.48	14.15
6	14.79	7.42	5.39	14.83
7	18.81	17.25	13.42	4.85
8	28.52	7.08	6.01	14.15
9	8.97	25.84	17.93	12.33
10	21.50	7.42	8.02	14.83

Using the centre location and dimension of the departments from the optimal solution, a block layout of the facility is presented in the following figure:

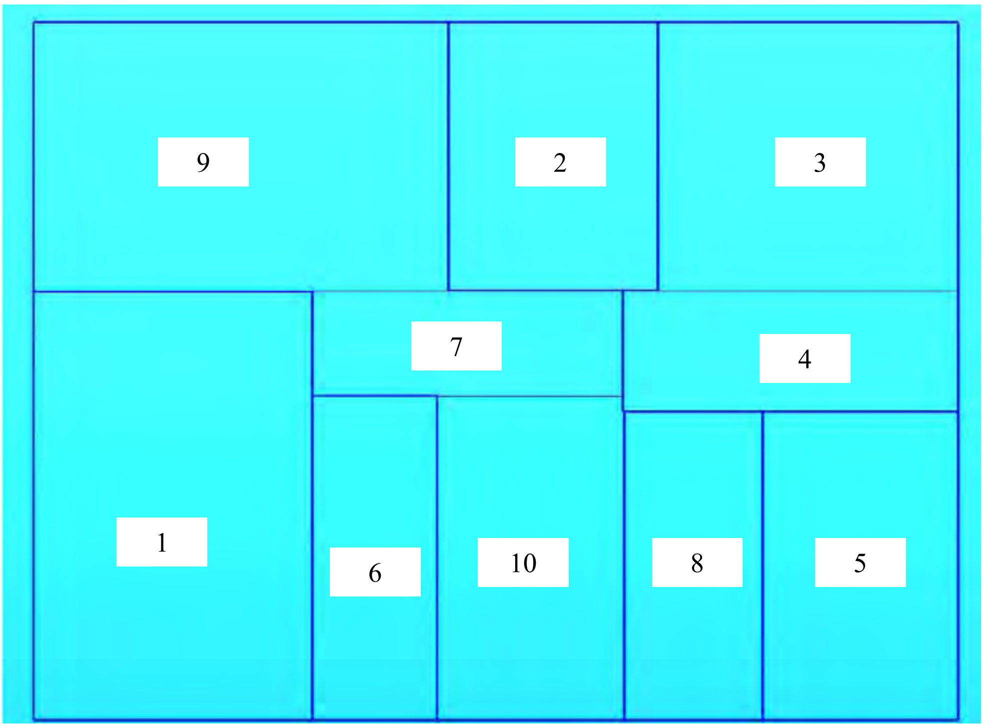


Figure 6.3: Block layout obtained from RWGA

From the block layout, it can be seen that, all the constraints of the developed model has been satisfied such as no-overlapping of the departments, bounding the departments within the facility area, departmental area constraints etc.

Since, in a multi-objective optimization problem (MOOP) there can never exist a single absolute solution that can satisfy all the objectives to their best. For two or more objectives, each objective corresponds to a different optimal solution, but none of the trade-off solutions is optimal with respect to all objectives. Therefore, multi-objective evolutionary algorithms (MOEA)s like RWGA do not try to find one optimal solution but all the trade-off solutions.

Though the above optimal solution has the minimum total combined objective function value, it cannot be the best solution with respect to all the objectives simultaneously. Because of the nature of MOOP, this solution may be optimal with respect to one objective or the total fitness becomes minimum by best satisfying all the objectives but may be a poor candidate for a particular objective. Hence, it is desirable to generate many optimal layouts considering all the three objectives. Therefore, RWGA is designed to obtain non-dominated pareto optimal where one of them are not better than another. Some of the pareto optimal solutions are shown in the following table:

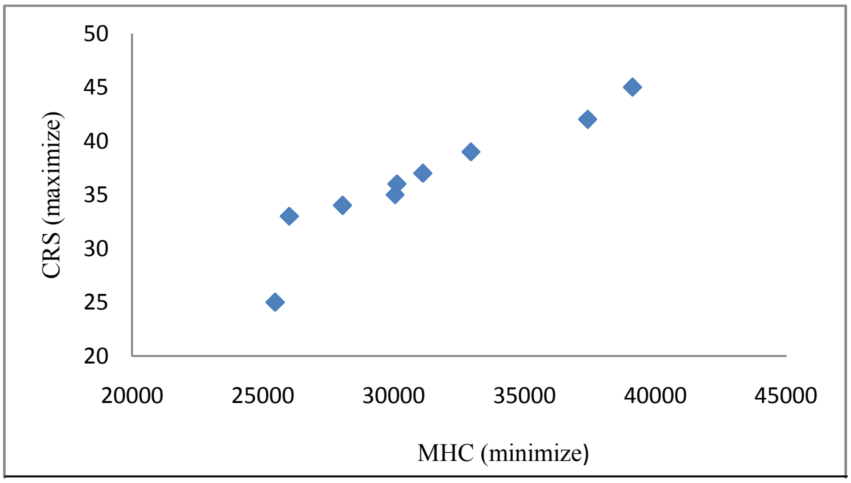
Table 6.16: Pareto optimal solutions obtained from RWGA

	MHC (USD)	**CRS**	**Makespan (Hour)**
Pareto solution 1	26013.54	33	656.36
Pareto solution 2	25473.12	23	710.51
Pareto solution 3	28085.46	34	635.76
Pareto solution 4	30125.68	36	915.39
Pareto solution 5	32946.14	39	790.14
Pareto solution 6	29182.34	26	621.5
Pareto solution 7	39120.32	45	933.51

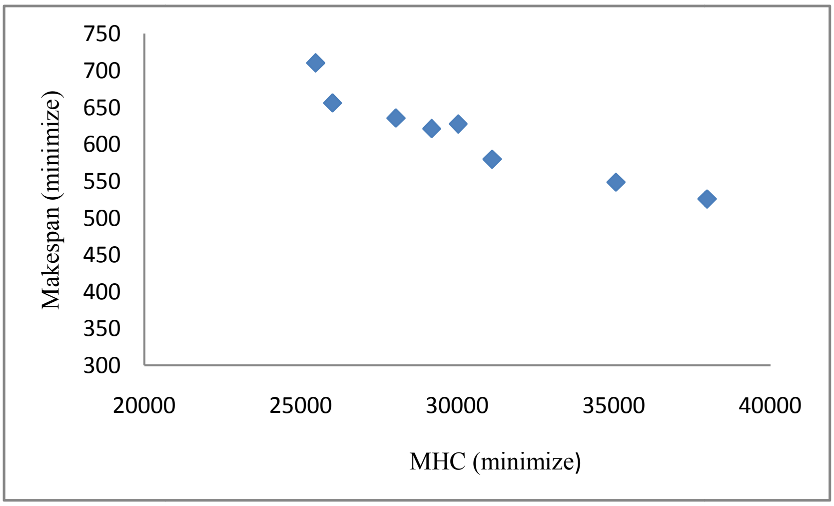
From the above table, it is evident that, none of the solution is better than others with respect to all three objectives. Therefore, these pareto optimal solutions are non-dominated. Thus this solution set will be useful for the decision maker as he can selectively choose the most demanding one among all of the generated layouts for a specific order or customer demands. Thus the developed model is multi-objective in nature and RWGA successfully represented the multi-objective nature of the developed facility layout model.

However, the gaps between the best and average values are a little high as compared to that found from sBB. It may caused due to the randomized parallel search of RWGA as well as the heuristic nature of the algorithm and this type of gap is usual as seen in the literature (e.g., Ripon et al., 2011). Again, it provides a wide range of trade-off solutions and also brings diversity in the pareto optimal solutions and this in turn helps the decision maker in choosing the best layout for a specific criteria as compared to that of sBB. Therefore, the diversity in the pareto front is useful since the goal of multi-objective optimization is to find as many trade-off solutions as possible.

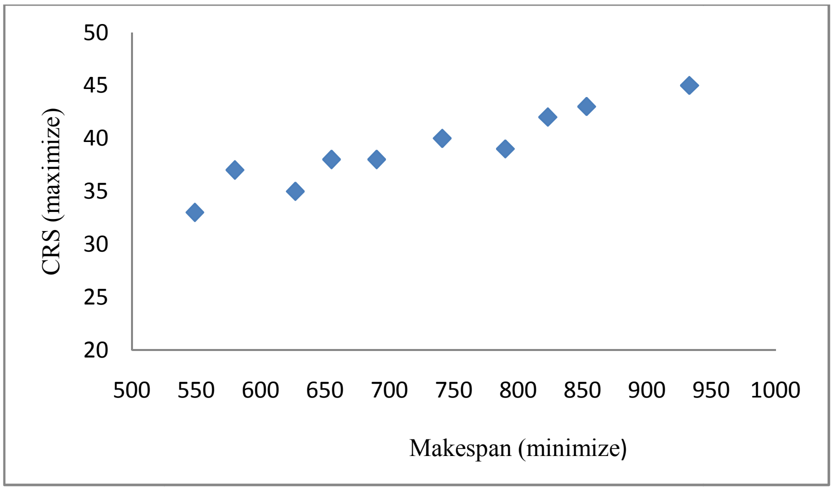
To illustrate the diversity of the solutions, non-dominated (Pareto-optimal) solutions generated using the best parameter combination stated above are presented in Fig.6.2:



(a) Pareto front with MHC and CRS



(b) Pareto front with MHC and Makespan



(c) Pareto front with Makespan and CRS

Figure 6.4: Pareto front obtained from RWGA

It is clearly evident from the above figures that, most of the solutions of the final generation are Pareto optimal. Because without sacrificing one of the objectives others cannot be improved. Again, these figures illustrate the diversity of the solutions and provide clear evident that GA produces a wide range of trade-off solutions than obtained from the sBB.

6.1.2.1 Convergence Analysis

In order to demonstrate the optimization behavior of the proposed RWGA method over generations, a convergence analysis has been performed using the best values of the total fitness function in each generation. In genetic algorithm a population of candidate solutions is generated. The more fit individuals are stochastically selected for crossover and mutation to improve the fitness. In this process, the population of an optimization problem evolves towards better solution. This evolution continues from one generation to the next and eventually converges to an optimal solution. The result of the convergence analysis is presented in the following figure:

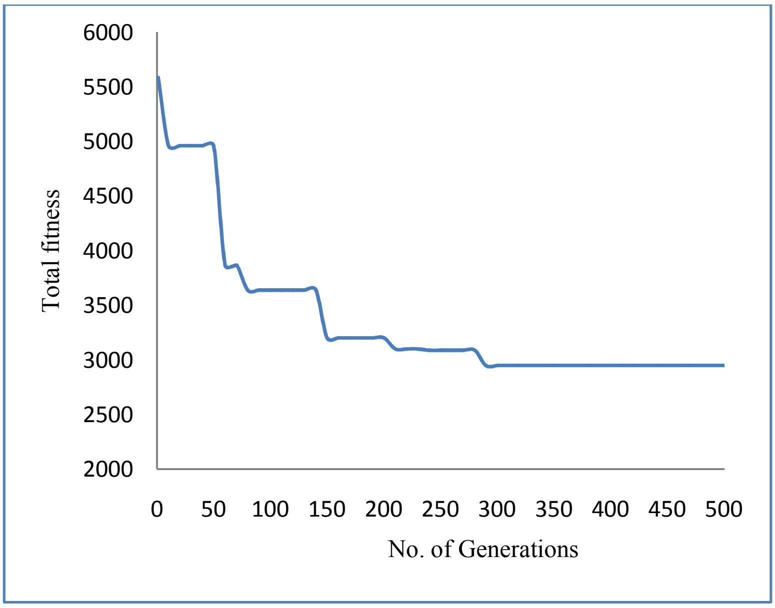


Figure 6.5: Convergence path of the combined objective function by RWGA

The above figure justifies that the proposed RWGA approach clearly optimizes the combined objective function as well as the individual objectives with generations.

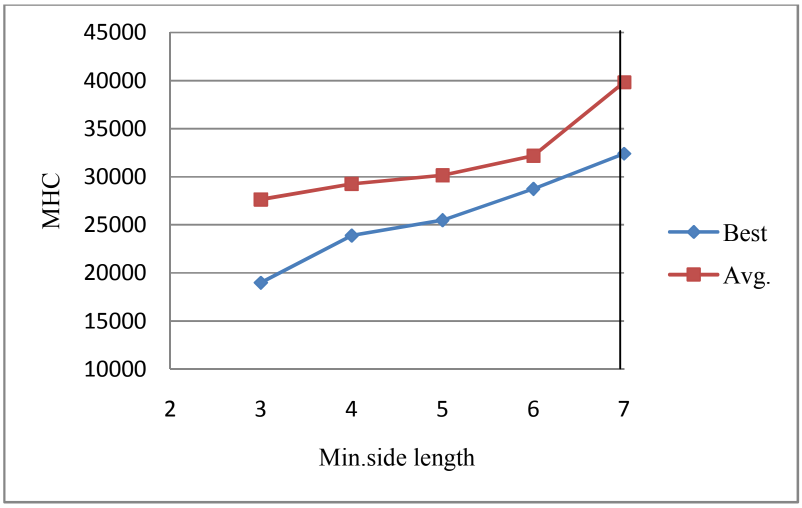
6.1.3 Sensitivity Analysis

A sensitivity analysis has been performed to analyze how the objective function values are affected as the minimum side length of departments varies. The minimum side length of the departments has been varied within a range of 3m to 7m. Since, it has been observed that, the results from RWGA are more responsive to changes in the parameters, therefore the changes has been examined for the results of RWGA. The mean and best values for the three objective functions corresponding to different minimum side length of departments are given in the following table:

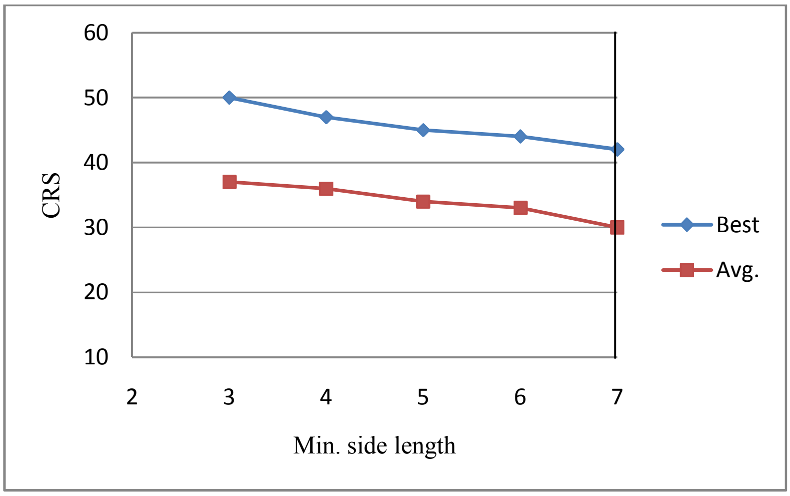
Table 6.17: Results of sensitivity analysis

Min. side length (m)	**MHC (USD)**		**CRS**		**Makespan (Hour)**	
	Best.	**Avg.**	**Best**	**Avg.**	**Best**	**Avg.**
3	18993.46	27636.82	50	37	494.5	624.6
4	23886.81	29254.4	47	36	566.12	669.31
5	25473.12	30134.65	45	34	621.5	751.58
6	28735.36	32161.92	44	33	681.45	812.13
7	31298.93	39853.24	42	30	736.48	896.2

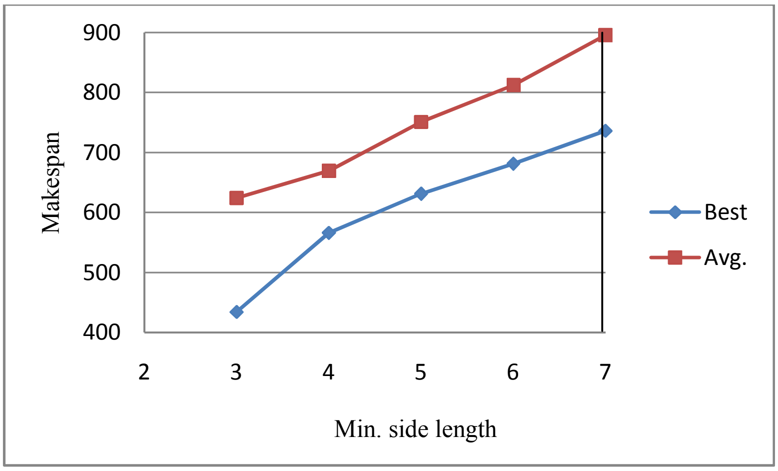
To make the changes in the objective functions in response to changes in the minimum side length of each department more evident, the changes of the best and mean value of the objectives have been presented in the following figures:



(a) Sensitivity of MHC



(b) Sensitivity of CRS



(c) Sensitivity of Makespan

Figure 6.6: Sensitivity analysis

From the above figures, it is evident that, as the minimum side length increases, the solution quality decreases. This occurs because, as the minimum side length increases, the aspect ratio decreases and thus the optimization problem become more constrained and feasible solutions are harder to find. As the minimum side length increases, the departments tend to be square especially the departments having smaller area. It is computationally difficult to arrange the square or nearly square shaped departments within a facility as compared to the long and narrow departments.

It is also evident that, changes in the MHC and Makespan are higher compared to CRS. Since, MHC and Makespan are distance-based function therefore changes in the length or width of the departments have a significant effect on them. Therefore, when narrow departments are place side by side the distance between their centres becomes smaller and thus the distance based functions becomes much smaller. On the other hand, Changes in the CRS are not so noticeable as it is an adjacency based objective.

6.2 Numerical Example 2

In this case, a production plant is considered with six departments of unequal areas. These areas, a_i, in square meters is given in table 6.17. The cost per unit distance of material flow between the departments is given in table 6.18 which depends on mode of transportation. The closeness rating value among the departments required for calculating the closeness rating score is given in table 6.19. Material handling time per unit distance between the departments which depends on the mode of transportation is given in table 6.20 which is required to calculate the makespan.

The dimension of the facility is 20m×18m. Thus, a_T = 360 sq-m. It has been specified that, no department could be narrower than 4m.

Table 6.18: Departmental areas (sq-m) for the 6 departments

Department	**1**	**2**	**3**	**4**	**5**	**6**
Area (a_i)	70	60	90	70	30	40

Table 6.19: Material handling cost (USD) per unit distance

Dept.	**1**	**2**	**3**	**4**	**5**	**6**
1	-	100	0	50	50	0
2		-	0	300	0	200
3			-	50	400	0
4				-	0	250
5					-	350
6						-

Table 6.20: Closeness rating value

Dept.	**1**	**2**	**3**	**4**	**5**	**6**
1	-	0	0	3	2	5
2		-	2	0	3	-1
3			-	5	4	0
4				-	-1	0
5					-	2
6						-

Table 6.21: Material handling time (Hour) per unit distance

Dept.	**1**	**2**	**3**	**4**	**5**	**6**
1	-	5	0	6	4	0
2		-	0	3	0	6
3			-	8	4	0
4				-	0	6
5					-	5
6						-

These data are normalized using Eq. (3.14) given in chapter 3.

6.2.1Optimization using Spatial Branch-and-Bound Algorithm

The developed model is solved using the normalized data and the pareto optimal solutions obtained from sBB is provided in the following table:

Table 6.22: Pareto optimal solutions obtained from sBB

	MHC (USD)	**CRS**	**Makespan (Hour)**
Pareto solution 1	15089.25	17	527.76
Pareto solution 2	16189.13	18	518.24
Pareto solution 3	15121.87	18	567.38
Pareto solution 4	16592.65	19	598.16
Pareto solution 5	17988.43	21	576.41
Pareto solution 6	17013.28	20	609.81
Pareto solution 7	18008.89	22	625.18

It is clear from the above table that, none of the solutions are better than another.

6.2.2 Optimization using Random Weighted Genetic Algorithm

The proposed random weighted genetic algorithm is employed to solve the model using the data and the pareto optimal solutions found is provided in the following table:

Table 6.23: Pareto optimal solutions obtained from RWGA

	MHC (USD)	**CRS**	**Makespan (Hour)**
Pareto solution 1	15971.43	18	456.07
Pareto solution 2	15891.67	23	597.56
Pareto solution 3	14338.09	22	534.69
Pareto solution 4	17138.10	24	474.74
Pareto solution 5	18037.96	25	453.46
Pareto solution 6	13935.86	15	497.06
Pareto solution 7	19313.89	26	448.11

It is seen that all the solutions are non-dominated solutions.

6.3 Comparison of Results between sBB and RWGA

From the above results it is seen that the developed slicing tree structure embedded RWGA can obtain a wide range of trade-off solutions as compared to that of sBB. The RWGA outperforms sBB because the former is specially designed to handle multi-objective optimization problems. Also, the best value of RWGA is slightly better than that of sBB. Again, it is seen that, when the problem size becomes smaller as in numerical example 2 the performance of sBB becomes better than in example 1. This may due to the fact that, though the exact optimization procedures provides a rigorous guarantee of global optimal solutions, their efficiency as well as solution quality decreases as the problem becomes more complex and problem size increases. However, the sBB provides satisfactory results in comparison to RWGA in case of this multi-objective constrained mixed-integer nonlinear programming facility layout model.

CHAPTER VII

CONCLUSIONS AND RECOMMENDATIONS

7.1 Conclusions

Over the years, the facility layout model was developed considering the departments having equal areas. But in reality, the departments of a facility are usually unequal in size. Another assumption in developing the facility layout model was to consider only the material transportation cost in measuring the effectiveness of a facility layout. But in today's competitive world other factors need to be considered in defining the effectiveness of a layout. One important factor is the qualitative factors such as sharing common resources, safety, noise or cleanliness which are not taken in to account by the cost function. Therefore, the qualitative factors aim at maximizing the adjacency preferences which can be quantified by total closeness rating score. Another most important factor in an effective facility layout design is customer satisfaction which is very significant in today's competitive market. Makespan of the overall system significantly affects the customer satisfaction and the goal of an effective layout should be to minimize the makespan. In this thesis work these two important factors have been considered along with material transportation cost with the realistic assumption that the departments are unequal in size. Constraints are also developed for non-overlapping of the departments, bounding the departments within the facility premises, nonlinear departmental area constraints and aspect ratio constraints. Therefore, a multi-objective constrained facility layout model have been developed in order to minimize the material transportation cost and makespan of the overall system as well as to maximize the closeness rating score simultaneously. A reformulation-based Spatial Branch and Bound algorithm approach has been used to optimize the constrained mixed-integer nonlinear multi-objective model. Again, a Random Weighted Genetic algorithm has been combined with Slicing Tree Structure and then this slicing tree based genetic algorithm was employed to optimize the multi-objective model. To validate the effectiveness of the solution process performed for optimization of the developed facility layout model, two approaches are used and approximately similar result is found though the multi-objective genetic algorithm showed better performance in developing diversified pareto optimal solution. Finally a sensitivity analysis is provided to analyze the effects of parameters used in the model and their interactions on the objective functions values.

This research suggests that traditional assumptions that had been customarily been made over the years during the formulation of facility layout model should be relaxed in order to take the model as close as possible to the real scenario. This will help in designing an efficient layout for an organization that will have a better yield in response to real life competitive circumstances.

7.2 Recommendations

There are some possible directions to which this research can be extended. In this thesis, the static facility layout model has been considered where the optimal arrangements of the departments are determined for a single period. This can be relaxed by developing this multi-objective facility layout model considering a dynamic facility layout where, the optimal arrangements of the departments are to be determined for multiple periods. Thus the optimal layout changes from one period to another and thereby requiring the consideration of rearrangement costs. Again, flexibility can be incorporated in the facility layout model by employing a fuzzy rule-based system (FRBS). The flexibility makes a layout robust in cost performance for varying production demands and provides adaptability to new production requirements.

REFERENCES

Adjiman, C. S., and Floudas, C. A., (1996). Rigorous convex underestimators for general twice differentiable problems. *Journal of Global Optimization*, 9:23-40.

Adjiman, C. S., Androulakis, I. P., & Floudas, C. A., (1997). Global optimization of MINLP problems in process synthesis and design. *Computers and Chemical Engineering,* 21:445–450.

Aiello, G., Scalia, G. L. and 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.

Aiello, G., Scalia, G. L. and Enea, M., (2013). A non-dominated ranking multi objective genetic algorithm and electre method for unequal area facility layout problems. *Expert Systems with Applications*, 40(12): 4812-4819.

Anjos, M. and Vanneli, A., (2006). A new mathematical programming framework for facility layout design. *Journal on Computing*, 18 (1):111-118.

Anderson, E. J. and Ferris, M. C., (1994). Genetic algorithm for combinatorial optimization: the assembly line balancing problem. *ORSA Journal of Computing*, 6: 161-173.

Androulakis, I. P., Maranas, C. D. and Floudas, C. A., (1995). αBB: A global optimization method for general constrained nonconvex problems. *Journal of Global Optimization*, 7(4): 337-363.

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

Banerjee, P., Montreuil, B., Moodie, C.L. and Kashyap, R.L., (1992). A modeling of interactive facilities layout designer reasoning using qualitative patterns. *International Journal of Production Research*, 30(3): 433–453.

Bazaraa, M., (1975). Computerized layout design: A branch and bound approach. *AIIE Transactions,* 7(4): 432–438.

Belotti, P., (2009) Couenne: a user's manual. *Tech. Rep., Lehigh University*.

Bozer, Y. A. and Meller, R. D., (1997). A re-examination of the distance based facility layout problem, *IIE Transactions*, 29(7): 549-560.

Bozer, Y. A. and Wang, C. T., (2012). A graph-pair representation and MIP-model-based heuristic for the unequal-area facility layout problem. *European Journal of Operational Research,* 218(2):382–391.

Castillo, I. and Westerlund, T., (2005). An €-accurate model for optimal unequal-area block layout design. *Computers & Operations Research*, 32(3): 429–447.

Chiang, W.C. and Chiang, C., (1998). Intelligent local search strategies for solving facility layout problems with the quadratic assignment problem formulation. *European Journal of Operational Research*, 106(2-3): 457–488.

Chen, Y. K., Lin, S. W. and Chou, S. Y., (2002). An efficient two-staged approach for generating block layouts. *Computers & Operations Research,* 29(5):489:504.

Chang, M. S. and Lin, H. Y., (2012). A flexible bay structure representation and ant colony system for unequal area facility layout problems. *In Proceedings of the World Congress on Engineering,* 3, July 4 - 6, London, U.K.

Davis, L., (1987). Genetic algorithms and simulated annealing (First ed.) *Morgan Kaufmann*, Los Altos, CA, USA.

De Castro, L. N., and Timmis, J., (2002). Artificial immune systems: A new computational intelligence approach. *Springer-Heidelberg.*

Dorigo, M. and Gambardella, L. M., (1997). Ant colony system: a cooperative learning approach to the traveling salesman problem. *IEEE Transactions on Evolutionary Computation*, 1(1): 53-66.

Forrest, S., (1993). *Fifth International Conference on Genetic Algorithms*, University of Illinois at Urbana-Champaign. Morgan Kaufmann, San Mateo, CA, USA, 2-660.

Fourer, R., Gay, D.M., Kernighan, B.W., (2002). AMPL: A modeling language for mathematical programming. *Duxbury Press, Cole Publishing Co.*

Francis, R. L., McGinnis, L. F., and White, J. A., (1992). Facility layout and location: An analytical approach(2nd ed.). *Englewood Cliffs, NJ: Prentice Hall*.

Goetschalckx, M., (1998). Models and algorithms for block layout design with departmental shape constraints. *In Progress in Material Handling Research*.

Goldberg, D.E., (1989). Genetic algorithms in search, optimization, and machine learning. *Addison-Wesley, Reading*, MA.

Grefenstette, J. J., (1987). Incorporating problem specific knowledge into genetic algorithms. *Genetic Algorithms and Simulated Annealing*. Morgan Kaufmann, Los Altos CA: 42-60.

Harmonosky, C. M. and Tothero, G. K., (1992). A multi-factor plant layout methodology. *International Journal of Production Research,* 30(8): 1773-1789.

Holland, J.H., (1975). Adaptation in Natural and Artificial Systems. *University of Michigan Press*, Ann Arbor, MI, USA; reprinted by MIT Press, Cambridge, MA (1992).

Imam, M. H. and Mir, M., (1993). Automated layout of facilities of unequal areas. *Computers in Industrial Engineering*, 24(3): 355-366.

Jankovits, I., Luo, C., Anjos, M. F. and Vannelli, A., (2011). A convex optimization framework for the unequal-areas facility layout problem. *European Journal of Operational Research,* 214(2): 199-215.

Komarudin and Wong, K. Y., (2010). Applying ant system for solving unequal area facility layout problems. *European Journal of Operations Research*, 202(3), 730–746.

Konak, K. S., Norman, B. A., Coit, D. W., and Smith, A. E., (2004). Exploring tabu search memory in constrained problems. *INFORMS Journal on Computing*, 16(3): 241–254.

Kusiak, A. and Heragu, S., (1987). The facility layout problem. *European Journal of Operational Research*, 29(3):229–253.

Lacksonen, T.A., (1994). Static and dynamic layout problems with varying areas. *Journal of the Operations Research Society*, 45 (1): 59–69.

Lacksonen, T.A., (1997). Preprocessing for static and dynamic facility layout problems. *International Journal of Production Research*, 35(4): 1095–1106.

Langevin, A., Montreuil, B. and Riopel, D., (1994). Spine layout design. *International Journal of Production Research*, 32(2): 429–442.

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

Liu, Q. and 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.

McCormick, G. P., (1976). Computability of global solutions to factorable nonconvex programs: Part I —convex underestimating problems. *Mathematical Programming*, 10(1):146 – 175.

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

Meller, R.D., Narayanan, V. and Vance, P.H., (1999). Optimal facility layout design. *Operations Research Letters*, 23(3-5): 117–127.

Meller, R.D. and Gau, K.-Y., (1996). The facility layout problem: recent and emerging trends and perspectives. *Journal of Manufacturing Systems*, 15(5): 351–366.

Meller, R. D., Chen, W. and Sherali, H. D., (2007). Applying the sequence –pair representation to optimal facility layout designs. *Operations Research Letters,* 35(5): 651-659.

Mir, M. and Imam, M. H., (2001). A hybrid optimization approach for layout design of unequal-area facilities. *Computers & Industrial Engineering,* 39(1-2): 49-63.

Montreuil, B., (1990). A modeling framework for integrating layout design and flow network design. Proceedings of *Material Handling Research Colloquium*. Hebron, KY, USA, 43-58.

Montreuil, B., Venkatadri, U. and Ratliff, H.D., (1993). Generating a layout from a design skeleton. *IIE Transactions*, 25(1): 3–15.

Murata, H., Fjuiyoshi, K., Nakatake, S. and Kajitani, Y., (1995). Rectangle-packing-based module placement, in *Proceedings of the IEEE International Conference on Computer Aided Design*, San Jose, CA, USA, 472–479.

Murata T, Ishibuchi H., (1995). MOGA: multi-objective genetic algorithms. *In: Proceedings of the 1995 IEEE international conference on evolutionary computation*, 29 November–1 December, 1995. Perth,WA, Australia: IEEE; 1995.

Murata T, Ishibuchi H, Tanaka H., (1996). Multi-objective genetic algorithm and its applications to flowshop scheduling. *Computers and Industrial Engineering*, 30(4):957–968.

Murata, H. and Kuh, E., (1998a). Sequence-pair based placement method for hard/soft/pre-placed modules, in *Proceedings of the International Symposium on Physical Design*, Monterey, CA,USA, 167–172.

Murata, H. and Kuh, E. (1998b) VLSI/PCB placement with obstacles based on sequence pair. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 17: 60–68.

Osman, I.H., (2006). A tabu search procedure based on a random Roulette diversification for the weighted maximal planar graph problem. *Computers & Operations Research,* 33(9): 2526–2546.

Quesada, I. and Grossmann, I. E., (1995). A global optimization algorithm for linear fractional and bilinear programs. *Journal of Global Optimization*, 6 (1): 39 – 76.

Radcliffe,N.J., (1991). Equivalence class analysis of genetic algorithms. *Complex Systems*, 5(2):183-205.

Rawlins, G.J. E., (1991). Foundations of Genetic Algorithms. *Morgan Kaufmann*, San Mateo, CA, USA.

Ripon, K. S. N., Khan, K. N., Glette, K., Hovin, M. and Torresen, J. (2011). Using pareto-optimality for solving multi-objective unequal area facility layout problem. Proceedings of *13th Annual Conference on Genetic and Evolutionary Computation.* Dublin, Ireland, 681-687.

Ripon, K. S. N., Glette, K., Khan, K. N., Hovin, M. and Torresen, J., (2013). Adaptive variable neighborhood search for solving multi-objective facility layout problems with unequal area facilities. *Swarm and Evolutionary Computation,* 8(1): 1-12.

Safizadeh M.H., McKenna D.R., (1996). Application of multidimensional scaling techniques to facilities layout. *European Journal of Operational Research*, 92(1):54-62.

Sadrzadeh, A., (2012). A genetic algorithm with the heuristic procedure to solve the multi-line layout problem. *Computers & Industrial Engineering*, 62(4): 1055-1064.

Smith, A. E. and Tate, D. M., (1993). Genetic optimization using a penalty function, in *Proceedings of the Fifth International Conference on Genetic Algorithms and Their Applications*, 499-505.

Smith, E.M.B., (1996). On the optimal design of continuous processes. Ph.D. thesis, *University of London, UK*.

Smith, E. M. B. and Pantelides, C. C., (1996). Global optimisation of general process models. In I. E. Grossmann (ed.), *Global Optimization for Engineering Design,* 356 – 386.

Smith, E. M. B. and Pantelides, C. C., (1999). A symbolic reformulation: spatial branch-and-bound algorithm for the global optimisation of nonconvex MINLPs. *Computers and Chemical Engineering,* 23:457 – 478.

Schockaert, S., Smart, P. D., and Twaroch, F. A., (2011). Generating approximate region boundaries from heterogeneous spatial information: An evolutionary approach. *Information Sciences*, 181(2): 257–283.

Scholz, D., Petrick, A. and Domschke, W., (2009). A Slicing Tree and Tabu Search based heuristic for the unequal area facility layout problem. *European Journal of Operational Research,* 197(1):166–178.

Scholz, D., Jaehn, F. and Junker, A., (2010). Extensions to STaTS for practical applications of the facility layout problem. *European Journal of Operational Research,* 204(3): 463-472.

Solimanpur, M. and 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.

Sha, L. and Dutton, R.W., (1985). An analytical algorithm for placement of arbitrarily sized rectangular blocks, in *Proceedings of the 22nd ACM/IEEE Design Automation Conference*, San Diego, CA, USA, 602–608.

Sherali, H.D., Fraticelli, B.M.P. and Meller, R.D., (2003). Enhanced model formulations for optimal facility layout. *Operations Research*, 51(4): 629–644.

Sule, D.R., (1994). Manufacturing Facilities: Location, Planning and Design, 2nd Ed.(PWS Publishing: Boston).

Taghavi, A. and Murat, A., (2011). A heuristic procedure for the integrated facility layout design and flow assignment problem. *Computers & Industrial Engineering*, 61(1):55-63.

Takouda, P. L., Anjos, M. F. and Vannelli, A., (2005). Global lower bounds for the VLSI macrocell floorplanning problem using semidefinite optimization. *In: Proceedings of IWSOC*, 275–280.

Tate, D. M. and Smith, A. E., (1995). Unequal-area facility layout by genetic search, *IIE Transactions,* 27 (4): 465-472.

Tompkins J.A., White, J. A., Bozer, Y. A. and Tanchoco, J. M. A., (2003). *Facilities planning (3rd ed.)*. New Jersey: Wiley.

Tong, X., (1991). SECOT: a sequential construction technique for facility design, *Doctoral Dissertation,* University of Pittsburgh.

Ulutas, B. H. and Konak, S. K., (2012). An artificial immune system based algorithm to solve unequal area facility layout problem. *Expert Systems with Applications*, 39(5):5384–5395.

Van Camp, D. J., Carter, M. W. and Vannelli, A., (1991). A nonlinear optimization approach for solving facility layout problems. *European Journal of Operational Research,* 57(2): 174-189.

Winston, W.L., Venkataramanan, M., (2003). Introduction to Mathematical Programming, *Fourth ed. Thomson* (Chapter 9).

Wong, D.F. and Liu, C.L., (1986). A new algorithm for floorplan design. *Proceedings of the 23rd ACM/IEEE Conference on Design Automation*. *IEEE Press,* 101–107.

Wong, D.F. and Liu, C.L., (1989). Floor plan design of VLSI circuits. *Algorithmica*, 4(1-4): 263–291.

Wong, K. Y. and Komarudin, (2010). Solving facility layout problems using flexible bay structure representation and ant system algorithm. *Expert Systems with Applications,* 37(7): 5523–5527.

Zamora, J. M., and Grossmann, I. E., (1997). A comprehensive global optimization approach for the synthesis of heat exchanger networks with no stream splits. *Computers and Chemical Engineering*, 21: 65 – 70.