An Intelligent Expert System

for

Decision Analysis and Support

in

Multi-Attribute Layout Optimization

by

Abdul-Rahim Ahmad

A thesis

presented to the

University of Waterloo

in fulfillment of the

thesis requirement for the degree of

Doctor of Philosophy

in

Systems Design Engineering

Waterloo, Ontario, Canada, 2005

AUTHOR'S DECLARATION

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to public.

Abdul-Rahim Ahmad

ABSTRACT

Layout Decision Analysis and Design is a ubiquitous problem in a variety of work domains that is important from both strategic and operational perspectives. It is largely a complex, vague, difficult, and ill-structured problem that requires intelligent and sophisticated decision analysis and design support.

Inadequate information availability, combinatorial complexity, subjective and uncertain preferences, and cognitive biases of decision makers often hamper the procurement of a superior layout configuration. Consequently, it is desirable to develop an intelligent decision support system for layout design that could deal with such challenging issues by providing efficient and effective means of generating, analyzing, enumerating, ranking, and manipulating superior alternative layouts.

We present a research framework and a functional prototype for an interactive Intelligent System for Decision Support and Expert Analysis in Multi-Attribute Layout Optimization (IDEAL) based on soft computing tools. A fundamental issue in layout design is efficient production of superior alternatives through the incorporation of subjective and uncertain design preferences. Consequently, we have developed an efficient and Intelligent Layout Design Generator (ILG) using a generic two-dimensional bin-packing formulation that utilizes multiple preference weights furnished by a fuzzy Preference Inferencing Agent (PIA). The sub-cognitive, intuitive, multi-facet, and dynamic nature of design preferences indicates that an automated Preference Discovery Agent (PDA) could be an important component of such a system. A user-friendly, interactive, and effective User Interface is deemed critical for the success of the system. The effectiveness of the proposed solution paradigm and the implemented prototype is demonstrated through examples and cases.

This research framework and prototype contribute to the field of layout decision analysis and design by enabling explicit representation of experts' knowledge, formal modeling of fuzzy user preferences, and swift generation and manipulation of superior layout alternatives. Such efforts are expected to afford efficient procurement of superior outcomes and to facilitate cognitive, ergonomic, and economic efficiency of layout designers as well as future research in related areas.

Applications of this research are broad ranging including facilities layout design, VLSI circuit layout design, newspaper layout design, cutting and packing, adaptive user interfaces, dynamic memory allocation, multi-processor scheduling, metacomputing, etc.

ACKNOWLEDGEMENTS

Although I am indeed the sole author of this thesis, as customarily declared in the beginning, I am surely not the sole contributor! So many people have contributed to my thesis, to my education, and to my life, and it is now my great pleasure to take this opportunity to thank them. I would like to gratefully acknowledge the enthusiastic supervision of Professor Otman Basir and Professor Khaled Hassanein. I express my appreciation and deep gratitude for their unprecedented support, guidance, great companionship, and belief in me. Their incessant guidance helped me in making valuable contributions. They taught me how to adopt an interdisciplinary and systems approach in scientific inquiry to important research problems. Despite their hectic schedules, they always had time for me.

I am grateful to Dr. Basir for so many other things including enlightening discussions, moral support, and advices in my career development/search. Dr. Basir's extensive and award-winning contributions to leading edge industrial research have been great assets for me. His tenacity, wisdom, and perseverance have been sources of inspiration and guidance for me. Above all, he understands the contrast between creativity and productivity and he is appreciative of both in scientific inquiries. His supreme level of enthusiasm fascinates me and often inspires and challenges me to make my best efforts. His award winning teaching skills, passion for teaching, and sensitivity to students' needs influenced me very much. His infectious optimism and dynamic personality transformed the manner in which I approach challenges in my career. He is an ideal role model for aspiring scientists and academicians.

Dr. Hassanein's sound judgment and project management skills were instrumental in the successful completion of this thesis. I would like to thank him for his mentoring, support, and encouragement. I greatly admire his meticulous and patient approach to research. He has guided and helped me with his extensive knowledge and experience during my research. He has an open mind and a sharp eye for the merit of ideas; this was particularly valued when I was faced with the challenge of choosing among many possible research tracks. I thank him for his efforts towards the development of my career and this work. His contributions to every stage of this research made this dissertation possible. His continued support never let me feel that he was actually stationed in a different city.

Special thanks to Dr. C. Burns, Dr. F. Karray, Dr. F. Safayeni, and Dr. M. Wang, members of the thesis examining committee, for devoting their time out of their busy schedule. Dr. Burns, Dr. Karray, and Dr. Safayeni served on the committee for my comprehensive examination and contributed valuable insights and ideas from their own areas of specialization. Dr. Karray introduced me to the potential of soft computing and machine intelligence tools in solving complex decision problems. Dr. Burns and Dr. Safayeni are among those few people who taught me to think differently and never leave behavioral and ecological perspectives out of the frame in solving a problem. Despite being on sabbatical leave, Dr. Burns and Dr. Wang agreed to serve in the thesis examining committee.

Several people have been an endless source of support and inspiration including my father Dr. Nasir Ahmad, my father-in-law Dr. Muhammad Hasan Imam, Dr. Imran Tasadduq, Umar Rizwan, Dr. Azhar Quddus, Dr. Mushtaq Luqmani, Rizwan Qureshi, and many more.

Dr. Imam was instrumental in providing technical support. It is difficult to overstate my gratitude to him. Throughout my research, he provided inspiration, sound advice, and many good ideas. Dr. Imam and his colleagues employed preliminary versions of IDEAL as additional teaching aids in undergraduate courses related to layout design, beside other commercially available automated layout design software, contributing valuable insights to my thesis work.

I also want to express my gratitude to an anonymous technical support specialist at VIP-PlanOpt™ (www.planopt.com) for assisting in comprehending the market demands and assessing my work against what may be termed as commercial layout design software. Furthermore, many thanks to scores of those undergraduate students who used preliminary versions of our software and provided us valuable suggestions for improvement.

Above all, thanks to my better half for her encouragement, understanding, and patience and to my wonderful kids for their affection that had been among the primary sources of inspiration for me.

Dedicated

to

My Much-loved Mother

whose tacit desires lead me to this daunting enterprise

to

My Wonderful Wife

whose support and encouragement were instrumental in my success

to

My Charming Children

who missed me when I was needed most

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

Short Form	Long Form
AER	Area of Enclosing Rectangle.
ANN	Artificial Neural Network(s).
AQ	Area of a Quadrant of Enclosing Rectangle.
AR	Aspect Ratio.
AUI	Adaptive User Interface.
ALD	Automated Layout Design System(s).
BAM	Bi-directional Associative Memory.
BF	"Best-Fit" Algorithm(s).
BL	"Bottom-Left" Algorithm(s).
BLF	"Bottom-Left Fill" Algorithm(s).
BN	Bayesian Networks.
BP	Back-Propagation.
BPH	A Hypothetical Benchmark Best Packing Height.
BPN	Back-Propagation Network.
BPP	Bin-Packing Problem.
CAD	Computer-Aided/Assisted Design.
CF	Certainty Factors approach to quantitative modeling under uncertain environments.
CLD	Automated Circuit Layout Design System(s).
CR	Contiguous Remainder.
$C\hat{R}$	Normalized Contiguous Remainder.
DA	Decreasing Area Sequence (of modules)
DL	Decreasing Length Sequence (of modules)
DM	Decision Maker(s).
DS	Dempster-Shafer (DS) theory of evidence.
DSS	Decision Support System(s).
e-Commerce	Electronic Commerce.
ER	Enclosing Rectangle.

Short Form	**Long Form**
ES	Expert System(s).
FDM	Fuzzy Decision-Making.
FDMS	Fuzzy Decision-Making System.
FL	Fuzzy Logic or Fuzzy Logic Theory.
FLD	Automated Facility Layout Design System(s)
FMS	Flexible Manufacturing System.
GA	Genetic Algorithm(s).
GUI	Graphical User Interface(s).
HT	Height of the Packing Pattern or the Packing Space.
$H\hat{T}$	Normalized Height of the Packing Pattern.
HCI	Human-Computer Interaction.
IBL	Improved BL-Algorithm
IDEAL	Intelligent System for Decision Support & Expert Analysis in the Layout Design.
ILD	Automated Interface Layout Design System(s)
ILG	Intelligent Layout Generator.
IMD	Sum of Inter-Module Distances.
IMI	Inter-Module Interaction.
KBLD	Knowledge-Based Layout Design.
LD	Layout Design.
LP	Linear Programming.
LLF	“Least Flexibility First” Algorithm.
MERAM	“Minimization of Enclosure under Magnetic Attraction” Algorithm.
MCDM	Multi-Criteria Decision Making.
MERAG	“Minimization of Enclosure under Gravitational Attraction” Algorithm.
MERA	“Minimization of Enclosing Rectangle Area” Algorithm.
MF	Membership Function.
MLP	Multi-Layer Perceptron.
MSE	Mean Square Error.

Short Form	Long Form
MT	Module Tightness.
NBES	Neuro-Based Expert System.
NE	Naive Evolution.
NP-Hard	Non-Polynomial Hard.
PDA	Preference Discovery Agent.
PIA	Preference Inferencing Agent.
PP	Preference Parameter.
QAP	Quadratic Assignment Problem.
QR	Quality Rating.
QSC	Quadratic Set-Covering Problem.
SP	Significance Parameter.
RBES	Rule-Based Expert System.
REL	REL Chart(s).
RL	Reinforcement Learning.
RLA	Reinforcement Learning Agent.
RS	Random Search.
TS	Tabu Search.
UI	User Interface(s).
VLSI	Very Large Scale Integration.
Web / WWW	World Wide Web.

LIST OF NOTATIONS

General Notations	
n	Number of given modules
m	Number of given identical packing-spaces
i	The index representing a module; i.e., $i = 1, 2, 3, \ldots n$.
k	The index representing the candidate location of a module
t	The index representing the module in the packing space
M_i	The i^{th} module represented by: $M_i = (w_i, h_i, \alpha_i)$.
W	Width of the bin or packing space.
H	Height of the bin or packing space.
w_i	The width of the module M_i ($w_i \leq W$).
h_i	The height of the module M_i ($h_i \leq H$).
x_i	X-coordinate of the position of the bottom-left corner of the module M_i.
y_i	Y-coordinate of the position of the bottom-left corner of the module M_i.
α_i	Intrinsic Utility (or fitness weight) per unit area of the module M_i.
X_c	X-coordinate of the point of focus/interest on the packing space.
Y_c	Y-coordinate of the point of focus/interest on the packing space.
b_i	A binary decision variable set to 1 when the i^{th} module is included in the solution, otherwise b_i is set to zero.
Additional Notations for the QAP Formulation	
$C(i,j)$	The cost due to inter-module interaction between the module M_i and the module M_j.
$\delta(i,j)$.	The cost due to spatial separation between the module M_i and the module M_j.
$F(i,k)$	The fixed cost involved with the placement of the module M_i at location k.
$S(i)$	Denotes the location to which the module M_i is allocated in a mapping modules to locations.
Additional Notations for the Set Covering Formulation	

b_{ik}	A binary decision variable set to 1 when k^{th} candidate location of the module M_i is included in the solution.
$I(i)$	Total number of possible locations for the module M_i.
$J_i(k)$	Set of basic unit blocks occupied by the module M_i when assigned to its candidate location j.
Additional Notations for the 2D-BPP Formulation	
β_i	The utility by virtue of positioning the module M_i in the quadrant β.
u_i	A function of the intrinsic utility (α_i) and the spatial location of the module M_i in the bin (x_i,y_i), i.e. $u_i = f(\alpha_i, x_i, y_i)$.
ζ	Penalty factor per unit area for wastage/unused area.
θ_i	Angular separation between the x-axis the line joining the geometric center of the module M_i .
δ_i	The Euclidean distance between the geometric center of the module M_i from the point of focus/interest on the packing space.
δ_{max}	The maximum Euclidean distance between the geometric center of the module M_i from the point of focus/interest on the packing space.
$f_{i,j}$	The inter-module interaction as measured from the module M_i to the module M_j.
Additional Notations for Genetic Algorithms based Optimization	
HT	Height of a Layout Configuration.
$\hat{H}T$	The Normalized Height of a Layout Configuration.
BPH	A Hypothetical Benchmark Packing Height. (Calculated as: $\sum_{i=1}^{n} w_i h_i / W$)
Additional Notations for Fuzzy Preference Modeling	
μ	Fuzzy Membership Function
κ	An index representing a Fuzzy Attribute
f_κ	Fitness value of a Fuzzy Attribute κ
S_κ	Significance assigned to the fitness value of Attribute κ
P_κ	Preference assigned to the fitness value of Attribute κ

Chapter 1

INTRODUCTION

1.1 Motivation

The continuous development of such sophisticated and pervasive applications as Facilities buildings, VLSI Circuits, Human-Machine Interfaces, and the Web Page layout design has engendered a strong appeal in formulating and automating layout design algorithms and guidelines (Dowsland *et al.*, 2002; Tompkins *et al.*, 2002; Youssef *et al.*, 2003). These applications have motivated many areas of Operations Research and Decision Sciences and culminated in significant research in formalizing layout design algorithms, preferences, and fitness measures. Nevertheless, despite being an active research area, layout design is still a vaguely defined field (Tommelein, 1997; Youssef *et al.*, 2003). The existing research largely provides design algorithms and guidelines in a very rigid and overly simplistic framework, largely without an elaborate methodology for utilizing those (Tompkins *et al.*, 2002). The usefulness of such overwhelmingly scattered knowledge is further limited by cognitive limitations of users. In order to address some shortcomings of the existing research, this thesis presents a new research paradigm and solution methodology for undertaking the Layout Design problem. It tackles some important issues encountered in layout design by providing means for incorporating sub-cognitive, subjective, and uncertain preferences into the design process and fast generation and manipulation of superior layout alternatives.

1.2 Layout Design

The Layout Design (henceforth, LD) process is geared towards seeking superior outcomes in assigning space to various activities and components. More specifically, it involves spatial configuration of modules in a specified space, satisfying given preferences and constraints, and optimizing some fitness evaluation objectives. These preferences involve a high degree of subjectivity, uncertainty, dynamism, and complexity (Karray *et al.*, 2001b; Singh & Wang, 1994; Youssef *et al.*, 2003a). The knowledge-intensive nature, absence of accurate models that capture complex decision-making dynamics, and non-availability of expert advice in a timely or economical fashion highlight the need for resorting to some knowledge-based decision support and expert system methodologies. Indeed, decision support and expert system technologies have been successfully developed and deployed in such diverse disciplines as Engineering, Business, Mining, Medicine, etc. (Marakas, 2002; Negnevitsky, 2002; Turban and Aronson, 2001). However, little such efforts have been expended in the important area of LD (Ahmad *et* al., 2004b; Tommelein, 1997; Eom and Lee, 1990).

Evidently, the creativity of designers as well as synergy and trade-offs of various competing and complementing faculties lie at the core of the LD process (Berkun, 2001; Tompkins *et al.*, 2002). Incidentally, it has long been noted that " … most [computerized] layout design techniques have limited or ignored the creativity and the natural ability of [the layout planner] to understand complex flow and spatial relationships" (Blair, 1985, pp. 92). Paradoxically, such statements are as true today as were two decades ago (Ligget, 2000; Tompkins *et al.*, 2002). We believe that this state of affairs is primarily an outcome of adopting an Optimization paradigm instead of a more relevant Decision-Making paradigm.

In this regard, we recommend a paradigm shift towards seeking a synergistic bliss of the cognitive and sub-cognitive expertise of decision-makers as well as technologies that are efficacious in modeling such subjective problems. It should be noted that the Expert System paradigm is known to be very effective in such uncertain, subjective, and knowledge-intensive application domains as LD. On the same note, Fuzzy Logic has been an effective and increasingly popular technology for incorporating subjective and uncertain preferences in the knowledge-base of expert systems (Negnevitsky, 2002; Turban & Aronson, 2001). Similarly, such machine learning technologies as Artificial Neural Networks and Reinforcement Learning hold the promise of automatically discovering, validating, and updating some of the implicit and explicit preferences (Lok & Feiner, 2001). In short, we believe that automated LD systems based on an intelligent expert system paradigm, utilizing synergic strengths of various complementary soft computing technologies, provide a promising research direction. Such integrated approaches are largely missing from the LD literature. Nevertheless, persistent and concerted efforts in this direction are expected to enhance the cognitive, economic, and ergonomic efficiency and efficacy of layout designers and produce outcomes sensitive to financial, social, political, or environmental merits.

1.3 Proliferation of the Layout Design Problem

This thesis is motivated by the widespread operational and strategic applications of the LD problem and the inadequacy of existing automated LD systems. The significance and prevalence of this problem is also evident from hundreds of research papers and scores of books written in this area. Consequently, it is not possible to outline the application areas and proliferation of the LD problem in few lines. As such, here we summarize some prominent LD applications to underscore its importance.

1.3.1 Facilities Planning

Facility Layout Design problem has been addressed analytically for the last several decades and received considerable interest from the research community. Indeed, the facility location and layout has profound effects on organizational productivity and profitability. For instance, about 20-50% of operating costs in manufacturing relates to materials handling, a factor highly correlated to the quality of the facility layout. Furthermore, US businesses spend about a trillion dollars on new facilities annually (Tompkins *et al.*, 2002). Consequently, Facilities LD remains an active research area with significant potential for automation.

1.3.2 VLSI Circuit Design

Efforts in automating the Circuit Layout Design gained prominence with the onset of such challenging applications as VLSI. Conceivably, designing the layout configuration of a circuit containing hundreds of millions of components is a very hard problem. As such, the VLSI circuit design process is broken down into such steps as defining macrocells, their connectivity, and placement, etc. Macrocells are a collection of circuit elements that are grouped together based on some connectivity or functionality criteria (Mazumdar & Pianki, 1999). Once macrocells are defined, a block layout is developed where the physical location of each macrocell is specified. This block LD problem is very similar to the bin-packing problem discussed in Chapter 2. An automatically optimized VLSI layout would result not only in a shorter development cycle time but also bring about improvements in various critical circuit performance parameters (Youssef *et al.*, 2003).

1.3.3 Cutting and Packing

In packing problems, the objective is to pack the most objects in the least number of fixed space bins, while maximizing the space utilization. This problem is also relevant to industries where cutting of rectangular patterns from a larger rectangular piece is involved such as sheet metal, paper, plastic, and textile industries (Al-Sultan *et al.*, 1996). Interestingly, applications of packing techniques span apparently disparage domains from scheduling of jobs in assembly lines to dynamic memory allocation and multi-processor scheduling in computing and metacomputing services (Burke *et al.*, 2004). In addition, this problem is also pertinent to transportation industries (Islier, 1998). The enormity of the transportation market is indicated by over $1.5 trillion spent every year in this sector in USA (DOT, 2004). Conceivably, even a slender improvement in the packing efficiency could result in savings of billions of dollars.

1.3.4 Adaptive User Interfaces

Research in automating the visual User Interface design acquired eminence through such pervasive applications as adaptive Web and Mobile services (Akouminiakis, 2000). The process of mapping the domain objects and their properties into corresponding visual properties in the LD is critical to the success of such services. However, an extremely diverse target population makes it an extremely difficult problem and a popular strategy is to use stereotypical categories and deliver visual layouts based on behavioral and cognitive traits of users. In this regard, the need of a knowledge-based decision aid system cannot be overemphasized. Indeed, the return on investment in designing superior interface layouts can be dramatically large. For instance, studies in Sun Microsystems™ have shown how spending about $20,000 in improving the layout and other usability determinants could yield a savings of more than $150 million dollars (Rhodes, 2000).

1.4 Problem Statement

Our intention is to build a generic research framework for the layout design. However, here we employ an oriented and orthogonal two-dimensional rectangular packing problem (2D-BPP) for investigation and illustration purposes, which is considered to be among the most general and difficult formulations for the two-dimensional LD (Burke *et al.*, 2004; Dyckhoff, 1990; Garey & Johnson, 1981; Ligget, 2000). Importantly, this formulation may easily be adapted to other LD problems by largely changing the knowledge-base that contains rules and preferences, thus providing a basis for developing a generic approach to the LD problem. In an orthogonal 2D-BPP, the objective is to pack rectangular modules in a rectangular packing space in an orthogonal manner. This problem has been formulated variously by different researchers (Bazaraa, 1974; Dowsland *et al.*, 2002; Garey & Johnson, 1979). One representative and generic formulation for 2D-BPP is provided in Section 4.2.

1.5 Thesis Objectives

The foremost task in automating the layout design process is the generation of superior Layout Decision Alternatives (Francis & White, 1992; Gomez *et al.*, 2003; Muther, 1961; Tompkins *et al.*, 2002). Such decision alternatives characterize various choices available to layout planners for further consideration and manipulation. In this regard, our approach involves elaboration, articulation, enumeration, categorization, and manipulation of competing design alternatives through propagation of design knowledge into the development cycle (Akoumianakis 2000; Triantaphyllou *et al.*, 1998). An interactive decision support system directed at mitigating the cognitive and information overload of the layout planners by providing fast and easy approaches to generating, analyzing, and revising superior alternatives seems to be a rational choice. Such a system would afford various salient domain specific aspects that go beyond the expertise or the cognitive capabilities of users.

Following are the thesis objectives in specific terms:

1. Develop a conceptual framework for intelligent and knowledge-based decision support in the layout design and analysis.
2. Develop fast layout design heuristics that provide superior layout alternatives with higher aesthetic contents.
3. Develop an Intelligent Layout Generator that could efficiently generate superior decision alternatives in an automated manner.
4. Develop a fuzzy Preference Inferencing Agent for employing subjective and uncertain preferences in guiding the generation of superior decision alternatives.
5. Design and implement computer interfaces for systems developers, domain experts, and end-users that provide visible, effective, and efficient means for modifying various system/design parameters as well as manipulating and refining layout alternatives.
6. Test the viability of the notion of automated preference discovery in the automated layout design context.

The main objective of this dissertation is to develop, implement, and validate an architectural research framework and a research prototype of an **I**ntelligent System for **D**ecision Support and **E**xpert **A**nalysis in **L**ayout Design (henceforth, IDEAL) for facilitating effective and efficient layout planning process. Notably, the emphasis of this research framework is not on the pursuit of some perfect methods but rather on the development of a generic research paradigm and a tool that could be used in furthering the research in layout planning by supplementing the knowledge, experience, and design intuition of layout planners. In achieving these objectives, our approach involves tackling various important aspects of the problem through a synergistic utilization of some promising soft computing techniques, advanced heuristics, and metaheuristics.

Indeed, the demonstrated success of the expert system paradigm in a variety of complex and subjective task domains makes it our rational choice (Ayyub, 2001; Marakas 2002; Negnevitsky 2000; Turban & Aronson, 2001). An efficient algorithm for generating superior layout decision alternatives is an important step in this regard. Consequently, we have developed various efficient algorithms for the use in a hybrid fuzzy-genetic Intelligent Layout Generator (ILG). The intelligence aspect emanates from the use of fuzzy rules and preferences for obtaining penalties and rewards in the evaluation of a genetic fitness function. Accordingly, a fuzzy logic based Preference Inferencing Agent (PIA) seems to be a logical component for such a tool. However, the evolutionary and implicit nature of knowledge suggests that an online Artificial Neural Network based Pattern Discovery and Validation Agent (PDA) could add value by providing some cognitive and sub-cognitive preferences in an automated and self-updated manner. Consequently, we have built a small-scale PDA for testing the concept. A prototype for an interactive end-user interface has also been developed and tested. Details of our approach to the LD problem are provided in Chapter 3.

1.6 Challenges Involved

The LD problem is so vast in scope and engages such a myriad of tangible and intangible factors as well as dynamism that legitimacy of any decreed optimal outcome could easily be contested (Berkun, 2001; Irani & Huang, 2000). Furthermore, the clientele of certain LD applications, such as e-Store or e-Learning services, have diverse backgrounds, cultures, and mental/social metaphors. Conceivably, they have conflicting needs and preferences.

Moreover, the efficacy and applicability of existing LD procedures are severely constrained by the cognitive abilities of users in comprehending, appreciating, and quantifying system related characteristics (Akoumianakis, 2000). In addition, various mathematical formulations for the LD problem are known to be NP-Hard in a strong sense. Consequently, even for modestly large problems, reasonably fast procurement of an optimal solution becomes elusive even with powerful computing resources (Mak *et al.,* 1998; Martello & Vigo, 1998).

In short, layout design is a tedious process and requires sophisticated decision modeling and tools (Tompkins *et al.*, 2002; Epstein *et al.*, 2001). It is a complex and ill-structured problem where evolving tasks and inadequate information availability and processing capability hamper the realization of a superior outcome (Abdinnour-Helm & Hadley, 2000).

Such barriers imply that LD is not readily amenable to automation and have a role in the reluctance to the study of the problem in an encompassing manner (Whyte & Wilhelm, 1999a). Nevertheless, computerized tools and heuristics may be advantageous in modeling some salient features and enumerating some superior LD alternatives (Tompkins *et al.*, 2002). Accordingly, it is possible to support layout designers through various prudently selected means. The ubiquity of layout design applications further highlights the significance of automating the design and analysis process (Burke *et al.*, 2004). Consequently, it is important to alleviate the cognitive and information overload encountered by layout designers in acquiring, remembering, understanding, analyzing, and employing the vast body of the existing domain-specific LD knowledge.

1.7 Organization of the Thesis

This thesis is organized as follows. Chapter 1 provides motivation and rationale for this thesis. Chapter 2 presents a brief literature review of some of the relevant faculties and tools as well as their significance in this research. Chapter 3 delineates the research framework and its various major constituents. Chapter 4 describes the proposed layout optimization algorithms and approaches as well as the implementation and working of the Intelligent Layout Generator. Chapter 5 contains details about modeling of, and inferencing from, subjective and uncertain preferences as well as the design, implementation, and working of the Preference Inferencing Agent. It also describes a small-scale Preference Discovery Agent for testing the idea of automated preference discovery and revision. Chapter 6 concludes the thesis with a summary of a number of valuable interpretations of results and insights gained through this research. In addition, some promising future research directions are summarized.

1.8 Summary

In this chapter, we have described the layout design problem, its significance and relevance, as well as the solution methodology adopted by us. We presented the idea of designing and developing an intelligent layout decision analysis and design support system based on the expert system paradigm and other soft computing methodologies. We indicated that the synergy of three intelligent complementary components namely an Intelligent Layout Generator, a Preference Inferencing Agent, and a Preference Discovery Agent offers the promise of tackling the layout deign problem more effectively. This approach is expected to provide good reference to experts and novice layout designers and facilitate their cognitive advancement. In the subsequent chapter, we provide an overview of the related literature.

Chapter 2

LITERATURE REVIEW

2.1 Introduction

This thesis draws from broad ranging disciplines including layout design, human factors, human cognition, decision support systems, expert systems, soft computing, intelligent systems, etc. Consequently, a comprehensive review of all such concepts, literature, and ensuing efforts is beyond the scope of this thesis. Nevertheless, we provide an overview of relevant concepts and literature from an automated LD perspective. We delineate some of the major limitations of the existing automated LD approaches. Furthermore, we propose some promising research methodologies to alleviate some of these limitations.

2.2 Layout Design

The Layout Design process is geared towards seeking some superior outcome in the spatial arrangement of modules in a given space, satisfying given preferences and constraints, and optimizing some fitness metrics. It is a tedious process necessitating sophisticated modeling techniques and decision aids (Epstein *et al.*, 2001; Tompkins *et al.*, 2002; Zhang *et al.*, 2002). More specifically, it is a complex, subjective, and ill-structured problem where the evolving task dynamics, inadequate information availability/processing capabilities, cognitive biases, as well as uncertain and conflicting preferences often hamper the achievement of a superior outcome (Abdinnour-Helm & Hadley, 2000). Indeed, there is overwhelming evidence that due to complexities of the task and cognitive limitations decision-makers often resort to simplified cognitive strategies or heuristics (Stam & Silva, 2003; Yang & Kuo, 2003).

Indeed, the diverse scope of the LD problem means that a substantial literature is available in a variety of work domains (Abdinnour-Helm & Hadley, 2000; Ahmad *et al.*, 2004e; Burke *et al.*, 2004; Akoumianakis 2000; Karray *et al.*, 2000a; Tompkins *et al.* 2002; Youssef *et al.*, 2003a; Zhang *et al.*, 2002). It has been variously referred to as *topology optimization* (Mir & Imam, 1992), *block placement* (Siarry *et al.,* 1987)*, macro cell placement* or *VLSI layout design* (Esbensen & Mazumder, 1994; Schnecke & Vornberger, 1997), *layout optimization* (Cohoon *et al*., 1991), *facilities layout* (Tam & Li, 1991), *plant layout* or *machine layout* (Hassan & Hogg, 1994; Singh & Wang, 1994), *bin-*

packing (Jakobs, 1996), *partitioning* (Moon & Kim, 1998), etc. However, we may classify LD problems into four major application categories that would include Facilities LD, Circuit LD, User Interface LD, and Cutting/Packing. A brief description of the significance and prevalence of the LD problem within these contexts is provided here.

2.2.1 Facilities Layout Design

In facilities LD, various activities and components are allocated spaces in the given periphery (Abdinnour-Helm & Hadley, 2000). The layout of facility establishes the physical relationship among activities and their objectives (Badiru & Arif, 2000; Welgama *et al.,* 1995). It may also have profound effects on such relatively intangible matters as environment and safety. Consequently, these space allocation decisions are based on various commutation, communication, political, social, environmental, and safety considerations (Meller & Gau, 1996). Indeed, an adequately designed facility layout improves the efficiency, efficacy, productivity, and profitability of an organization (Norman & Smith, 2002). The relative permanency of outcome and the scale of strategic investment stipulations mean more efforts have been dedicated to facility LD than any other LD area.

2.2.2 VLSI Circuit Layout Design

The design of VLSI microchips involves several phases including specification, functional design, circuit design, physical design, and fabrication (Mazumder & Rudnick, 1999). An important step in physical design is the macrocell placement based on a range of subjective and conflicting preferences and constraints (Moon & Kim, 1998; Kang *et al.*, 1994; Murata *et al.*, 1996). Macrocells are the circuit components lumped together in functional entities with connection terminals along their borders. These terminals are required to be connected by signal nets, along which signals or power is transmitted among the various components. As such, the macrocell placement also characterizes routes selected for the signal nets. Generally, the fitness metric is the space utilization in terms of the area of the enclosing rectangle, where the enclosing rectangle is the smallest rectangle that completely encompasses the whole layout (Ahmad *et al.*, 2004b). The area of the enclosing rectangle is also dependent on the routing space, i.e., the space between the macrocells occupied by the signal net wirings. Nevertheless, due to combinatorial complexity, the placement phase is typically kept independent of the computation of actual routes for the signal nets (Mazumder & Rudnick, 1999). Consequently, during the macrocell placement phase, an estimated amount of routing space or white space is added between the cells. These estimates for inter-module separation are mainly determined using experts' judgments and heuristics.

Estimates for inter-module separation are critical for the quality of solution and typically based on subjective, uncertain, and incomplete information. Moreover, several critical design and performance parameters are inherently dependent on the block layout of macrocells. For instance, the capacitive crosstalk in interconnect is the foremost factor in signal delay and results in a variety of signal integrity problems (Mazumder & Rudnick, 1999; Youssef *et al.*, 2003a). Indeed, Macrocell Layout-based solution for signal integrity is an important consideration in chip design (Murata *et al.*, 1996). In reality, the macrocell placement phase might be repeated several times before a workable solution can be found. In short, increasing innovations, complexities, and miniaturizations in VLSI designs underscore the need for continuously advancing layout optimization techniques (Khan & Sait, 2002).

2.2.3 User Interface Layout Design

The physical manifestation of a user interface communicates structure, purpose, operations, and significance of the underlying system (Ahmad *et al.*, 2004c; McTear, 2000). Conceivably, a good interface layout performs a crucial role in the success of the whole system (Ngo & Law, 2003). Users of various human-machine interfaces come from different backgrounds. Moreover, the user behavior evolves with the change in context of use and the cognitive progress of the user (McTear, 2000; Ngo, 2001). Such a diverse and evolutionary nature of the user behavior signifies the need for adaptive user interfaces for meeting user needs and expectations in both layout structure and functionalities (Akoumianakis, 2000).

Adaptive User Interfaces have a long history rooted in the emergence of such technologies as Artificial Intelligence, Soft Computing, Graphical User Interface, Intelligent User Interfaces, JAVA, Intelligent Multimedia Educational Systems, Internet, and Mobile Services (Brusilovsky & Maybury, 2002; Conati *et al.*, 2002). More specifically, the advent and advancement of the Web and Mobile Services have brought forward adaptivity as an important issue for both efficacy and acceptability of such services (McTear, 2000). However, such applications constitute an extremely difficult class of LD problems due to the particularly diverse target population (Brusilovsky, 2001). Nevertheless, it has been demonstrated that good user-modeling endeavors can be very robust over a diverse population and may assist in customizing interfaces (Albrecht *et al.*, 1999). Conceviably, automated tools would play important role in this area (Corbett *et al.*, 2000). Extensive discussions on related issues are available in the literature (Ahmad *et al.*, 2004e; Brusilovsky, 2001; McTear, 2000).

2.2.4 Bin-Packing

The bin-packing problem is directed at packing the maximum number of items in the smallest number of specified size bins (Dyckhoff, 1990). As such, the typical goal is to maximize the space utilization

(Kim *et al.*, 2001). There are several variants of this general bin-packing problem based on dimensions and shapes of modules such as linear, 2D, 3D, Polygon Packing, Circle packing, fixed or flexible shaped modules, irregular shaped modules, etc. Furthermore, variation in performance criteria results in such variants as minimize volume, minimize weight, maximize total utility, maximize performance, etc. (Burke *et al.*, 2004; Dowsland *et al.*, 2002; Garey & Johnson, 1979; Wang *et al.*, 2002).

In this thesis, we limit ourselves to the discussion of oriented and orthogonal two-dimensional rectangular packing problem (2D-BPP). It is because 2D-BPP forms a generic formulation that can readily be adapted to many prevalent LD contexts (Dyckhoff, 1990; Garey & Johnson, 1981). Furthermore, it requires minimal post-optimization processing compared to other existing formulations (Ahmad, 2002; Mir and Imam, 2001). In oriented and orthogonal 2D-BPP, the packing of rectangular modules with definite shape and orientation is sought in a given two-dimensional packing space with edges of modules parallel to edges of the packing space. This problem is relevant to various cutting, packing, storing, transporting, and scheduling functions of businesses (Dyckhoff, 1990; Islier, 1998; Lodi *et al.*, 2002; Marten, 2004). Intriguingly, it has such seemingly unrelated applications as dynamic memory allocation and multi-processor scheduling in improving performance and utilization of computational resources (Ahmad *et al.*, 2004b,e; Burke *et al*, 2004). However, it should be noted that the 2D-BPP is also strongly NP-Hard (Ahmad, 2002; Garey & Johnson, 1979). As such, our emphasis is on efficient and robust heuristic, metaheuristic, and knowledge-based solution approaches.

2.3 Mathematical Formulations

A range of formulations for the LD problem has been proposed in the literature. A good account of such formulations can be found in Bozer & Meller (1997). The most popular of such formulations include the *Quadratic Assignment Problem* or QAP (Bazaraa, 1975), the *Quadratic Set-Covering* problem or QSC (Bazaraa, 1975) and the *Two-Dimensional Bin-Packing Problem* or 2D-BPP (Heragu & Kusiak, 1991). Here we briefly describe these formulations with their pros and cons.

2.3.1 Quadratic Assignment Problem

Quadratic Assignment problem (QAP) formulations in LD deal with scenarios involving location of interacting modules of equal area. It has been applied to a variety of LD applications. This approach works by assigning one module to every location and at most one module to a given location. The cost of positioning a module at a given location is deemed dependent on the location of interacting modules resulting in a quadratic objective function.

Some variations of QAP involve assigning interdepartmental distances to department pairs as a measure of inter-module interaction. Nevertheless, the underlying principle remains the same and the problem seeks an optimal mapping of n activities/modules to n or more locations each of which may accommodate at most one activity. Each pair of activity (i,j) has some cost due to inter-module interaction $C(i,j)$ and each pairs site (k,l) has some cost due to spatial separation $\delta(i,j)$. Furthermore, some sort of fixed cost $F(i,k)$ might be involved with the placement of activity i in site k. If $S(i)$ denotes the site to which activity i is allocated in a mapping A of activities to sites, the total cost of solution is:

$$Cost(A) = \sum_{i} F(i, S(i)) + \sum_{i}\sum_{j}\left[C(i,j)\times\delta(S(i),S(j))\right] \qquad \text{Equation 2-1}$$

Due to NP-Complete nature, it is very hard to procure a verifiably optimal solution for more than 16 modules even with very powerful computing facilities (Bazaraa, 1975; Garey & Johnson, 1979). Solution approaches to QAP are mostly based on branch-and-bound methodologies. Techniques that are more efficient involve heuristic fathoming of branches to expedite or limit the search (Kaku *et al.*, 1991; Meller & Gau, 1996).

Incidentally, QAP formulation assumes that all modules are identical in size and shape and possible locations for assignment of modules are fixed and known a priori. Nevertheless, QAP may also be employed for solving problems consisting of unequal modules by discretizing them into smaller sub-modules of equal area. The integrity of a module is ensured through assignment of an artificially very large interaction among the sub-modules (Liao, 1993; Kusiak & Heragu, 1987). Nevertheless, the increased size of the new problem means that QAP cannot even solve a problem with few non-identical modules (Gloria, 1994). In addition, it is hard to retain the original shape of the module and it often results in outcomes with irregular and unrealistic shapes, requiring substantial post-optimization processing (Mir & Imam, 2001).

Requirement of objectivity and accuracy of design preferences and fitness evaluation metrics adds to limitations of QAP. For instance, it is assumed that the interaction weight between any pair of modules is predetermined and remains fixed regardless of the arrangement of modules. It implies that the interaction between any pair of modules is independent of the interaction of the two modules with other modules. Such simplifying assumptions are not valid in most LD applications.

2.3.2 Quadratic Set-Covering Problem

The LD problem may also be formulated as a Quadratic Set Covering (QSC) problem. The main data requirements for the QSC formulation include the size of each module, candidate locations of each module, and utilities of each module. Allowing layout designers to introduce candidate locations of each module helps in eliminating undesirable placements. It also takes the advantage of the intuition and expertise of the user, while reducing computational efforts by restricting the search space. Nevertheless, QSC requires a large number of user inputs for every module under consideration (Bazara, 1973; Ligget, 2000).

Here we provide a simplified QSC formulation where we ignore the effect of mutual position of modules. We also, assume that all utilities are calculated based on a module's distance from a pre-determined focal point in the packing space as shown in Figure 2-1.

Assumptions

- The total area of packing space is divided into small blocks of equal size.
- A list of *n* modules with positive utilities is given.
- All modules are rectangular and oriented with width w_i and height h_i.
- Modules can be packed immediately one after another without any need for white space.
- Space between modules does not give rise to usability and utility considerations.
- All packing spaces are rectangular and have standard size with width W and height H.
- The geometric center of a module determines its position for utility calculations.
- The reference point for the bin is its bottom-left corner with coordinates $(x,y)=(0,0)$.
- The reference point for each module M_i is its bottom-left corner with coordinates (x_i,y_i).

Module Position

The position of any module on the packing space could be determined by measuring two parameters: the Euclidean distance (alternatively, Manhattan distance) between the point of focus on the packing space and the center of the module, denoted by δ_i, and the angular distance between the line joining the point of focus on the packing space to the center of the module and the positive x-axis, denoted by θ_i.

These two parameters can be calculated using the following formulae:

$$\delta_i = \sqrt{(x_i + w_i/2 - X_c)^2 + (y_i + h_i/2 - Y_c)^2} \quad \text{Equation 2-2}$$

$$\theta_i = Tan^{-1}\left[\frac{(y_i + h_i/2 - Y_c)}{(x_i + w_i/2 - X_c)}\right] \quad \text{Equation 2-3}$$

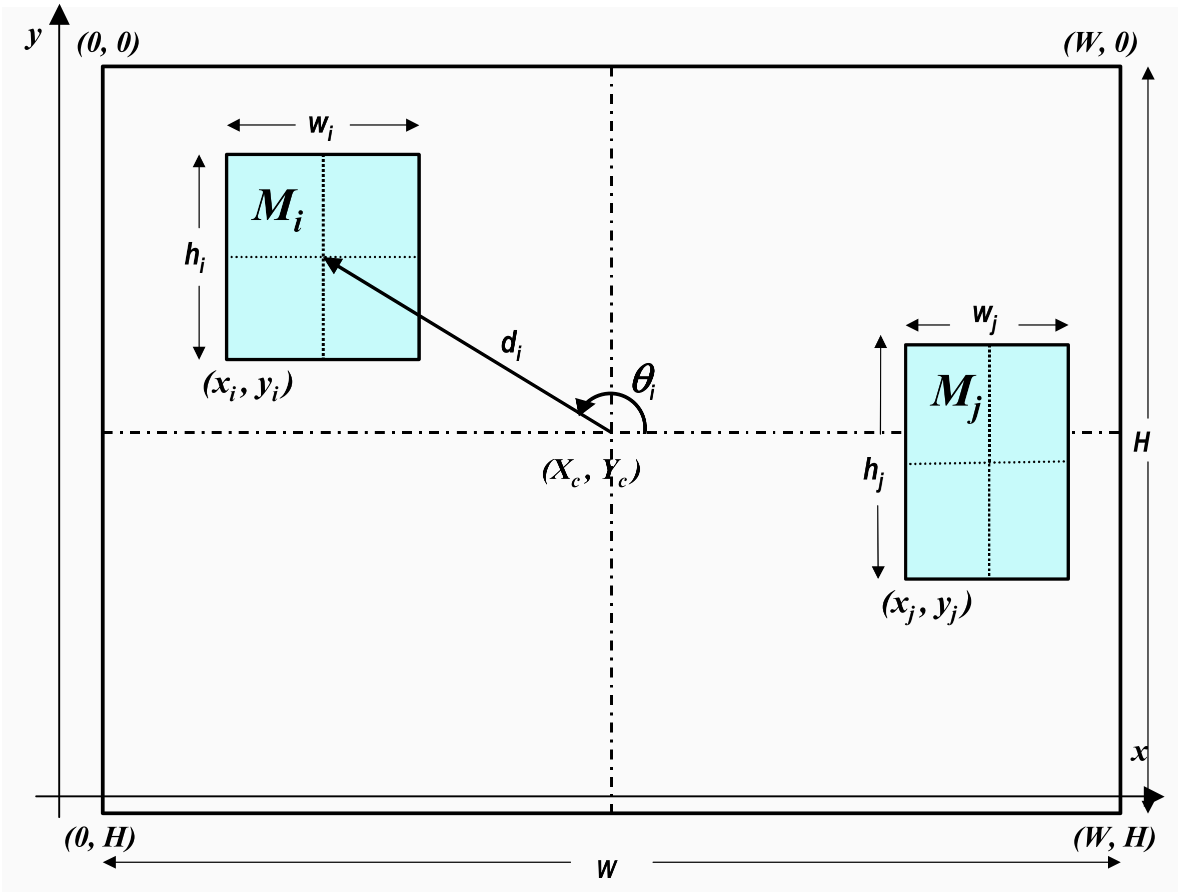


Figure 2-1: Location and Angle of approach from the point of Focus in Placement Decisions

The farthest a module could be placed from the center could be:

$$\delta_{\max} = \sqrt{(W - X_c)^2 + (H - Y_c)^2} \qquad \text{Equation 2-4}$$

Decision Variables

The x_i and yi are the two decision variables that would determine the position of any module in the packing space. Our objective is to maximize the total utility of the layout configuration as given by:

$$Max.\left[\sum_{i=1}^{n}\sum_{k=1}^{I(i)} u_i b_{ik} - \zeta\left(WH - \sum_{i}^{n} w_i h_i\right)\right] \qquad \text{Equation 2-5}$$

Given that:

$$u_i = \alpha_i (1 + \beta_i)$$
$$\beta_i = (\delta_{\max} - \delta_i) / \delta_{\max}$$

Subject to:

$$\sum_{k=1}^{I(i)} b_{ik} \leq 1 \qquad i = 1, 2, \ldots n. \qquad \text{Equation 2-6}$$

$$\sum_{i=1}^{n} L_{ikt} b_{ik} \leq 1 \qquad for\ all\ t \qquad \text{Equation 2-7}$$

$$L_{ikt} = \begin{cases} 1 & if\ t \in J_i(k) \\ 0 & otherwise \end{cases} \qquad \text{Equation 2-8}$$

$$b_{ik} = 0, 1 \qquad i = 1, 2, \ldots, n; \qquad k = 1, 2, \ldots, I(i)$$

In this QSC formulation of LD problem, the first set of constraints restricts the assignment of each module to at most one location. Whereas, the second set of constraints ensures that each location is occupied by at most one module. The objective function consists of the difference between the utility of modules and the disutility of the unused space.

2.3.3 Two-Dimensional Bin-Packing Problem

The LD problem may also be formulated as an oriented and orthogonal two-dimensional rectangular packing problem (2D-BPP). It has the advantage of maintaining the integrity and the shape of modules. Furthermore, it requires minimal post-optimization processing in comparison with other prevailing LD problem formulations. Furthermore, it constitutes a generic approach to many LD problems (Burke *et al.*, 2004; Dyckhoff, 1990; Garey & Johnson, 1981). However, the traditional bin packing formulations strive for maximizing the number of modules to be packed in the bin(s). In contrast, our objective is to maximize the total utility of the modules packed into one or more bins or packing spaces. Furthermore, a number of domain specific constraints needs to be incorporated, including consideration of relative positions of modules (Ahmad, 2002; Lodi *et al.*, 2002).

In Chapter 4, we provide a bin-packing formulation of the problem, within the context of the ILG, where a finite number of rectangular modules M_i, $i = 1, 2, 3, \ldots, n,$ and m rectangular bins are given.

2.3.4 Graph-Theoretic Formulations

Graph algorithms are formulated in terms of operations on the vertices and edges of the graph representing a layout (Foulds, 1995, 1998). In such formulations, each module is represented as a node in a graph with an arc representing the adjacency stipulations, ignoring the size and shape of the module. It is assumed that the adjacency requirements of each pair of modules are known in advance (Meller & Gau, 1996).

The objective function in such formulations is the optimization of the inter-module interaction (Foulds, 1998). It is optimized when all module-pairs with positive interactions have arcs between them, which means those are adjacent to each other. Accordingly, such approaches require constructing a dual of the problem (a graph) that maximizes the sum of pair-wise weights on adjacency (arcs) between modules (nodes). As such, these approaches are primarily relevant to such problems as minimizing material handling costs, in facility LD context, under static and objectively known preferences.

2.3.5 Constraint Satisfaction

Instead of focusing on optimization of some LD fitness objective, the constraint satisfaction approach focuses on finding an arrangement that satisfies a diverse set of constraints or relations (Ligget, 2000). Such constraints may involve factors like position, orientation, adjacency, distance, shape, etc. In short, the problem becomes finding a feasible solution. This approach has formed the basis of various small-scale research prototypes in automated LD (Hower & Graf, 1996). Some examples include General Space Solver (Eastman, 1973), Design Problem Solver (Pfefferkorn, 1975), SEED (Flemming *et al.*, 1994), etc. However, such systems do not have the ability to handle problems of any realistic scale (Borning *et al.*, 2000; Lok & Feiner, 2001; Zahn & Hower, 1996). Conceivably, this approach has not been the basis of commercial software in LD (Ligget, 2000).

2.4 Shortcomings of Existing Formulations

Existing mathematical formulations of LD problem have substantial limitations that make these formulations somewhat incompatible with most real world applications. For instance, the QAP does not allow control over the shape of modules in the resulting layout and QSC requires a large number of user inputs for every module under consideration (Deb & Bhattacharyya, 2004). These mathematical models offer little practical advantage in dealing with real layouts of any consequence due to the prohibitive size of the associated mathematical program. Such core issues as ill-structured, subjective and uncertain character of the layout preferences further exacerbate the situation (Malakooti & Tsurushima, 1989).

In addition, such mathematical programs rely on crisp values of various parameters that are, presumably, measured accurately and attributed to specific dynamics of the problem (Irani & Huang, 2000; Mir & Imam, 2001). In reality, such data is often available only for some unrealistically simplified layout planning scenarios. Consequently, these formulations are of little practical advantage when a modestly large size problem, involving subjective and uncertain preferences, is considered. Consequently, fast and efficient heuristics that consistently provide superior solutions are the major focus in this area (Burke *et al.*, 2004).

2.5 Solution Methodologies

Various heuristic and analytical techniques have been published for finding solutions to the LD problem. The heuristic techniques find solutions to the problem mostly by treating it as QAP (Bazaraa, 1975; Welgama & Gibson, 1993). The 2-dimensional plane is discretized into a grid structure resulting in high computational costs (Gloria *et al.,* 1994). Other solution approaches include tree search algorithms (Pierce & Crowston, 1971), binary mixed integer-programming (Love & Wong, 1976), and network decomposition (Mak *et al.,* 1998) etc. Here we provide a brief overview of various methodologies for solving the LD problem.

2.5.1 Traditional Approaches

The NP-Hard and subjective nature of the LD problem means that traditional hard optimization approaches do not hold much promise. However, a significant body of research is available in this area. Here we briefly discuss some existing traditional approaches to the LD problem with an emphasis on their limitations.

Graph Algorithms

The development of a layout through a graph theoretic approach involves three main steps. First, developing an adjacency graph using inter-module interactions of adjacent pairs of modules. Second, constructing the dual graph of the adjacency graph. Third, converting the dual graph to a block layout specifying actual shapes and areas of modules.

It should be noted that the combinatorial nature of the number of arcs in the second step makes the problem particularly difficult to solve. It implies that some heuristics must be employed to limit the number of arc incidents on each module. In addition, similar to the QAP approach, even a small size problem involving unequal modules cannot be solved with guaranteed optimal solution. Detailed review of such graph-search approaches and heuristics can be found in the literature (Foulds, 1995; Hassan, 1995; Hassan & Hogg, 1994).

Tree Search

Tree search methods are more relevant to constraint satisfaction style formulation of the LD problem (Hower, 1997). Such search mechanisms incrementally construct layout solutions by adding one module at a time to a partial layout while testing for any violation of feasibility constraints (Zahn & Hower, 1996).

A tree search method may employ a breadth-first search, enumerating all possible ways of adding a new module, or depth-first search, creating a full layout by placing all the modules sequentially (Akin *et al.*, 1992; Flemming *et al.*, 1992). However, such an approach is inherently inefficient and frequently requires backtracking when some feasibility constraint is violated, which adds to the computational complexity (Ligget, 2000; Zahn & Hower, 1996).

Analytical Algorithms

There are various analytical techniques dealing with continuous design space with relatively minimal computational requirements (Adya *et al.*, 2003; Mir & Imam, 19992, 1996, 2001; Tam, 1992; Welgama & Gibson, 1993). However, analytical approaches have yet to be developed to furnish results comparable to advanced heuristic and metaheuristic techniques (Mir & Imam, 2001). Nevertheless, such research provides more insights to the structure of the problem leading to advance and effective heuristics.

2.5.2 Heuristic Approaches

Decision-makers often resort to heuristics for dealing with difficult and uncertain problems. The NP-Hard and subjective nature of the LD problem suggests that heuristics can be very effective in solving the problem. Accordingly, various heuristic algorithms for solving the difficult 2D-BPP are available in the literature (Ahmad *et al.*, 2004d, 2004f; Berky & Wang, 1987; Chung *et al.*, 1982; Dowsland *et al.*, 2002; El-Bouri *et al.*, 1994; Hopper & Turton, 2001; Jakobs, 1996; Johnson *et al.*, 1982; Kim *et al.*, 2001; Leung *et al.*, 2003; Liu & Teng, 1999; Lodi *et al.*, 1999; Martens, 2004).

In this regard, the importance of effectively limiting the otherwise very large and intractable search space to some reasonable subset of possible solution topologies cannot be overemphasized (Dowsland *et al.* 2002; Hopper & Turton, 2001; Tompkins *et al*., 2002). Understandably, several effective metaheuristic solution methodologies are proposed in the literature. The core of such approaches is quite simple and involves treating the LD problem as a packing problem by defining an *ordering of modules* in the form of a sequence or permutation and a *placement* or *decoding heuristic* for placing modules in the determined order (Ahmad, 2002; Leung *et al.*, 2003). Recent metaheuristics that have shown good results for LD include simulated annealing (Adya *et al.*, 2003; Souilah, 1995), genetic algorithms (Ahmad *et al.*, 2004f; Gloria *et al.,* 1994; Martens, 2004), tabu search (Hopper & Turton, 2001), random search (Ahmad *et al.*, 2004f; Jakobs, 1996; Liu & Teng, 1999), naive evolution (Hopper & Turton, 2001), and hybrids (Lee & Lee, 2002). The key to these methods generally lies in some effective means for getting out of local minima. However, the speed and effectiveness of such metaheuristic approaches are largely determined by the speed and effectiveness of decoding heuristics (Hopper & Turton, 2001).

Earlier research on the relative performance of some of these popular metaheuristics in solving the LD problem, at best, provides mixed results (Hopper & Turton, 2001; Leung *et al.*, 2003; Youssef *et al.*, 2003b). Nevertheless, some knowledge of the merits and the demerits of these metaheuristic approaches, within the context of the LD problem, could result in a more judicious selection of optimization approach. Consequently, here we discuss some merits and demerits to provide some insights to these popular metaheuristics.

Genetic Algorithms

Genetic Algorithms (GA) are primarily used due to the non-deterministic and global optimization approach that has the potential to provide several near optimal and diverse layout alternatives (Youssef *et al.*, 2003b). Furthermore, GA allow incorporation of domain-specific knowledge into the fitness of individual solutions, which guides the search. The domain specific knowledge may also be exploited in other forms such as selection and genetic operations (Youssef *et al.*, 2003b). It should be noted that GA demand more efforts, when compared to other popular metaheuristics, in terms of complexity of implementation and tuning of parameters. Nevertheless, the GA have inherent characteristics that, if employed judiciously, may result in significant computational savings and performance gains. Moreover, GA maintain a population of solutions that are optimized simultaneously. Accordingly, it takes advantage of the experience gained from past explorations and directs more extensive search, or exploitation, of promising regions in the solution space, while the mutation operator provides a mechanism for escaping local optima. (Sait *et al.*, 2003). In this regard, detailed discussion of philosophical, theoretical, and practical aspects of GA can be found in the literature (Goldberg, 1989).

GA have been applied to the LD problem in various ways. However, much of the research deals with relatively simple problems requiring assignment of identical modules to given locations. Comparative studies of GA with other metaheuristics show superiority of GA in LD (Hopper & Turton, 2001). Consequently, GA provide a very promising approach for LD through generation of a diverse set of superior alternatives (Ahmad *et al.*, 2004b; Lee & Lee, 2002; Martens, 2004; Moon & Kim, 1998; Sait *et al.*, 2003). Some favorable characteristics of GA within the context of LD are further discussed in Section 2.8.1.

Simulated Annealing

Simulated Annealing (SA) is a well-known, high-performance, and effective stochastic optimization technique for combinatorial problems (Mir & Imam, 2001; Souliah, 1995). SA is motivated by an analogy to a phenomenon in crystallization. It is very effective in solving complex and large LD problems (Hopper and Turton, 2001; Tompkins *et al*., 2002). Any domain specific knowledge is incorporated mainly in the SA cost function (Youssef *et al.*, 2003b). SA starts with a random solution and makes incremental refinements by moving genes from their current location to new locations, generating new solutions. Moves that decrease the cost are accepted while moves that increase the cost are also accepted with a probability that decreases exponentially with time. Accordingly, in the beginning many high cost or inferior solutions are accepted and, subsequently, fewer high cost solutions are accepted. Thus, it avoids being trapped in a local optimum by accepting inferior solutions, too.

SA is known to be a stable metaheuristic approach capable of finding a global optimal solution (Youssef *et al.*, 2003b). However, SA is generally very slow to converge to good solutions when compared to GA. Nevertheless, when sufficient time is available, SA may provide solutions comparable to or marginally better than GA (Hopper & Turton, 2001; Youssef *et al.*, 2003b). However, the downside is that SA operates on only one solution at a time and has a meager history or memory for learning from past explorations. In short, SA can be characterized as a serial algorithm that is not easily amenable to parallel processing without significant communications overhead. Another implication is the production of closely related solutions, eluding the requirement of having both superior and diverse layout alternatives (Ahmad *et al.*, 2004b).

Tabu Search

Tabu Search (TS) is another successful, effective, and robust metaheuristic approach for solving complex combinatorial and continuous optimization problems (Youssef *et al.*, 2003b). Tabu search has a huge range of sophistications and adaptations in many of its applications. However, the generic TS is an iterative procedure that starts from some initial feasible solution and attempts to determine a better solution. It works by making several neighborhood moves. The set of admissible solutions explored at a particular iteration forms a candidate list and TS selects the best solution from the candidate list. The candidate list size is a trade-off between quality and performance (Wang *et al.*, 2002).

A distinguishing feature of TS is its exploitation of an adaptive and explicit form of memory in the shape of a tabu list, which is used to prevent back cycling and influence the search (Youssef *et al.*, 2003b). The tabu list is analogous to a window on accepted moves (Sait *et al.*, 2003). In short, tabu restrictions permit the search to go beyond the points of local optimality while making the best possible move.

Naive Evolution

The rationale behind Naive Evolution (NE) search is somewhat similar to that of GA. However, it employs only a mutation operator in order to generate successive populations of solutions (Youssef *et al.*, 2003b). Understandably, it is very easy to implement. However, NE lacks the structured search engendered by crossover operators in GA. Nevertheless, the complexity and subjectivity involved in most LD applications mean that the even NE may turn out to be an effective and efficient search strategy (Hopper & Turton, 2001).

Random Search

Random Search (RS) is another naive search strategy where the ordering of modules is generated randomly (Ahmad *et al.*, 2004c, 2004f; Hopper & Turton, 2001). Again, the subjectivity and complexity in most LD applications mean that an RS strategy could result in quite superior outcomes. However, the superiority of such solutions does not match to those generated by such advanced metaheuristics as SA and GA (Youssef *et al.*, 2003b).

2.5.3 Decoding or Placement Heuristics

As already mentioned that the GA and other metaheuristic based solution approaches to the LD problem require effective and efficient placement or decoding heuristics for determining the physical position of modules in the resulting layout configuration. In effect, a module placement algorithm takes one gene (or module) at a time from a chromosome (or sequence of modules) and determines its position in the packing space based on pre-specified steps. The position of modules in the layout is used for calculating the fitness of the sequence of modules generated by the metaheuristic. Such decoding heuristics are usually designed to realize some local search improvements (Healy *et al.*, 1999; Wu *et al.*, 2002).

Indeed, it has been argued that the computational cost of such a metaheuristic-based layout optimization process is determined by the cost of the decoding heuristic, as it is executed every time the solution quality is evaluated (Burke *et al.*, 2004). Consequently, an efficient module placement strategy that generates superior quality layouts is critical for the efficacy of such an endeavor (Dowsland *et al.*, 2002).

Keeping the importance of placement heuristics in perspective, we discuss some of the simplest as well as the most popular and efficient decoding heuristics and their deficiencies. Inquisitive readers may find details of various heuristics in the literature (Hopper & Turton, 2001; Lodi *et al.*, 1999; Leung *et al.*, 2003; Lodi *et al.*, 2002; Wu *et al.*, 2002). However, the majority of heuristics are either not very effective or inappropriate due to speed and scalability issues as well as tedium involved in comprehending and implementing those. Indeed, the efficiency and efficacy of such algorithms are also known to deteriorate drastically with increase in the problem size (Burke *et al.*, 2004; Hopper & Turton, 2001; Wu *et al.*, 2002). Time complexities of such algorithms are also quite prohibitive. For instance, the time complexities of a couple of Least Flexibility First (LFF) algorithms proposed by Wu *et al.* (2002) are $O(n^4 \log n)$ and $O(n^5 \log n)$. Similarly, the Bottom-Left Fill (BLF) algorithm originally proposed by Chazelle (1983) has a time complexity of $O(n^3)$ and for large-scale problems it provides results comparable to our proposed algorithms even after 50,000 evaluations against less than 1,000 evaluations of proposed algorithms. The Difference Process (DP) strategy is another decoding heuristic, but it is not designed to take advantage of any non-rectangular empty spaces in the partial layout configuration (Leung *et al.*, 2003). As such, even for moderately sized problems, about 25 modules, the trim loss exceeds 6% and the performance declines drastically with the increase in the problem size. In addition, there exist such efficient greedy algorithms as Best-Fit and its adaptations (Burke *et al.*, 2004; Garey and Johnson, 1979). However, such greedy algorithms do not meet the requirements of most LD applications and are not suitable for some generic approach to the problem (Dowsland, *et al.*, 2002).

Consequently, here we limit ourselves only to the discussion of most efficient, effective, and documented decoding heuristics, namely Bottom-Left, Improved Bottom-Left, and Bottom-Left Fill (Burke *et al.*, 2004; Dowsland, *et al.*, 2002; Hopper & Turton, 2001). In Chapter 4, we provide some new decoding heuristics and demonstrate their efficiency, effectiveness, and robustness through an extensive/multi-facet comparison regime.

Bottom-Left Algorithm

The Bottom-Left placement algorithm or BL has drawn considerable attention from researchers (Ahmad *et al.* 2004d; Baker *et al.*, 1980; Chazelle, 1983; Dowsland *et al.*, 2002; Healy *et al.*, 1999; Hopper & Turton, 2001; Jakobs, 1996; Lai & Chen, 1997; Liu & Teng, 1999). It calls for placing a module at the bottom-most and left-most feasible position through successive vertical and horizontal movements of the module. Starting from the top-right corner of the packing space, each module is pushed as far as possible to the bottom and then as far as possible to the left of the packing space

(Jakobs, 1996). These operations are repeated until the module occupies a stable position from where it can be moved neither to the bottom nor to the left.

The basic premise in BL is quite long-standing and widespread. The apparent advantages of such approaches include speed and simplicity (Dowsland *et al.* 2002). However, BL tends to leave holes in the packing pattern resulting in poor space utilization. Nevertheless, the interest in the BL placement strategy has been boosted by modern metaheuristics like TS, SA, and GA (Hopper & Turton, 2001).

Improved Bottom-Left Algorithm

Various improvement schemes have been proposed for the BL such as the Improved-BL heuristic or IBL (Liu & Teng, 1999). Such improved strategies comprise of refinement of placement decisions in BL by conferring precedence to a shift towards the bottom and some allowance for module rotations. However, even these improvised strategies encounter such problems as dead-area and inferior aesthetic contents. Nevertheless, such improvement schemes are quite popular and deemed reasonably efficient and effective. Consequently, we have included IBL in our comparison analyses.

Bottom-Left Fill Algorithm

Bottom-Left Fill (BLF) is another more sophisticated version of BL heuristic. It attempts to fill empty spaces by placing a rectangle into the lowest available position and left-justifying it. As such, it is capable of filling existing gaps in the packing pattern and results in denser packing. Here a list of location points in a bottom-left ordering is maintained to indicate candidate placement locations. The algorithm starts with the lower-most and left-most point, places the module, and left justifies it. It then checks for overlap and boundary conditions. If no violations occur then the module is placed and the list of candidate placement locations is updated. However, if an overlap occurs the next point in the list is tested and the process is repeated until the module can be placed without an overlap. Consequently, BLF overcomes the problem of poor space utilization in BL or IBL. Nevertheless, as mentioned, the major disadvantage lies in its $O(n^3)$ time complexity without furnishing significant improvement in aesthetic contents of solutions (Burke *et al.*, 2004; Chazelle, 1983; Hopper & Turton, 2001).

Deficiencies of BL, IBL, and BLF

The BL and the IBL are overly simplistic placement strategies with such inherent deficiencies as poor space utilization and inability to obtain some simple optimal solutions. The poor space utilization is also evident from Figure 2-2 that shows creation of dead area that cannot be used by the BL or the

IBL through subsequent placements. Furthermore, if we know an optimal packing pattern of *n* modules that fulfills the BL-condition, we cannot always write out a permutation for the BL-algorithm corresponding to it (Jakobs, 1996). In other words, the optimal packing configuration cannot be obtained by the BL-algorithm even if all permutations are enumerated, as the case with the layout shown in Figure 2-3. In addition, the BL, the IBL, and the BLF are not very effective in incorporating such qualitative design considerations as symmetry, balance, equilibrium, cohesion, etc.

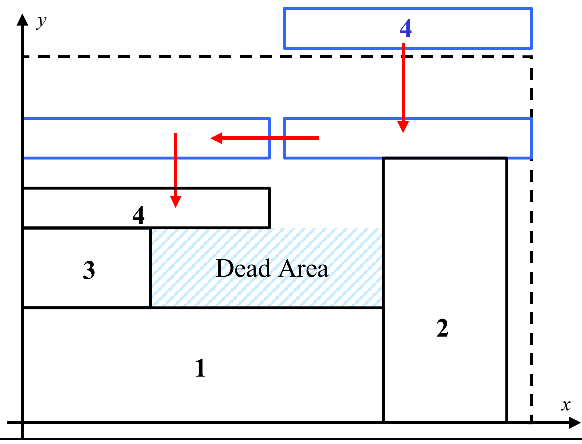


Figure 2-2: Poor Space Utilization with BL.

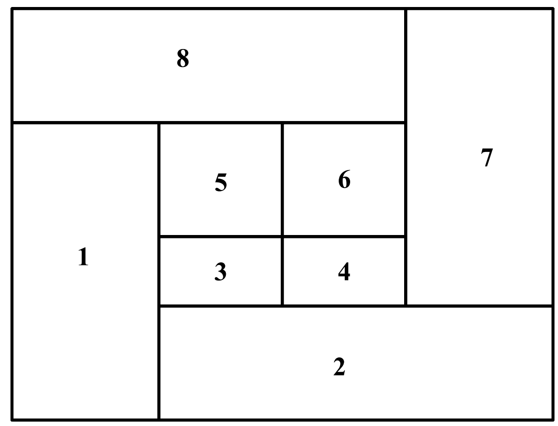


Figure 2-3: A Simple Pattern Not Possible with BL.

It should also be noted that the BL and its adaptations are more appropriate for cases where the objective is directed at minimizing the height of the packing pattern; for instance, stamping patterns out of a fixed width roll of a steel sheet. Under such fitness objective if two configurations have the same height then their fitness values are same. Nevertheless, one of the packing patterns can be deemed superior to the other based on other objectives like the sum of inter-module distances. In addition, the BL converges modules at the bottom-left corner of the packing space, which might not be a useful strategy in many cases. For instance, some facility LD application may require modules near/around some focal point(s).

The BLF provides better space utilization than BL and IBL at the expense of computational time. Nevertheless, layouts generated by BL, IBL, and BLF lack aesthetic value with BLF faring better than BL and IBL. Consequently, the quest for efficient and effective module placement strategies is an interesting and popular research direction (Burke *et al.*, 2004).

2.6 Subjectivity and Uncertainty in LD

Despite being an active research area, LD is still a vaguely and inadequately defined field and the existing literature offers LD guidelines and algorithms largely without a thorough scheme for utilizing those (Abdinnour-Helm & Hadley, 2000; Khan & Seit, 2002; Youssef *et al.*, 2003b; Zhang *et al.*, 2002). The usefulness of such vastly scattered knowledge is further limited by the cognitive capacity of layout designers.

2.6.1 Sources of Uncertainty

Subjective and uncertain considerations intrinsic to LD preferences stem from multiple sources. Detailed discussions on such considerations in various LD contexts can be found in the literature (Ahmad, 2002; Ahmad *et al.*, 2004c; Head & Hassanein, 2002; Jain & Krishnapuram, 2001; Karray *et al.*, 2001b; Mazumder & Rudnick, 1999; Youssef *et al.*, 2003a). However, here we briefly mention a few instances of such subjective, uncertain, and inconsistent preferences, in various LD contexts, for elaboration purposes.

In Facility LD context, layout planning requires a priori specifications of the objectives of the facility, its primary and support activities, the interrelationships among those activities, and the space requirements of all activities (Benjaafar, 2000). Aside from this largely abstract definition process, preferences obtained from experts about inter-module relationships are inherently linguistic and uncertain, which are amenable to different interpretations by different people at different times. Such preferences encompass safety, structural integrity, operability, maintainability, flow relationships, control relationships, process relationships, organizational or personnel relationships, political considerations, and environmental relationships (Baykasoglu & Gindy, 2001; Tompkins *et al.*, 2002).

In Circuit LD context, almost every preference in the macro cell placement is in conflict with some other equally critical consideration (Khan & Sait, 2002; Youssef *et al.*, 2003a; Mazumder & Rudnick, 1999). Furthermore, these considerations are often vague and inadequately defined. For instance, one consideration in reducing circuit delays and increasing the speed of the circuit is to keep circuit components close to each other, which may simultaneously deteriorate the circuit performance through crosstalk and overheating. Some other examples of subjective and uncertain considerations in a circuit layout configuration are wire length, wire congestion, power dissipation, circuit delays, crosstalk, layout width/area (Khan & Sait, 2002; Murata *et al.*, 1996; Schnecke & Vonberger, 1995; Youssef *et al.*, 2003a). Indeed, such parameters could only be imprecisely estimated before finalizing the ultimate VLSI circuit design (Youssef *et al.*, 2003a).

A few representative instances of subjective preference parameters in the User Interface LD context would include the amount of white space, symmetry, color scheme, chronological value, intrinsic utility, size of modules, location of modules, etc. The diversity in cultural and emotional preferences of users makes it a particularly difficult problem (Akoumianakis *et al.*, 2000; Head & Hassanein, 2002; McTear, 2000). Furthermore, the potentially diverse scope of such applications means that these considerations cannot be static across demographic and temporal dimensions (Brinck *et al.*, 2000; Brusilovsky, 2001).

In general, the quality of an optimal or superior layout is always judged using some fitness metric. Nevertheless, optimality itself is a quite subjective notion that refers to the most desirable outcome under specific constraints and evaluated using some specific criterion. These LD fitness metrics may involve a wide range of perspectives that may be varying, overlapping, and even conflicting (Singh & Wang, 1994). The ensuing lack of consensus on accommodating conflicting requirements makes the task particularly difficult. Consequently, the diversity, volatility, and subjectivity of LD objectives make consideration of the entire spectrum of goals beyond the cognitive and functional capabilities of decision makers as well as information processing capabilities of most automated LD system (Abdinnour-Helm & Hadley, 2000; Tompkins *et al.*, 2002; Zhang *et al.*, 2002). Furthermore, these complexities render the solution space quite noisy, underscoring the need for a composite or multi-criteria fitness evaluation regime combining various desirable characteristics (Schnecke & Vonberger, 1997). Recently, various composite multi-criteria fitness measurement schemes in LD have been proposed for obtaining more encompassing layout fitness evaluation regimes (Ahmad *et al.* 2004b; Khan & Sait, 2002; Youssef *et al.*, 2003b).

2.6.2 Classification of Uncertainty

A generic classification scheme of subjective and uncertain preferences, constraints, and fitness metrics could serve as a sound basis for choosing an appropriate modeling and reasoning mechanism for a specific consideration. As such, we classify subjectivity and uncertainty in the domain knowledge into categories, namely, incomplete, inconsistent, imprecise, and vague knowledge (Ahmad *et al.* 2003; Negnevitsky 2002). The term *incompleteness* suggests the unavailability of some of the information and necessitates the use of rules of thumb and approximate reasoning. *Inconsistency* indicates the difference or conflict in the knowledge elicited from experts highlighting the problem in transforming the available information into working rules and guidelines. *Imprecision* refers to values that are imprecisely or loosely defined or measured inaccurately. *Vagueness* points towards the subjectivity in the estimate about some value or rule and underscores the impediments in appropriately interpreting the available information. In general, the LD guidelines, preferences, and constraints are intrinsically incomplete, inconsistent, imprecise, and vague (Abdinnour-Helm & Hadley, 2000; Ahmad *et al.*, 2004c; Youssef *et al.*, 2003a).

Any layout design and evaluation mechanism requires robust ways of coping with such uncertainties (Chung, 1999). Nevertheless, the majority of theories and tools devised to handle subjectivities and uncertainties in information are quantitative in nature. In general, such tools do not afford the subjectivity and uncertainty in information that falls into more than one of the aforementioned classes. In the next section, we provide a brief overview of uncertainty management techniques employed in LD with some pros and cons. However, a somewhat detailed survey of uncertainty management techniques in a generic LD context can be found in Ahmad *et al.* (2004c).

2.6.3 Traditional Approaches to Uncertainty Management

It should be noted that such a high degree of complexity, subjectivity, and ambiguity is encountered in a variety of other work domains and extensive research has been done on modeling of subjective preferences. However, some peculiar characteristics render LD problem distinct from other subjective problems. For instance, it is often effortlessly easy for a domain expert to judge the quality of a layout by just taking one glance (Berkun, 2001). Unfortunately, there is a relative dearth of research literature on modeling of subjective design considerations within the context of the LD problem (Whyte & Wilhelm, 1999a). Likewise, there is little work on integrative or comparative review of techniques available for tackling such issues.

Incidentally, the first research work employing some stochastic parameters in LD is attributed to Shore & Tompkins (1990) who studied various discrete scenarios using lowest likelihood penalty. This notion of multiple discrete scenarios, or layout alternatives, formed the core of the research in stochastic layout planning. Since then, there were several publications employing the idea of uncertainty in the LD preferences and constraints (Cheng *et al.*, 1996; Kouvelis *et al.*, 1992; Norman & Smith, 2002; Rosenblatt & Kropp, 1992). However, the uncertainty tackled in such research is largely limited to the uncertainty in the occurrence of a well-defined event such as uncertainty in demand forecasts. Nevertheless, LD problems also involve uncertainty in the event itself due to incomplete, imprecise, inconsistent, and vague information.

As stated, uncertainties and dynamics of the LD problem require a methodology pertinent to incomplete, imprecise, inconsistent, and vague preferences and rules. However, the majority of existing preference modeling techniques fail to deliver in uncertain environments falling in more than one of these classes. This shortcoming is more evident and imperative under incomplete information (Abdinnour-Helm & Hadley, 2000; Zhang *et al.*, 2002). In addition, most uncertainty management techniques are suitable only when deterministic data is reliably available and assignable to specific dynamics of the design process. Furthermore, the majority of such techniques are ad hoc in the sense that there is no underlying theory to support them. Such methodologies may only be validated through empirical testing.

A sound understanding of existing and promising techniques could motivate research in uncertainty in LD preferences, constraints, and fitness objectives. As such, we present an overview of modeling techniques for subjective and uncertain preferences in LD.

Deterministic Approaches

Deterministic approaches work under the simplifying assumption that all subjective preferences can easily be quantified and made available when needed. Such approaches usually use some arbitrary default or expected values, which are possibly further refined by the user during interaction with alternative solutions (Tompkins *et al.*, 2002). For instance, some extended formulations of the unequal area facilities LD problem are available that explicitly consider uncertainty in information by using expected values of forecasts. However, even under relatively deterministic environment, the cost of procuring exact and complete information could often be prohibitive (Ahmad *et al.*, 2004c; Francis & White, 1992; Tsuchiya *et al.* 1996). Furthermore, most LD applications are so intricate that the validity of these deterministic approaches could easily be disputed. Conceivably, these myopic approaches are ineffective in such complex and dynamic areas as LD. Paradoxically, the major portion of relevant literature builds on such incredibly simplifying assumptions (Ahmad *et al.*, 2004b, 2004c; Ligget, 2000; Francis & White, 1992; White & Taket, 1994).

User-Oriented Approaches

Similar to the deterministic approach, many existing models in LD rely on manually procuring weights, preferences, and properties through user inputs (EOS, 2005; Unigraphics, 2005). Although such an approach renders a higher degree of flexibility and control, users are usually overwhelmed by the flood of data and domain specific knowledge. Consequently, the usefulness of such approaches is severely limited in wake of prevailing cognitive, economic, ergonomic, and other resource constraints. Moreover, at times, it is not possible to get access to a domain expert for procuring the required inputs. Consequently, we deem the user-supplied approach as inflexible and counter productive.

Constraint-Based Approaches

Constraint processing techniques such as knowledge representation and inference mechanism have extensively been used for automated graphical LD (Hower & Graf, 1996). Research suggests that constraints provide a powerful yet simple formalism for specifying preferences in such dynamic domains as the Web page LD (Hurst, 2003). It has been argued that constraints like semantic-pragmatic, inter-relational, spatial, temporal, etc. can be represented in terms of equalities and inequalities permitting a flexible, intuitive, and declarative representation of complex preferences (Graf *et al.*, 1998). Such an approach could result in simplification of perceived informal domain features (Hurst, 2002). For instance, the relationship between two certain categories of modules can be specified using a single set of constraints instead of delineating it for each pair of modules. Consequently, this approach is close to the knowledge-based approach in a broad sense. Nevertheless, the goal is feasibility and not satisfiability or satisficing, as the general case in knowledge-based approaches. Although the work in this area is largely theoretical, inquisitive readers can find extensive surveys of constraint-based approaches in the literature (Hower & Graf, 1996; Hurst, 2002).

From-To Chart (FTC)

From-To Chart (FTC), or Flow Matrix, is one of the earliest tools adopted for assisting layout designers (Francis & White, 1992). It normally contains numbers representing some measure of interactions between pairs of modules. For instance, it could contain some measure of material, personnel, and information flow between two departments in the facility LD context. These FTC values are ultimately translated into some sort of proximity measure or Closeness Rating (Heragu, 1997). Despite being intended as a tool for representing quantitative values, FTC has often been used for representing qualitative values, as well (Heragu, 1997). However, it is a widely accepted premise that even the values that are generally considered quantitative in nature are not easily quantifiable due to subjectivity and uncertainty in the collection and processing of such data (Ahmad *et al.*, 2004b, 2004c; Francis & White, 1992; Heragu, 1997; Liggit, 2000; Tompkins *et al.*, 2002). Consequently, FTC provides a very rigid and myopic solution. Nevertheless, FTC was one of the earliest tools used to provide necessary inputs in a simplified form for computerized LD systems.

Relationship Chart (REL)

Activity Relationship Chart (REL) is among the earliest and most popular tools for expressing subjective, uncertain, and linguistic considerations in LD (Tompkins *et al.*, 2002). Typically, activity relationships are translated into some relative proximity requirement between pairs of modules for use in placement decisions. These proximity requirements are expressed in REL through a closeness rating such as: *A* (**A**bsolutely necessary), *E* (**E**specially important), *I* (**I**mportant), *O* (**O**rdinary proximity is all right), *U* (**U**nimportant) and *X* (**U**ndesirable). In short, REL contains ordinal proximity rating information for evaluating the utility of layouts in the form of a Total Closeness Rating (Tompkins *et al.*, 2002).

The REL is originally designed to facilitate consideration of qualitative factors, political needs, or dynamic situations where precise data cannot be made available due to temporal, financial, and other practical constraints (Francis & White, 1992). Yet, the underlying idea remains deterministic and crisp ratings do not provide a means for handling conflicting and inconsistent preferences. For instance, one expert could assign an *A* rating for a certain pair of modules while a second expert might provide a *U* rating for the same pair of modules creating a conflict. Moreover, there is no elaborate methodology to work with incomplete and/or dynamic preferences. Furthermore, the multiplicity of inter-module interaction modes requires separate REL for every dimension of interaction contributing to its inflexibility and limitations (Tompkins *et al.*, 2002). However, several innate advantages of REL such as ease of use, ease of understanding, and the structured nature have positively contributed to the wide acceptance of its various adaptations.

An interesting extension in this direction is the use of Fuzzy REL charts (Blair & Miller, 1985). Fuzzy Logic provides excellent means for tackling such inconsistencies in the information. In f-REL, fuzzy inferencing mechanism is used to generate activity relationship charts. Some small-scale and tightly defined simulation studies have demonstrated the effectiveness of f-REL charts in generating superior layouts against both fuzzy and non-fuzzy fitness metrics (Dweiri, 1999). However, REL or f-REL cannot encompass all subjective and uncertain preferences, constraints and fitness objectives in LD (Ahmad *et al.*, 2004c; Deb & Bhattacharyya, 2004). Consequently, we believe that a general fuzzy logic based approach, encompassing modeling, analytic, and algorithmic aspects of design process, would deliver most.

Computer Simulations

The typical absence of some encompassing, closed-form, and analytical fitness functions in LD render computer simulations a useful alternative (Azadivar & Tompkins, 1999; Gupta, 1986). Such an approach would provide detailed analysis, modeling, and evaluation of complex LD problems (Azadivar & Wang, 2000; Bookbinder & Higginson, 1986; Grobley, 1986). However, simulation models are not easily amenable to optimization, and amendments, and make procurement of a superior layout alternative difficult to achieve (Chan *et al.*, 1995). Recently, some efforts have been made to optimize LD simulation models using GA in various facility LD contexts to expedite the process and procure a diverse set of superior LD alternatives (Azadivar & Wang, 2000). Nevertheless, computer simulations are usually very time consuming and become prohibitive in the LD process.

2.6.4 Soft Computing Approaches to Uncertainty Management

Most existing methodologies for handling uncertainties in the domain knowledge through approximate reasoning are mainly quantitative in nature. In such approaches, uncertainties are quantified in the form of some measures that are propagated during reasoning (Jameson, 1996; Zukerman & Albrecht, 2001). Examples include the Bayesian Networks, Certainty Factors, Dempster-Shafer, etc. The key issue in using such quantitative approaches is accurate representation of the probabilistic, or otherwise, dependencies in the task domain. Here we describe some popular soft computing approaches to uncertainty management.

Bayesian Networks

Bayesian Networks (BN) are a popular formalism for establishing policies in handling uncertainty in information. BN is a probabilistic approach based on Bayes' theorem in which evidence is encoded in a directed acyclical graph with nodes corresponding to variables and links corresponding to probabilistic influence relationships (Conati *et al.*, 2002; Mayo & Mitrovic, 2000). It requires a very large number of probabilities and, hence, large number of experiments. Moreover, such a probabilistic approach is suited when there is uncertainty in the occurrence of the event (Albrecht *et al.*, 1999; Bunt & Conati, 2003). Alternatively, such conditional probabilities could be estimated by domain experts or obtained through some general theory about the interactions among variables. However, ad hoc estimations of such conditional probabilities by human experts are often inconsistent and biased (Bianchi-Berthouze *et al.*, 1999). Moreover, the approach is valid only under the simplifying assumption that the presence of evidence also affects the negation of conclusion, which is often an invalid assumption (Negnevitsky, 2002). Furthermore, people are not reliable Bayesian reasoners and are prone to discount older information and accord more weight to more recently presented evidence, commonly termed as *availability heuristic* (Tversky & Kahneman, 1973). Moreover, people are over-confident in judgments and have inadequate comprehension of sampling and probability theory (Tversky & Kahneman, 1974).

Furthermore, such an approach requires a large number of inputs from experts making knowledge elicitation both a tedious and an expensive enterprise. Indeed, the computational complexity of BN is often prohibitive and representing a realistic problem solution could be quite large. In fact, it has been shown that the exact application of the BN technique is an NP-hard problem (Jameson, 1996). Under dynamic conditions, the size and topology of the networks may hamper updating BN in real time. Moreover, even a small change in the knowledge representation may affect, and require updating of, a large number of sub-networks (Jameson, 1996). Approximation techniques for applying BN can be useful; however, such techniques are effective only under specified conditions. In addition, the BN is rather inappropriate for providing explanation facilities, which are deemed essential in such knowledge-intensive and uncertain domains (Jameson, 1996; Negnevitsky 2002).

Certainty Factors

Certainty Factors (CF) is another quantitative modeling approach that attempts to address such problems as the need for repeated experiments required in estimating probabilities in the BN. In CF, the knowledge is expressed in the form of rules and a confidence factor associated with each rule. It does not call for some statistical basis for supplying beliefs in events. Furthermore, it allows simultaneous rule representation and quantification of uncertainty that makes it a simpler and efficient approach in comparison to BN. However, the CF approach is also an ad hoc regime, which is not built on a solid theoretical foundation. It often results in many weaknesses in the reasoning mechanism (Negnevitsky, 2002). For instance, the CF approach works under the implicit assumption of independence among hypotheses, which is often an invalid postulation. Furthermore, the need for a large number of inputs tends to become a major preoccupation for the user (Ahmad *et al*, 2004e).

Dempster-Shafer Theory

The Dempster-Shafer (DS) theory of evidence addresses some of the weaknesses of the probabilistic approaches including the representation of ignorance, the unnecessary requirement that the sum of beliefs in an event and its negation be unity, etc. DS formalism has been applied to the quantitative modeling of preferences in situations with partially or even completely missing statistical data and to compute the impact of new observations on the resulting assessment. However, it does not specify how the probabilities are to be computed or results are to be interpreted. Furthermore, in certain instances, obviously incorrect conclusions can be reached (Negnevitsky, 2002). Moreover, the exponential nature of evidence and hypothesis spaces means application of DS is in the NP class. The only way around is to use heuristics to compute approximate solutions (Jameson, 1996). In short, DS is also an ad hoc approach and not suitable for incorporating explanation facilities.

Fuzzy Logic

Fuzzy Logic (FL) is a set of formal mathematical principles for knowledge representation involving degrees of membership of a given piece of information. It ventures to model the cognitive uncertainty and vagueness in human sense of words, opinions, decision-making and common sense tainted with imprecision, incompleteness, inconsistency, and vagueness (Cordon *et al.*, 2004; Tam *et al.*, 2002; Triantaphyllou & Lin, 1996). Indeed, people often reason in terms of vague and context dependent concepts in dealing with uncertain situations (Turban & Aronson, 2001). For instance, experts may describe preferences regarding the amount of white space in the layout in fuzzy terms as ‘small’, ‘medium’ or ‘large’.

The use of ‘degrees of membership’ and ‘partial matching’ techniques in FL offers formalism to model uncertainty in linguistic rules (Raoot & Rakshit, 1993). Indeed, FL has been shown to be an effective and robust technique in a variety of fields involving reasoning with incomplete, inconsistent, imprecise, and vague information (Negnevitsky, 2002). In addition, fuzzy logic focuses on the imprecision of the event itself; whereas techniques like CF and BN are concerned with the imprecision associated with the outcome of a well-defined event (Negnevitsky, 2002). Moreover, FL is superior to other uncertainty management tools in computational complexity. Consequently, we believe, FL has a significant role to play in cost-effective and robust LD preference modeling and reasoning under uncertain conditions. Further discussions on this promising technique follows in Section 2.8.

Automated Preference Discovery

Machine Learning (ML) techniques are capable of expressing a rich variety of non-linear decision surfaces (Zukerman & Albrecht, 2001). Observations of user behavior and history of interactions are treated as training examples by learning components. The knowledge acquisition is automatic and incremental (Conati & Zhao, 2004). Examples of ML include Artificial Neural Networks, Case-Based Reasoning, Memory-based Learning, Decision Tree Induction, Reinforcement Learning, Learning Automata, and hybrids (Billsus *et al.*, 2002; Jameson, 1996). Nevertheless, the knowledge-representation in ML is implicit and formats of learning results (probabilities, decision trees, etc.) are specific to the learning algorithm.

2.6.5 Comparison of Uncertainty Management Techniques

In order to have some kind of comparative evaluation of some of the aforementioned modeling techniques, we carried out an exploratory survey. In this survey, eight long time researchers and practitioners in LD field were asked to subjectively rank these uncertainty management techniques on a scale of 1 to 10 against various considerations such that higher scores represent rankings that are more favorable. Among those practitioners, three have expertise in facility LD, three have expertise in VLSI circuit LD (macrocell placement), and two have expertise in visual interface LD (interface design). The given techniques had varying *familiarity* rating among those evaluators. The ranking scores based on averages of experts’ evaluations as well familiarity with techniques are shown in Table 2-1.

Technique	Evaluation Criteria						
	Familiarity	Ease of Concept.	Ease of Use	Flexibility	Expressive Power	Robustness	Tractability
Deterministic	**9.6**	**9.9**	**8.3**	3.7	3.1	1.2	2.9
User-Controlled	**9.5**	8.6	6.3	3.9	3.8	3.6	2.0
REL-Chart	9.0	8.0	6.5	6.0	5.5	5.3	4.8
Bayesian	7.2	3.5	3.5	2.3	4.3	3.2	2.8
Certainty Factors	8.1	6.7	5.9	4.3	5.8	5.1	**6.5**
Dempster-Shafer	7.1	5.1	3.7	3.7	4.2	3.8	4.4
Fuzzy Logic	7.3	7.2	6.9	**7.3**	**7.5**	**7.8**	6.3
Simulations	5.8	2.9	1.7	1.4	6.9	2.2	1.8

Table 2-1: Subjective Ranking of Uncertainty Modeling Techniques in Layout Design

Among other evaluation criteria, the *Ease of Conceptualization* or understanding plays a key role in adoption of any methodology. Another important determinant of the success of a methodology is the *Flexibility*, which refers to the ease with which parameters and components of the system can be modified under dynamic conditions or during some sort of scenario analyses (Baldoni *et al.*, 2000). *Robustness* signifies the capability to perform reliably and effectively in changing situations and is another important determinant of the success of any technology employed in subjective, uncertain, and dynamic problem domains. *Expressive Power* implies the capability to represent a given scenario as accurately as possible. *Tractability* is an important dimension in modeling tools for complex domains such as the LD where combinatorial explosion could severely limit the efficiency and efficacy of the solution methodology.

Intuitively, deterministic and user-controlled techniques are rated as simplest to conceptualize and easiest to employ among all techniques. Nevertheless, these techniques received very low ratings for flexibility, expressive power, robustness and tractability of the approach, which are very important factors in determining the applicability and efficacy of such techniques. Indeed, higher ratings for ease of conceptualization and ease of use do not endorse these tools as valid option for large-scale problems where large volume of information is involved.

Similarly, computer simulation received the lowest ratings for all considerations reflecting the enormity of the task involved in simulation modeling and the difficulty in swiftly adapting to changing scenarios. Notably, FL received high ratings for these important determinants of efficiency and efficacy, implying its perceived promise in the LD research.

Despite significant limitations, this exploratory study provides grounds for comparing modeling techniques and selecting a more appropriate one. A well-thought out empirical study in this area would be an interesting research direction. A more critical classification and comparative review of effective modeling techniques for subjective and uncertain preferences specific to each uncertainty category, or combination of categories, delineated earlier would be a valuable extension of this work.

It is evident from the brief overview that some approximate reasoning mechanisms are more suitable for modeling subjective and uncertain preferences, constraints and fitness objectives in the LD. Indeed, a serious impediment in the extraction of knowledge from human experts is the imprecise, linguistic, or fuzzy manner of human conceptualization and articulation (Jackson, 1999). Experts think in vague and imprecise terms, for instance, 'very high' and 'low'; 'fast' and 'slow'; 'heavy' and 'light' etc. Furthermore, complex decision-making problems, such as LD, are full of uncertainties and ambiguities (Ahmad *et al.*, 2003, 2004b; Ayyub, 2001; Jameson, 1996; Tompkins *et al.*, 2002). Accordingly, the majority of LD guidelines and rules are essentially vague, ambiguous, and even conflicting in character.

Moreover, very little or no a priori knowledge is available in such dynamic domains. As such, the complexity and dynamics of the LD process make it impossible to gather meaningful statistical data, or even subjective probabilities, that could allow the use of some objective probabilistic approach. Furthermore, probabilistic approaches are concerned with the imprecision associated with the outcome of a well-defined event. For instance, there is 50% chance that a fair coin will come tails; however, when the coin is actually tossed, it comes down either 100% heads or 100% tails (Jackson, 1999).

In contrast, soft computing approaches like FL also focus on the uncertainty and imprecision inherent of the event itself (Negnevitsky, 2002). Such uncertainty is also referred to as *non-statistical uncertainty*. While *statistical uncertainty* may be resolved through observations, non-statistical uncertainty or fuzziness cannot be altered or resolved by observations (Engelbrecht, 2002). As mentioned, decisions in LD are based more on human intuition, creativity, common sense, and experience rather than the availability and precision of data. FL provides a very natural representation of human abstraction and partial matching by permitting incorporation of imprecision, incompleteness, and subjectivity in information into the model formulation, solution process, and analysis of alternatives. As a result, FL based modeling of predominantly subjective design guidelines applications has been proposed in various LD contexts (Ahmad *et al.*, 2003, 2004b; Badiru & Arif, 1996; Dweiri & Meier, 1996; Youssef *et al.*, 2003a). We see significant potential in using 'degrees of membership' and 'partial matching' techniques provided by the FL.

2.7 Automated Layout Design

Automating the LD process through computerized generation, evaluation, and treatment of superior layout alternatives by decision-makers is deemed a fundamental ingredient in any layout planning process (Ahmad *et al.*, 2004b, 2004c; Akoumianakis & Stephenidis 1997; Akoumianakis *et al.*, 2000; Francis & White 1992; Tompkins *et al.*, 2002). However, the majority of available computerized LD tools go little further than CAD-style drawing and documentation aids (Ahmad *et al.* 2004b; Ligget, 2000; Tompkins *et al.*, 2002). Conceivably, the high degree of complexity, subjectivity, uncertainty, and multiplicity of preferences and fitness objectives have deterred the study of the LD problem in an analytical manner (Ahmad *et al.* 2004b; Mir & Imam, 2001; Whyte & Wilhelm, 1999a).

Nevertheless, research in the LD area has resulted in several Automated Layout Design systems (henceforth, ALD). Such systems have their pros and cons. Consequently, we deem a critical overview of existing systems essential for any efforts directed at developing advanced ALD systems.

2.7.1 Existing Systems

Here we discuss some limitations of existing ALD systems in three most commonly encountered application domains, namely, Facilities Layout (e.g. Buildings), Circuit Layout (e.g. VLSI) and the Visual Interface Layout (e.g. the Web Page Layout). Despite all the assertions of flexibility and relative superiority, existing ALD systems have considerable limitations from various standpoints. Indeed, a majority of automated LD aids are computer-aided design (CAD) based documentation and drawing tools that typically do not afford automatic generation of superior layout alternatives or their improvement (Tompkins *et al.*, 2002; Whyte & Wilhelm, 1999b). Nevertheless, for reference purposes, here we mention some of the existing ALD systems in various application domains.

Facilities Layout Design

The development and application of automated Facility Layout Design (henceforth, FLD) systems started in the early 1960s. However, only a few such systems were made available in the market and still fewer earned some prominence (Tompkins *et al.*, 2002). A representative, nonetheless non-exhaustive, list of such FLD systems would include CRAFT (Buffa *et al.*, 1964), CORELAP (Lee & Moore, 1967), ALDEP (Seehof & Evans, 1967), SPACECRAFT (Johnson *et al.*, 1982), FLING (Blair & Miller, 1985), MOCRAFT (Svestka, 1990), BLOCPLAN (Donaghey & Pire 1990), LayOPT (Montreuil, 1991), CLASS (Jojodia *et al.*, 1992), MULTIPLE (Meller *et al.*, 1996), EDSLP (Osman *et al.*, 2003), FactoryOpt (Unigraphics, 2005), FACOPT (Balakrishnana *et al.*, 2003), FEMBUS, VIP-PlanOpt (EOS, 2005), etc. It should be noted that most existing FLD systems do not correspond to relatively more ambitious and pertinent research in FLD. In addition, only a handful of commercially available FLD systems incorporate layout generation, with the majority just furnishing some sort of layout evaluation coupled with computer-aided drawing (Meller & Gau, 1996). Apart from the few more recent FLD systems, an analysis of the capabilities, limitations, and other attributes of most available systems can be found in Sly (1995).

In existing FLD systems, various deterministic procedures are used to incorporate preferences regarding relative position of modules (Irani & Huang, 2000). These include Relationship Charts (REL) and Move Desirability Table (MDT) etc. (Francis & White, 1974). Such procedures typically handle preferences through some fixed inputs and essentially generate solutions alternatives by experimenting with those input values at different runs of the solution procedure (Tompkins *et al.*, 2002). Conceivably, such a complex and tedious process obliterates the very purpose of automation in this uncertain problem domain (Ahmad *et al.*, 2004c; Tompkins *et al.*, 2002).

Moreover, many existing systems do not treat a given module as an integral whole and often generate layouts with irregular and unrealistic module shapes entailing tedious manual adjustment and post-optimization processing. In addition, most existing systems tackle the relatively simpler one-to-one assignment problem in which all given modules are assumed to be of the same size and shape that are to be assigned to an equal number of fixed locations (Hassan, 1991). The inability of existing systems to consider a large number of modules for placement decisions is another important concern. Experience has shown that the VIP-PlanOpt™ is likely to be the fastest and the most robust among these available systems for handling FLD problems consisting of more than 40 modules (Ahmad *et al.*, 2004b).

Circuit Layout Design

Efforts in automated Circuit Layout Design systems (henceforth, CLD) gained prominence with the onset of such challenging, applications as VLSI. Due to rapid advances in complexity and miniaturization of VLSI chips, establishing a clearly dominant macro-cell placement strategy is not possible (Moon & Kim, 1998). Nevertheless, the importance of efficient and robust means of automatically generating superior alternatives cannot be overemphasized. Some examples of existing CLD systems are PROUD (Tsay *et al.*, 1988), GORDIAN (Kleinhans *et al.*, 1991), DOMINO (Doll *et al.* 1991), Cadence (CDS, 2004), Dragon (Yang *et al.*, 2002), Plato/Kraftwerk (Eisenmann & Johannes, 1998), Magic (Hamachi *et al.*, 1996). However, all existing CLD systems have good performance only under a rigidly specified set of conditions that cannot be attained in physical setting. An expert system, employing the automation capability of several good placement algorithms as well as flexibility of manual refinement, could afford the much needed efficiency and robustness in CLD systems.

Interface Design

Research in automating Interface Layout Design systems (henceforth, ILD) acquired eminence with the advent of such influential applications as Graphic User Interfaces (GUI) and the Web page layout in e-stores. These efforts could be surmised as a search for an adequate representation to encapsulate the required design knowledge and development of reasoning mechanisms to address interface related problems (Ahmad *et al.*, 2004e; McTear, 2000). The goal is to improve the effectiveness and efficiency of human–computer interactions through intelligent, adaptive, or multiple GUI.

Some examples of existing ILD systems are ADDI (El-Said *et al.*, 1997), UIDE (Foley *et al.*, 1991), LayLab (Graf 1997), Intelligent Agent based GUI (Agah & Tanie, 2000), GENITOR (Kameas & Pintelas, 1997), InterBook (Brusilovsky *et al.*, 1998c), KBS Hyperbook (Henze & Nejdl, 1999); IMAGEN (De Bo & Vervenne, 2003). However, such interface builders are either very primitive or they typically furnish a set of widgets to facilitate layout design of the visual interface and little or no support is afforded for designing the domain-specific layouts used by end-users. The process of mapping the domain objects and their properties into corresponding visual properties in the layout configuration is largely left to the user who face high cognitive and information overload that might result in inadequate layout configuration even when designers have ample graphics expertise (El-Said, *et al*. 1997). Some work has been done towards creating a visual knowledge-base to ameliorate this problem (El-Said, *et al*. 1997). Furthermore, efforts have been expended to automate graphics layouts within the context of electronic albuming (Giegel & Loui, 2001).

However, incorporation of subjective and uncertain preferences and properties is still an elusive objective. Nevertheless, ensuing efforts are growing fast and there is some research work directed at designing the *model-based interaction* (Brusilovsky & Eklund, 1998a; Foly *et al.,* 1991; Neches *et al.,* 1993), the *knowledge-based interaction* (Akoumianakis & Stephenidas, 1997; El-Said *et al.,* 1997; Keeble & Macredie, 2000; McTear, 2000), and the *agent-based interaction* (Agah & Tanie, 2000). It should be noted that the design of such static human-machine interfaces as process controllers are also an important category of the ILD problem. In general, the ultimate goal in all such human-machine interface design problems is to improve efficiency and efficacy of human-computer or human-machine interactions through intelligent, and possibly adaptive, user interfaces by tapping on knowledge of behavior and ergonomics of users as well as criticality and economics of operations.

2.7.2 Limitations of Existing Systems

Despite much advancement and innovation in ALD systems, the core issues of automated layout design have largely remained the same (Ligget, 2000; Tompkins *et al.*, 2002; Youssef *et al.*, 2003a). Although existing ALD systems have their pros and cons, in general, the existing systems are inflexible, myopic, slow, and incapable of tackling large-scale problems. The laggings become more evident when we observe that most LD applications are not well-defined. Typical examples of such applications are circuit layout design (Youssef *et al.*, 2003b) and Web page layout design (Ahmad *et al.*, 2002; McTear, 2000). Such ill-structured applications involve highly subjective, ill-defined as well as domain- or user-dependent issues. Here we discuss some limitations of existing ALD systems in more details.

Flexibility

Inflexibility is the hallmark of existing ALD systems that emanates from many sources. One such source is the rigidity and myopia of the fitness function(s) based on which the optimized layout alternatives are afforded (Ahmad *et al.* 2003, 2004b, 2004d; Azadivar, 1999; Irani & Huang, 2000; Singh & Wang, 1994; Tompkins *et al.*, 2002). Most existing systems deal with the LD problem as a single rigid criteria optimization problem. However, the superiority of a layout is often determined by a multitude of competing formal and informal criteria.

Furthermore, it has been argued, frequently and vehemently, that due to high subjectivity and strong NP-Hard character of structured mathematical formulations the procurement of such rigidly optimal layouts is not a compelling strategy (Ahmad *et al.*, 2004b; Tompkins *et al.*, 2002; Youssef *et al.*, 2003b). The rigid data and preference handling methods add to the inflexibility of existing ALD systems (Azadivar, 2000). The special format of data to enable storage and manipulation of layout descriptions using ‘crisp’ values often results in a large overhead in ‘digitizing’ the linguistic values and tackling scaling issues (scaling problems are further discussed in Section 5.3 and Section 5.4 within the context of multi-criteria decision-making). As such, existing ALD systems lack the flexibility to meet most needs of layout designers. Such inflexibility is, in part, responsible for the rather lethargic response from layout design practitioners in adopting ALD systems.

Creativity

The data handling methods in most existing ALD systems is suitable only when reliable deterministic or crisp interaction data is available and assignable to specific activities. Nonetheless, such data either does not exist or exists for some designated unknown and unrealistic modeling conditions. Thus, effective means of layout analysis and revision through the incorporation of subjective preferences as well as designers' intuition, creativity, and expertise are virtually nonexistent. Such disregard of less-formal and intuitive information results in an inability to capture many vital process dynamics (White & Taket, 1994). Moreover, the rationale of an ALD is to generate superior alternatives in a timely fashion as well as facilitate fast and easy means of manipulating those.

Primarily, it is because the existing techniques usually adopt an optimization approach, instead of a more pertinent decision-making paradigm (Abdinnour-Helm & Hadley, 2000; Ahmad, 2002; Ahmad *et al.*, 2004b; Badiru & Arif, 2000; Azadivar, 2000; Foulds, 1997; Osman *et al.*, 2003; Tam *et al.*, 2002; Zhang *et al.*, 2002). Consequently, such indifference to cognition, intuition, and vision of decision-makers is critical from practical perspective and severely limits the applicability of these techniques (Kintsch, 1998; Tompkins *et al.*, 2002).

Productivity

Productivity of the existing ALD systems, as measured by speed of execution in generating and manipulating superior layout alternatives, is another issue of concern. Indeed, the speed of existing ALD systems is among the bigger obstacles in the adoption of these systems (Ligget, 2000). Many dynamic and intricate applications cannot wait forever to acquire superior layout alternatives. Our personal experience has demonstrated that VIP-PlanOpt™ (EOS, 2004) is superior to many existing ALD systems in terms of speed and capability to handle large-scale problems. However, even VIP-PlanOpt seems to be severely constrained in terms of flexibility and applicability under most practical situations. Consequently, there is a considerable need for developing efficient and robust placement solutions. In this direction, employing a combination of multiple placement algorithms is also considered worthwhile, an approach we followed in building our ILG (Adya *et al.*, 2003; Yang *et al.*, 2002).

Scalability

The applicability of existing ALD systems is further constrained by an inability to consider more than few scores of modules in layout decisions (Ligget, 2000). Such ALD systems often cannot handle most real world problems. Incidentally, a problem of 40 or more modules has not been presented in the ALD literature until very recently (Ahmad *et al.*, 2004d, 2004f; Hopper & Turton, 2001; EOS, 2004). Once again, we found VIP-PlanOpt superior to other existing ALD systems in handling large-scale problems, keeping in view that it could handle problems consisting of more than 500 modules.

Generalizability

In addition, most systems solve the problem using relatively simpler linear or quadratic assignment models. In such one-to-one assignment models, it is assumed that all modules are of equal size and there are exact-fit locations available in the packing space to which these modules are to be assigned. Despite existence of some applications fulfilling such conditions, these are largely unwarranted assumptions. Furthermore, the use of single and rigidly defined fitness measure do not model the realistic LD scenarios (Irani & Huang, 2000; Ligget, 2000).

Diversity

In general, layout alternatives are generated by following a set of largely predefined steps and the optimization process works with various untenable simplifying assumptions (Ahmad *et al.*, 2004c; Irani & Huang, 2000; Ligget, 2000). Obviously, such unwarranted assumptions cannot be eliminated by the users from the underlying process. Consequently, such procedures do not search the solution space in appropriate and diversified manner. As such, the limitations are imposed on much desired comprehensiveness required for obtaining a truly superior solution.

Portability and Reusability

In general, the existing ALD systems are designed for the use on some specific platform, employing some rigid data representation schemes that cannot be readily used by other applications. User interfaces seem to lack many usability and effective human-computer interaction aspects.

Learnability

Inability to learn from the experience or user behavior is another source of ineffectiveness and does not bode well for the future of ALD systems. Indeed, some renowned layout design practitioners have been recommending that people design and code their own layout optimization aids, as each LD situation is unique in itself (Irani, 1992, 2003). Paradoxically, the inadequacy in assimilating subjective design preferences in ALD systems has lead researchers to recommend large input requirements to obtain both a better control on the LD process as well as the quality of outcome (Levary & Kalchik, 1985; Tompkins *et al.*, 2002). However, such approaches are inherently inefficient and irrational for most applications (Ahmad *et al.*, 2004b; Zhang *et al.*, 2002).

2.8 Promising Soft Computing Tools

This section provides an overview of the soft computing technologies that are considered valuable for providing intelligent decision support in the LD. The Soft Computing paradigm characterizes one of the most recent fields in the area of computational intelligence that could deal effectively with complex structuring and ill-defined dynamics of the LD problem. Intelligent systems are designed to solve problems that usually do not have clearly defined good or bad solutions (Cordon *et al.*, 2004; Karray & de Silva, 2004; Negnevitsky, 2002).

The Soft Computing paradigm is concerned with modes of computing in which precision is traded for facilitating tractability, efficiency, flexibility, efficacy, robustness, implementation, and user-acceptance. The common denominator in soft computing techniques is their digression from classical reasoning and modeling approaches (Baron *et al.*, 2001; Cordon *et al.*, 2004; Zha, 2003). The principal components of soft computing are fuzzy logic, artificial neural networks, genetic algorithms, probabilistic reasoning, approximate reasoning, decision support systems, expert systems, chaos theory, etc.

Such techniques could afford means for imitating the impressive human capability of expressing knowledge through linguistic representation of information. Indeed, such approaches are gaining acceptance for modeling cognition, intelligent systems, and artificial intelligence because the procedures involved are most analogous to human reasoning. Here we identify the key strengths, merits, and the synergy of some of the relevant tools, which are promising choices for tackling the problem at hand. It should be noted that some of the soft computing approaches such as Certainty Factors (CF), Dempster-Shafer (DS), and Bayesian Networks (BN) have already been discussed in previous sections, within the context of uncertainty management

2.8.1 Genetic Algorithms (GA)

Genetic Algorithms (GA) are motivated by biological reproduction process imitating the natural selection and biological evolution. GA combine the idea of ‘the survival of the fittest’, random but still structured search, and parallel evaluation of nodes in the search space (Karray & DeSilva, 2004; Holland, 1975). A typical GA evolution cycle consists of a generation of solutions out of which some parents are selected for genetic evolution. The genetic evolution results in a new generation of solutions and the cycle repeats.

The encoding of variables, simplicity and ease of operations, minimal computational requirements, random initial population, probabilistic search rules, multiple search points, suitability for parallel processing, independent control of exploitation and exploration aspects, robustness, and global perspective of GA have made them applicable to a wide variety of domains. Several of these advantages are, explicitly or implicitly, derived from the population-based search where GA determine next search points using fitness values of current search points, which are spread throughout the search space. GA have demonstrated their power for solving difficult problems making them a good choice for decision support in the LD due to the ability to provide a population or a set of superior alternatives.

There is a substantial body of literature available on application of GA in various LD work-domains in different ways (Ahmad *et al.*, 2004d; Chan & Tansri, 1994; Geigel & Loui, 2001; Mak *et al.,* 1998; Mazumder & Rudnick, 1999; Norman & Smith, 2002; Tate & Smith, 1995; Youssef *et al.*, 2003a, 2003b). Notably, GA have been shown to supersede other search methods in solving LD problems in terms of speed and efficacy (Hopper & Turton, 2001). Detailed surveys of research conducted in this direction can be found in the literature (Kado, 1995). Further details of GA in LD are discussed in Chapter 4 within the context of ILG.

2.8.2 Fuzzy Logic (FL)

The concept of Fuzzy Logic (FL) was pioneered by Lotfi Zadeh (1965) as a system of logic for representing conditions that could not be easily denoted by crisp values like ‘true’ or ‘false’ in Boolean and conventional logic. The role model for FL is the human mind in which a proposition is neither *True* nor *False*, but may be true or false to some degree. FL provides a means to model these continuums of values through fuzzy sets. Details of fuzzy logic and fuzzy set theory can be found in the literature (Karray & De Silva, 2004). However, here we discuss some salient and valuable features of FL from LD perspective.

Indeed, FL has been successfully employed for representation of, and reasoning with, the knowledge in expert systems (Guiffrida & Nagi, 1998; Negnevitsky, 2002). It furnishes a very natural representation of human conceptualization and partial matching, which inherently relies on common sense as well as vague and ambiguous terms (Nyongesa *et al.*, 2003). Consequently, apart from partial matching, another aspect of subjectivity and uncertainty in preferences is their linguistic and imprecise descriptions. For instance, experts can describe preferences regarding the amount of white space in the layout in fuzzily through such adjectives as ‘small’, ‘medium’ or ‘large’. In order to model the imprecision of such linguistic terms, Zadeh (1965) advocated the notion of a *linguistic variable* defined as a variable whose values are words or sentences in some natural or artificial language. The collection of all probable values of a linguistic variable defines the *universe of discourse*. A fuzzy set of a universe of discourse is defined by a function, commonly termed as the *Membership Function* (MF), that maps elements of a fuzzy set into a real value in an interval between 0 and 1.

Interestingly, FL furnishes the ability to separate the computational logic from the fuzziness in data and rules (Kelly, 1997). In conventional binary logic, rules need to be updated once either logic or fuzziness in data is changed. However, FL revises fuzzy rules when the logic needs to be changed and adapts membership functions that characterize the fuzziness when fuzziness should be changed (Negnevitsky, 2002). Moreover, FL incorporates the notion of *hedges* or terms that modify the shape of fuzzy sets. These linguistic hedges include such adverbs as very, significantly, somewhat, quite, more, less, slightly, etc. In general, human speak about linguistic hedges with *narrowing effect* (e.g. very, significantly, etc.) and with *widening effect* (e.g. more, less, slightly, etc.). These hedges act as operators and create new fuzzy sets from given fuzzy sets such as very small, slightly heavy, etc. (Dvorak & Novak, 2004). It means that given linguistic/fuzzy rules may be augmented without even changing the rule, but by only applying a hedge operation on given rules. It furnishes a powerful and robust tool in reducing the size of the rule-base required for any fuzzy inferencing system. Consequently, FL has the potential to significantly reduce not only the knowledge acquisition cost but also the computational cost (Cintula & Navara, 2004; Rommelfagner & Slowinski, 1998). By virtue of fuzzy sets and hedges, FL often renders more than 90 per cent reduction in the number of rules (Negnevitsky, 2002).

It should be noted that FL has been extensively applied in operations research. Furthermore, a large body of literature exists on fuzzy multi-criteria decision-making. The success of FL in a variety of subjective and uncertain domains also inspired efforts in employing fuzzy inferencing mechanisms in various LD work domains (Badiru & Arif, 1996; Dweiri & Meier, 1996; Grobelny, 1987a; Grobelny, 1987b; Karray *et al.* 2000a, 2000b; Kang *et al.*, 1994; Raoot & Rakshit, 1993; Raoot & Rakshit, 1991).

Applications modes of FL in Layout Design

Incidentally, FL can be utilized in LD in various forms (Cordon *et al.*, 2004;). For instance, it can be used as a *Linguistic Tool* to model problems comprising of fuzzy phenomena or relationships and to acquire/represent the domain-specific knowledge. In addition, FL can be employed as an *Analytical Tool* to advance insights into the problem through analysis of Fuzzy Decision Making (FDM) models. Furthermore, the use of FL as an *Algorithmic Tool* could make solution methods faster, robust, and stable. Notably, all these application modes of FL are inherently pertinent to the research in the layout optimization. Here, a brief overview of these application modes of FL in LD is provided.

The most popular application mode of FL in LD is as a linguistic tool (Dvorak & Novak, 2004). In such cases, FL is used to model linguistic patterns or preferences mainly in solving the FLD problem (Ahmad *et al.*, 2004b; Blair & Miller, 1985; Evans *et al.*, 1987; Kim *et al.*, 2001; Nyongesa *et al.*, 2003; Tompkins *et al.*, 2002). For instance, the subjective, uncertain, or linguistic preferences corresponding to relationships like 'importance' and 'closeness' of modules can effectively be modeled using FL.

In addition, FL has been used as an analytical tool where layout fitness metrics are modeled as a multi-criteria decision making (MCDM) problem (Khan & Sait, 2002; Nyongesa *et al.*, 2003; Soltani & Fernando, 2004; Youssef *et al.*, 2003). Such approaches essentially form a hybrid layout fitness metric using an amalgamation of both quantitative and qualitative criteria.

Moreover, FL has also been used in LD as an algorithmic tool where placement decisions and the spatial relationships are determined by fuzzy rules (Evans *et al.*, 1987; Youssef *et al.*, 2003). In such cases, the solution algorithm utilizes various linguistic variables for expressing qualitative and quantitative characteristics affecting placement decisions.

The efficacy of such procedures is demonstrated in the literature. Consequently, there has been growing interest in the use of FL in the LD. Nevertheless, an encompassing application of FL covering all aforementioned notions is largely missing. Furthermore, the important issue of efficacy and speed of these procedures for larger problems has not been adequately addressed. In Chapter 5, we provide the design and implementation of a fuzzy Preference Inferencing Agent (PIA) and details of some relevant issues.

2.8.3 Artificial Neural Networks (ANN)

A traditional knowledge-based system cannot learn and improve through experience. However, an automated learning mechanism could improve the speed and quality of knowledge acquisition. The ability of Artificial Neural Networks (ANN) to learn from historical cases could generate rules automatically, thus eluding tedious and expensive processes of knowledge acquisition, validation and revision. ANN represent a class of powerful and general-purpose tools that have shown enormous promise in a wide array of applications (Nauck *et al.* 1997; Zha & Lim, 2003). This information-processing and discovery paradigm is inspired by the structure and function of the human brain and consists of a number of simple and highly interconnected processors termed as *neurons*. These neurons are connected by weighted links that transmit signals from one neuron to another and ANN learn through repeated adjustments of weights. These weights implicitly store the knowledge required to solve specific problems (Nauck *et al.*, 1997).

As stated, the LD rules and preferences are quite dynamic and evolutionary in nature as people learn new concepts and outgrow old ideas. This dynamic nature of preferences often results from decision-makers' interaction with existing or intermediate layout solutions, a phenomenon often referred to as *dynamic rationality* (Billot, 1998; Bouyssou & Vincke, 1998). It suggests that some online ANN based Pattern Discovery and Validation Agent would offer value by providing patterns of LD rules and preferences in an automated and self-updated manner (Ahmad *et al.*, 2003, 2004b; Chung, 1997; Turban & Aronson, 2001; Tsuchiya *et al.*, 1996). Further issues related to automatic preference discovery are dealt with in details in Chapter 5 within the context of a Preference Discovery Agent (PDA).

2.8.4 Reinforcement Learning (RL)

Reinforcement Learning (RL) is another machine learning paradigm that attempts to learn from environment and adapt the system accordingly. RL could assist in automatic refinement and enhancement of knowledge or preferences by continuously monitoring the layout designers' interaction with, or ranking of, the available layout alternatives. Consequently, RL could be very useful in modestly evolving scenarios by reducing the need for the tedious revision of knowledge-base. However, little work can be found regarding the use of RL in LD problems. In Chapter 5, we discuss some issues related to automated preference revision in details. However, we have left incorporation of RL into our system as future work.

2.8.5 Knowledge-based Systems (KBS)

Various knowledge-based systems, such as decision support systems (DSS) and expert systems (ES), have been successfully deployed in a variety of work domains involving subjective and uncertain information (Ayyub, 2001; Hall & Kandel, 1992; Negnevitsky, 2002; Turban & Aronson, 2001). Indeed, knowledge is deemed as the only factor of production that is not subjected to diminishing returns (Hirji, 2001). Although descriptions of various DSS or ES for LD are reportedly available in literature, we believe that the real potential of such powerful paradigms is largely untapped. The existing knowledge-based systems in LD use very restricted definition of decision support or expert systems. Nevertheless, we deem these paradigms very valuable in the context of the LD problem. As such, in the next section, we introduce the capabilities and caveats pertinent to DSS and ES approaches. Moreover, a critical review of existing DSS and ES for LD is provided in Section 2.9.3.

2.9 Knowledge Based Layout Design (KBLD)

Complex decision making problems often require considering enormous amount of incomplete, imprecise, inconsistent, and vague information distributed across many variables (Jackson, 1999). Such is often a case with the LD problem and alleviating even some of the limitations in the existing ALD systems could prove a daunting task. However, such barriers should not deter researchers from addressing the problem, albeit in an incomplete sense (Bazaraa, 1973; Silver, 2004). In this regard, such knowledge-based approaches as DSS and ES appear quite promising. A solution approach that synergistically combines the benefits of knowledge-based systems technology with the optimization power of metaheuristics as well as powerful preference acquisition and modeling tools of soft computing should offer substantial benefits.

Although several existing ALD systems boast to be knowledge-based in one way or other, such claims are made in a very narrowly defined perspective of knowledge-based systems. Such systems tend to focus on integrated problem solving experience instead of the actual LD process (Ligget, 2000). As such, we deem it appropriate to include an overview of the two most popular knowledge-based problem solving approaches, namely DSS and ES, and a disjunctive survey of capabilities/limitations of the existing knowledge-based LD systems.

2.9.1 Decision Support Systems (DSS)

A decision strategy that is completely informed, perfectly logical, and oriented towards economic reward is referred to as *perfect rationality* (Simon, 1955). However, in practice, decision makers often resort to *bounded-rationality* reflecting on inadequacy of tangible and intangible resources (Greenberg *et al.*, 2000; Simon, 1957*a*, 1957*b*). Decision makers often face various impediments in solving LD problems. The most taxing issue is the *ill-structured* nature of problem that tends to be complex, relatively novel, subjective, and uncertain and require a high degree of creativity and expertise (Ignizio & Cavalier, 1993; Ignizio, 1991).

In addition, decision-makers often resort to making presumptions regarding some aspects of the available information, an affinity referred to as *framing* (Greenberg *et al.*, 2000). Such cognitive biases for procuring and dealing with information in an error-prone fashion may emerge in various ways. For instance, there is an instinctive human tendency to rely on more recent or readily available information, a phenomena commonly referred to as the *availability heuristic* (Johns, 1996). In addition, there is often a propensity to expose only the information that conforms to one's own analysis of the situation, referred to as the *confirmation bias* (Greenberg *et al.*, 2000). Such biases are also an outcome of the *information overload* where more information is acquired or available than is necessary to make effective decisions (George & Jones, 1996).

The Search of an objective measure for evaluation and comparison of alternatives adds to other temporal, computational, sociological, psychological, and resource constraints that hamper effective decision-making (Simon, 1955, 1957b). Consequently, decision makers often resort to *satisficing* approaches instead of rational optimizing methodologies (March & Simon, 1958). Satisficing refers to the process of formulating an adequate degree of acceptability for a solution to a problem and screening solutions until one that surpasses this benchmark is found (Bower & Zi-Lei, 1992).

Decision Support Systems (DSS) represent a class of computerized information systems that utilize the knowledge about a specific application domain to assist decision makers by recommending appropriate actions and strategies (Turban & Aronson, 2001). A typical DSS consists of a Database, a Model Base, a Communications Component, and a User Interface (Power, 2001). However, the term DSS may encompass a wide array of systems, tools, and technologies with the aim of achieving efficacious decision-making (Marakas, 2002).

The DSS problem-solving paradigm provides a means for assisting decision makers in retrieving, summarizing, and analyzing decision relevant data. Consequently, it results in a reduction in the cognitive overload faced by the decision maker(s). Research has shown that DSS techniques are useful in generating and evaluating a large number of alternative solutions and effectively helping decision-makers in arriving at better decisions (Turban & Aronson, 2001; Greenberg *et al.*, 2000). However, it should be emphasized that a DSS neither automates the decision process nor imposes solutions. It simply provides analytical and information processing support in an interactive environment.

Incidentally, the layout design is not an exact science. Indeed, it is irrational to expect that a specific layout would surpass all others for every evaluation objective (Tompkins *et al.*, 2002). Consequently, the generation of superior layout alternatives in a flexible and automated manner is critical to any layout planning process (Tompkins *et al.*, 2002). Conceivably, some DSS mechanism could be beneficial in solving the LD problem. As such, some research can found in the literature that attempts to solve the problem through the DSS paradigm. Here we describe a couple of such systems reported in the literature.

LayoutManager

Foulds (1993a; 1997) describes a system called LayoutManager that is reportedly deemed a decision support system in facilities planning. It employs various graph-theoretic algorithms from the existing literature (Foulds, 1991, 1993b; Hassan & Hogg, 1987). LayoutManager contains a menu-driven interface that permits users to select the layout design algorithm and other necessary starting conditions. Furthermore, it permits users to select one of several layout fitness evaluation metrics. The problem specific data must be provided in a standard format through a text file. Any modifications to the design parameters require direct editing of this text file. In order to generate a layout alternative, user selects a starting module, a graph search heuristic, and a fitness metric.

Further alternatives may be generated through successive trials in which the starting module, graph heuristic, or fitness metric is changed.

Proximity stipulations are provided in the form a of REL chart. Some provisions are available that allow users to assign values to various REL parameters. However, it does not provide any means for resolving conflicts in preferences. Furthermore, no means are available to restrict the combinatorial explosion of REL charts, which are required for every pair of modules. It is stated that for some heuristics this number reaches $(n-2)^3+1$, where n is the number of modules considered (Foulds, 1997).

The author mentions the possibility of employing stochastic search strategies like simulated annealing and tabu search as promising. However, the system described in Foulds (1993a) and Foulds (1997) does not contain any stochastic search capability. Layout solutions are created by using one of the deterministic graph search heuristics available with the system. Such algorithms, beside other laggings, do not allow diversified and extensive search of the solution space. Moreover, the tree search performed by the heuristics is also restricted to only three children per parent node in order to speed up the processing.

The notion of 'gradualism' is mentioned as a key idea in layout design, referring to the gradual progress towards the final solution through a series of manipulations and transformations of an initial layout alternative. However, the LayoutManager does not provide any means for giving users any real control over the proceedings. In addition, the shape or even the location of modules cannot be controlled. In short, it does not provide functionalities that would allow users to interactively make any informed or knowledge-based interventions or even manipulations of the layout alternatives produced by the system.

Furthermore, the LayoutManager provides users the ability to select one of the several rigid and myopic layout fitness metrics. As such, those metrics cannot be utilized in some hybrid multi-criteria decision-making (MCDM. User cannot make any informed amendments, hybridizations, or augmentations to the available fitness metrics. In addition, details of the system and its usage are very complex and difficult to follow. In short, the system lacks the flexibility, efficiency, efficacy, scalability, and robustness that would be logical requisites for a DSS in LD.

NSF-DSS

Tam *et al.* (2002) describe a nonstructural fuzzy decision support system (NSF-DSS) that integrates both experts' judgment and computer decision modeling, making it suitable for the appraisal of complicated construction problems. The system allows assessments based on pairwise comparisons of alternatives. However, this pairwise comparison approach is inherently inefficient and requires frequent and expensive backtracking. Nevertheless, the research by Tam *et al.* (2002) provides many useful insights and future research directions in this field.

2.9.2 Expert Systems (ES)

An Expert System (ES) is defined as an intelligent computer program that applies reasoning methodologies or the knowledge in a specific domain to render advice or recommendations – much like a human expert (Turban & Aronson, 2001). ES are usually characterized by the existence of a large repository of knowledge for solving problems in a very constricted work domain (Malakooti & Tsurushima, 1989; Turban, 1995). Such a knowledge repository may comprise of human knowledge and expertise formulated as specific rules and heuristics (Jackson, 1999; Turban & Aronson, 2001). An ES aids decision-making by overcoming the otherwise undesirable and unavoidable situation where there is too much to learn, too much to know, and too little time and resources to employ.

The distinguishing feature between ES and DSS is the separation of knowledge and the reasoning method involved in an ES. Such a separation of domain knowledge and inferencing mechanism results in greater modularity in the system (Negnevitsky, 2002). As such, it affords a greater degree of flexibility, thus making it the paradigm of choice for our research in automating the LD process. Furthermore, ES provide explanation capability as a mean of understanding the reasoning mechanism involved in arriving at some decision.

A traditional ES is shown in Figure 2-4. It has five basic components, namely a Knowledge Acquisition Module, a Knowledge Base, an Inference Engine, an Explanation Facility, and an interactive User Interface (Negnevitsky, 2002; Nikolopoulos, 1997). The details about individual components and their synergy follow in Chapter 3 within the context of the proposed intelligent system for decision support and expert analysis in layout design.

Indeed, an ES designed specifically to aid decision makers continuously increases productivity, lowers costs, and spurs innovation (Konar, 2001; Marakas, 2002). However, existing literature on the application of the ES paradigm in LD is quite meager. In addition, such systems have considerable shortcomings and do not engender most benefits that are usually attributed to ES. Here we provide an overview of such existing systems.

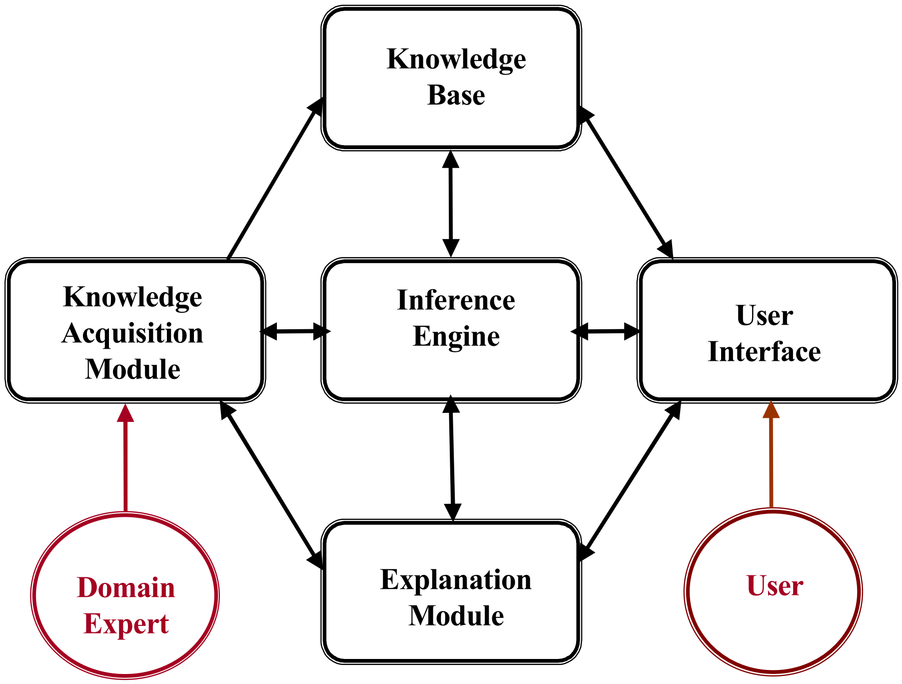


Figure 2-4: A Typical Expert System.

FADES

Fisher & Nof (1984) present a FAcilities Design Expert System (FADES) for machine LD applications. The reported prototype contains various FLD heuristics and an inferencing mechanism to select a heuristic appropriate for the given scenario. The inference engine is developed using a PROLOG interpreter with a forward-chaining depth-first search. Knowledge is represented using first-order predicate logic. However, FADES can only solve small-scale problems consisting of equal size modules. Furthermore, it cannot handle conflicting preferences. Moreover, the prohibitive computational cost means that the algorithms used in FADES are not very efficient. Above all, it does not engender a diverse set of layout alternatives, a key requisite in generation of LD decision alternatives.

IFLAPS

Kumara *et al.* (1985, 1986, 1988) present a machine layout design ES (IFLAPS) that deals with the one-to-one assignment type scenarios. It employs a few simple rules of thumb consisting of deterministic and pre-defined steps. It contains a module consisting of production rules that determine the facility to be assigned first in the block plan. However, simple deterministic rules of thumb mean that it neither affords any actual optimization nor furnishes any diversity in alternatives. Furthermore, IFLAPS requires a significantly high degree of user inputs and interventions. Moreover, it does not provide functionalities to modify or refine the alternative generated by the system. In addition, IFLAPS employs arbitrarily chosen priority indices or user intervention for resolving conflicts in preferences. Moreover, IFLAPS is not suited for problems with more than a score of modules.

ES-MCFL

Malakooti & Tsurushima (1989) report an ES for multiple-criteria FLD (ES-MCFL) that employs a forward chaining reasoning mechanism. Authors argue that despite the quantitative nature of MCDM, the ability to handle multiple conflicting goals might resemble experts' cognitive treatment of subjective and uncertain preferences. However, ES-MCFL considers only one criterion at a time based on priority rules and does not impart the requisite flexibility and robustness to the system. Furthermore, it uses mostly crisp data, crisp logic, and deterministic heuristics. In order to generate alternatives, users are required to change the priorities and repeat the procedure. Consequently, the layout alternatives do not offer much diversity. In addition, the user interface is not designed to permit decision-makers to manipulate and refine a given alternative. Moreover, the system cannot efficiently handle even modestly large problems.

KBML

Heragu (1990) presents a Knowledge-based Machine Layout (KBML) system that tackles one-to-one assignment type scenario. It is claimed to be capable of solving relatively larger problems in comparison to other KBLD systems existing at that time. It employs both quantitative and qualitative data. However, the crisp nature of data means it cannot adequately capture subjective and uncertain dynamics of the problem domain. Furthermore, conflicting preferences require user intervention. KBML employs various models and algorithms, each of which is suitable to some specific scenario, with a hope that a collection of models would cover most of the scenarios. KBML requires manual modification in parameters to generate new feasible solutions and may require several uninformed iterations before producing a workable solution. Furthermore, the deterministic nature of algorithms does not afford an adequate level of optimization and diversity in alternatives. In addition, the computational cost of procuring a viable alternative is still quite prohibitive.

SightPlan

SightPlan is an ES that generates layouts for temporary facilities on construction sites (Tommelein, 1989, 1997). However, it does not provide ways to incorporate soft constraints and preferences. Furthermore, it cannot handle conflicting preferences and requires user to rectify conflicts manually pre-defined steps. The layout solutions do not have any diversity, a key requirement in providing design support to LD experts.

2.9.3 Limitations of Existing Knowledge-based Systems

Most existing Knowledge-based Layout Design (KBLD) systems are not very robust and flexible, as users might want or as state of affairs might necessitate. Such lack of robustness and flexibility are a result of various factors. Here we describe some of the more salient factors.

Scope

In general, a relatively simpler version of the one-to-one assignment type LD scenario is tackled. Such problem formulations have some important applications in various work domains like machine or job shop LD. However, such formulations do not suffice for most LD domains. Consequently, the existing systems do not seem to be effective even in modestly subjective and complex situations.

Scalability

Existing KBLD systems may handle only small-scale problems reasonably fast. However, even for modestly large problems, these systems entail prohibitively large computational time. More general LD scenarios require solutions for large-scale continuous space layout problems consisting of unequal size modules with relatively little computational efforts.

Diversity of Alternatives

In general, heuristics employed for obtaining layout solutions are deterministic in nature. In some KBLD systems, it may involve adding a few production rules to guide the optimization search process. Consequently, despite some claims, these KBLD systems do not present a diverse set of superior layout alternatives. Nevertheless, the diversity in alternatives is a key ingredient in providing decision support in such complex problem domains.

Quality of Alternatives

In addition to diversity, the superiority of solution alternatives lies at the core of any knowledge-based solution methodology in layout decision analysis and design. However, the deterministic nature of LD algorithms and the lack of diversity in decision alternatives mean that the existing systems require many reruns before obtaining a satisficing alternative. The primary reason is the difficulty in modeling sub-cognitive and implicit preferences, which includes difficulty in quantifying the qualitative determinants of layout fitness.

Transparency

Existing KBLD systems offer very restricted, if any, explanation facilities. Towards this end, simply providing the sequence of the rules employed in reaching a decision is considered sufficient. Relating the accumulated heuristic knowledge to deeper understanding of the problem domain is still an elusive objective.

Learnability and Reusability

It should be noted that developing an ES for such a complex problem as LD might take efforts equivalent to several scores of person-years (Turban & Aronson, 2001). Conceivably, such gigantic and concerted efforts are hard to justify if most system improvements and adaptations call for significant and time-consuming additional labor from its developers (Negnevitsky, 2002). Consequently, there is a pressing need for developing ES that learn and update knowledge in an automated manner. Most existing KBLD systems offer little or no ability to learn from their experience and observation of user behavior.

Interactivity

The interactivity in KBLD systems would enable swift change of rules, parameters, algorithms, priorities etc. (Ligget, 2000). However, most existing KBLD systems lack user interface that could afford effective and interactive analysis and design. Apparently, most interfaces were designed by the LD practitioners themselves. Consequently, these interfaces lag considerably in interactivity, usability, and suitability to the ecology of the work domain.

2.10 Summary

In this chapter, we provided a review of concepts and the literature in domains relevant to the goal of enabling intelligent decision support in LD. We have identified the key problems associated with the task and existing automated layout design systems. In addition, we have described some promising tools for achieving the desired objective. Moreover, we identified key strengths, merits, and synergy of such tools that render them promising choices. In the subsequent chapter, we provide a new research paradigm for an Intelligent System for Decision Support and Expert Analysis in the Layout Design (IDEAL) as well as its philosophy.

Chapter 3

RESEARCH FRAMEWORK

3.1 Introduction

In this chapter, a research framework for an **I**ntelligent System for **D**ecision Support and **E**xpert **A**nalysis in **L**ayout Design (**IDEAL**) is presented. The research framework is aimed at addressing some of the major issues involved in using the sub-cognitive, subjective, and fuzzy design knowledge/preferences as a key to enhancing productivity of layout designers.

Layout design is a tedious process that entails sophisticated decision analysis and design support. Multiplicity, subjectivity, uncertainty, and evolving nature of layout design preferences and objectives mean that the synergistic use of available modeling and design tools as well as an expertise in tradeoffs lies at the heart of any layout design and analysis process. Consequently, any good automated layout design system should be flexible and robust enough to facilitate adaptation to the evolving scenarios as well as incorporation of cognitive and sub-cognitive expertise of domain experts. However, most traditional approaches to the layout design problem lack the requisite flexibility, efficacy, and robustness (Abdinnour-Helm & Hadley, 2000; Ahmad *et al.,* 2004b, 2004c; Badiru & Arif, 2000; Osman, 2003). Furthermore, layout designers encounter a high cognitive overhead in acquiring, remembering, understanding, and applying the vast body of subjective and uncertain information/preferences available to them.

Recent developments in the field of intelligent systems design have rendered powerful alternatives for tackling with such complex and uncertain problems as the layout design. Such soft computing tools include an array of emerging computing disciplines such as Fuzzy Logic, Neural Networks, Genetic Algorithms, and hybrids like neuro-fuzzy-genetic systems (Ahmad *et al.*, 2004b; Karray & de Silva, 2004; Zha, 2003). All these technologies share the common denominator in their digression from classical reasoning and modeling approaches through a set of more flexible computing technologies (Negnevitsky, 2002). Such technologies have demonstrated the power and philosophy to solve complex and ill-defined nonlinear problems and offer significant potential in dealing with layout design problems. Such approaches are gaining favor in modeling cognition, intelligent systems, and artificial intelligence as procedures involved are most analogous to human reasoning (Ahmad *et al.*, 2003, 2004c; Akoumianakis 2000; Zadeh, 1999). IDEAL seeks synergistic employment of some of these powerful tools, which is described in detail here.

The rest of the chapter is organized as follows. Section 3.2 presents the research framework. Section 3.3 describes rationale and role of various components in IDEAL. Section 3.4 outlines the philosophy and synergy of various intelligent components in IDEAL. Section 3.5 provides data requirements of IDEAL. Section 3.6 summarizes the chapter.

3.2 Research Framework

It should be noted that the computer-based algorithms could not replace human judgment and experience as these algorithms do not always capture the qualitative and intelligence aspects of layout design (Tompkins *et al.,* 2002; White & Taket, 1994). Nevertheless, it is often effortless for experts to visually inspect some layout alternative and endorse its acceptability or otherwise. Conceivably, there are strong prospects for devising some incomplete models and soft methods to supplement human erudition and intuition. For instance, computerized generations of alternate layouts could provide efficacious support to the layout analyst by assisting in aptly addressing some of the complex problem dynamics.

Indeed, the possibility of significantly enhancing the productivity of layout analyst and the quality of final solution through automated and expedited production, analysis, and treatment of a large number of superior layout alternatives has been advocated and sought since long (Armour & Buffa, 1963; Bazaraa, 1975; Tompkins *et al.*, 2002). In this regard, various popular solution approaches have their strengths and weaknesses. The usual tradeoff involved between the flexibility in incorporating the problem-specific details and the exhaustiveness of the search involved with various LD optimization tools is depicted in Figure 3-1 (Barakat *et al.*, 1995; Chung, 1999).

It can be seen from Figure 3-1 that on one end of the spectrum are enumerative search techniques, which are superior in terms of exhaustiveness in exploration of solution space. However, such general techniques incorporate very meager level of details from the problem-specific information. Furthermore, the application of such techniques is marred by the process speed and computational complexity. On the other end of the spectrum, human designers command high level of flexibility and are capable of incorporating high level of details of problem-specific information into the design process. However, the cognitive and information processing limitations of human designers translate

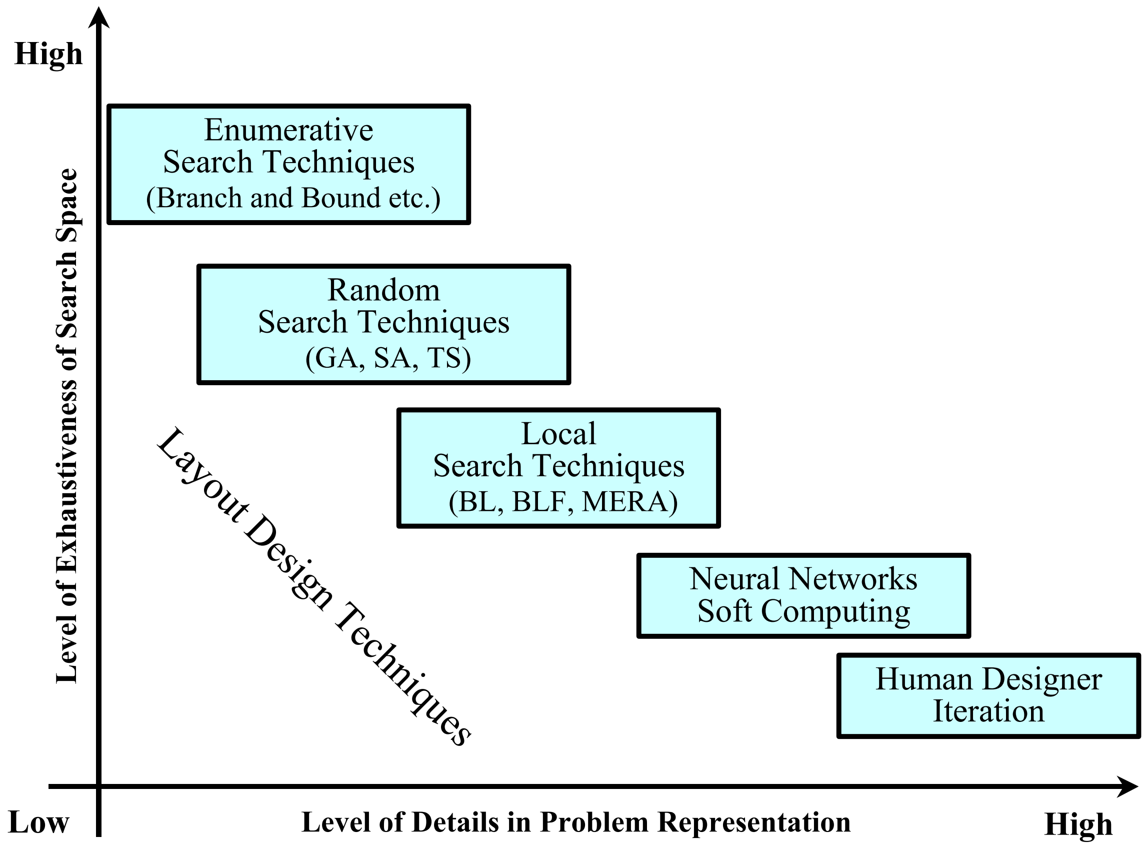


Figure 3-1: Flexibility and Robustness of various layout design approaches

into very meager level of search in the solution space. Between these two extremes are techniques that provide various degrees of flexibility through selection of tools, algorithms, and parameters that incorporate varying level of details in the representation of problem-specific information and design process. Conceivably, an intelligently formulated hybrid approach involving metaheuristics (random search), placement algorithms (local search), soft computing modeling and computational tools (approximate reasoning), and human intuition could deliver a higher degree of flexibility as well as superior outcomes.

In short, various modeling and computational tools and heuristics could help in characterizing possible outcomes, and the behavioral data may express some salient points about the designers' behavior and preferences (Moe & Fader, 2000). In this regard, computerized tools may be viewed as a mechanism for redistributing cognition (Walenstein, 2002). Indeed, provisions of some kind of decision support are largely *redistribution of cognition*. Such tools provide support through various means such as *process distribution*, *data distribution*, *plan distribution*, etc. (Walenstein, 2002).

Consequently, the emphasis of this thesis is not on the pursuit of some perfect system but rather on the development of a tool that could supplement the knowledge, experience, and design intuition and other cognitive resources of human layout designer.

Our research framework is based on the expert system paradigm for facilitating intelligent decision support in layout design, as depicted in Figure 3-2. Our selection of ES as a research paradigm is inspired by such inherent characteristics of an ES as the encoded knowledge, the separation of domain knowledge from the control knowledge, the ability to reason under uncertainty, the explanation facility, the knowledge acquisition capability, and the interactive user interface.

However, an efficient and effective means of tackling the subjectivity and uncertainty in the layout design problem requires complementing of the traditional ES paradigm, as shown in Figure 2-4, through various intelligent components. Such intelligent components in our research framework would afford effective, efficient, and robust means of capturing and utilizing subjective and uncertain design preferences, while generating a diverse suite of superior layout alternatives. Consequently, our research paradigm, as depicted in Figure 3-2, contains some components that are not associated with traditional expert systems. These include an Intelligent Layout Generator (ILG), a Preference Inferencing Agent (PIA), and a Preference Discovery Agent (PDA). It should be noted that this research framework evolved during the course of this thesis as more insights are about the structure of the problem at hand and underlying dynamics.

As already mentioned, an array of efficient algorithms for generating superior and diverse layout alternatives is an important step in automating the layout design process. Consequently, we use a hybrid fuzzy-genetic Intelligent Layout Generator towards this end. The intelligence aspect emerges from the employment of fuzzy rules/preferences in obtaining penalties and rewards for some composite genetic fitness evaluation function. Accordingly, a fuzzy Preference Inferencing Agent (PIA) seems to be a rational component for such an aiding tool.

However, layout design rules and preferences are both implicit and dynamic in nature. People learn new concepts and outgrow old ideas, thus pronouncing the necessity for re-learning of design rules by layout designers. Such an implicit and evolutionary character of preferences suggests that an online Artificial Neural Network based Preference Discovery and Validation Agent (PDA) could augment the overall power of the system by discovering some pattern of design rules and preferences in an automated and self-updated manner.

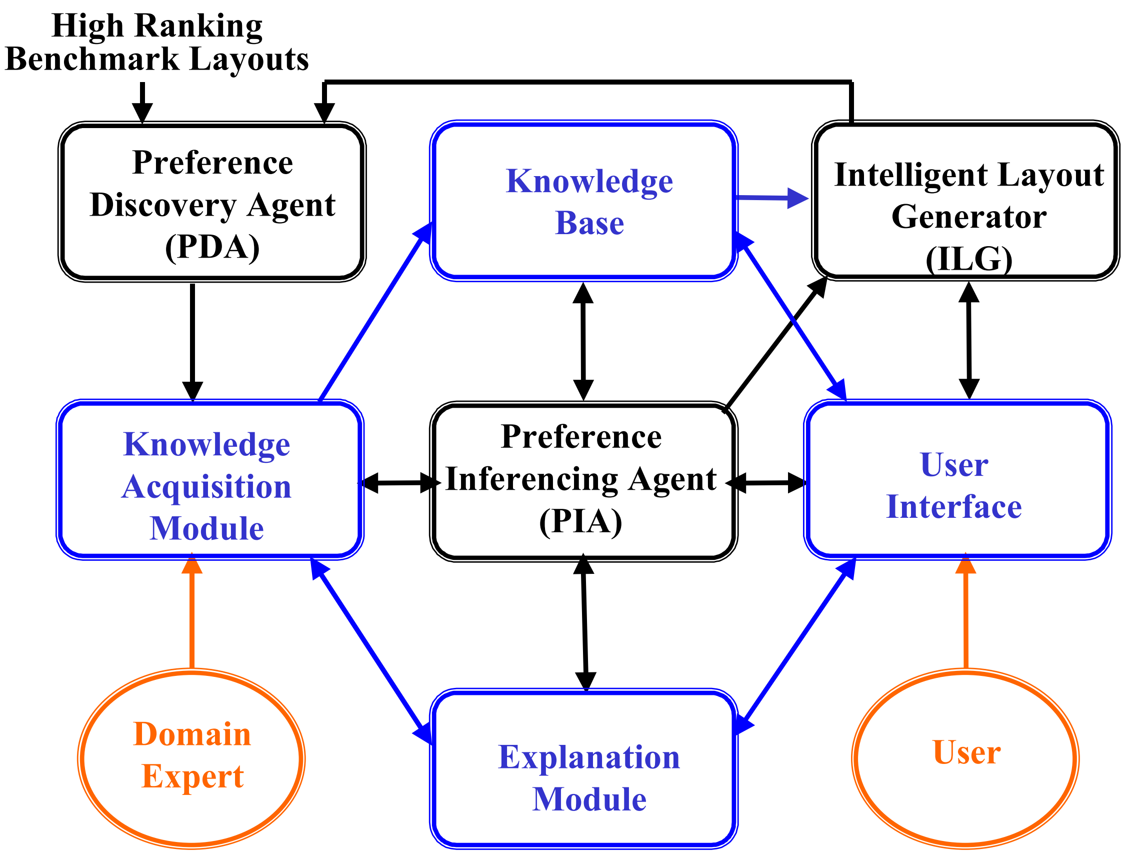


Figure 3-2: Intelligent System for Decision Support & Expert Analysis in Layout Design (IDEAL)

It should be mentioned that not all details of these components are made explicit in this framework for parsimony sake. For instance, our PDA is designed in a manner that it could furnish the learned knowledge in the form of usable knowledge by creating preference profiles of decision makers. As such, PDA would not require any explicit and separate knowledge acquisition module.

Through the employment of some meta-rules and heuristics, IDEAL has the capability to reason with uncertain and imprecise information and avoid the impractical search of an infinite solution space. An advantage of considering this kind of total solution approach is that, unlike usual partial layout construction procedures, it does not typically require backtracking over poor placement decisions. Indeed, backtracking is a computationally very expensive process (Ligget, 2000). In addition, backtracking procedures make the behavior of the system difficult to comprehend. The usual layout construction procedures do not have any robust means for pruning the search space, as there is no suitable evaluation measure for partial configurations allowing informed decisions in preferring one partial solution to another. In short, a system of spatial and functional relationships is a complex whole and it succeeds or fails as such (*Ibid.*). Consequently, we have adopted the approach of generating complete superior solutions for consideration by the decision maker.

3.3 Components of IDEAL

Here we provide further details of various components of IDEAL, including their philosophy and operation. Details regarding implementation of these components are tackled in subsequent chapters.

3.3.1 Intelligent Layout Generator (ILG)

The primary task involved in automating the layout design process is to produce superior layout alternatives for further consideration and treatment by decision makers (Akoumianakis, 2000; Tompkins *et al.*, 2002). In this regard, past studies have demonstrated that Genetic Algorithms provide a promising search and optimization approach (Abdinnour-Helm & Hadley, 2000; Ahmad *et al.*, 2004d, 2004e; Geigel & Loui, 2001; Kado *et al.*, 1995; Youssef *et al.*, 2003b). Our system incorporates experts' knowledge and user preferences in the layout design process through composite fitness functions of the ILG. This fitness function utilizes crisp preference weights furnished by the Preference Inferencing Agent.

Indeed, GA have been applied to the layout design problem in various modes (Chan & Tansri, 1994; Geigel & Loui, 2001; Mazumder & Rudnick, 1999; Mak *et al.,* 1998; Martens, 2004; Tate & Smith, 1995; Youssef *et al.*, 2003a). However, detailed discussions on these issues are beyond the scope of this thesis and some good surveys in these directions can be found elsewhere (Hopper & Turton, 2001; Kado, 1995). Nevertheless, most of the existing research applies GA in solving layout problem consisting of identical modules to be placed at identical locations. Such a problem can be treated as a relatively simpler one-to-one assignment of identical modules to the given cells/locations. In relatively advanced scenarios, the size of modules is considered fixed while leaving the determination of the shape of module to the solution procedure.

Still, some advanced research work employs GA in solving problems with fixed dimensional and oriented modules to be placed in a two-dimensional plane. However, employing GA in such more advanced and generic layout design scenarios requires efficient and efficacious decoding or placement heuristics. Such heuristics are important in order to generate layout alternatives in a timely fashion. Indeed, the importance of such pre-processor algorithms in terms of efficiency, efficacy, and reliability cannot be overemphasized. Various decoding or placement heuristics are available in the literature, for instance, BL (Dowsland *et al.*, 2002; Jakobs, 1996), IBL (Liu & Teng, 1999), BLF

(Chazelle, 1983), and DP (Leung *et al.* 2003). However, there is a relative dearth of decoding algorithms that are not only fast but also robust and effective in furnishing superior layout alternatives with higher aesthetic contents. In order to address this shortcoming, we have proposed some very effective decoding or placement heuristics. Details of these algorithms as well as our vision and implementation of ILG are provided in Chapter 4.

3.3.2 Preference Inferencing Agent

The brain of any ES is an Inference Engine that contains general algorithms capable of manipulating, and providing reasoning about, the knowledge stored in the knowledge base for solving problems by devising conclusions (Turban and Aronson, 2001). However, the inference engine in an ES is kept separate from the domain knowledge and is largely domain-independent.

A major problem in building intelligent systems is the extraction of knowledge from human experts who think in an imprecise or fuzzy manner. The same is true with the layout design problem where the knowledge associated with the layout decision analysis and design is usually imprecise, incomplete, inconsistent and uncertain. In the scope of our thesis, the term *imprecision* refers to values that cannot be measured accurately or are vaguely defined. Likewise, *incompleteness* implies the unavailability of some or all of the values of an attribute, *inconsistency* signifies the difference or even conflict in the knowledge elicited from experts, and *uncertainty* suggests the subjectivity involved in estimating the value or validity of a fact or rule.

The inherently vague, differing, and conflicting nature of most layout design guidelines and rules renders fuzzy technology an excellent candidate for modeling the system dynamics as well as implementation of the inference engine. Indeed, FL provides a means to work with these imprecise terms and has been successfully employed for automated reasoning in expert systems in various subjective and uncertain work-domains (Konar, 2000). However, little effort has been done in formalizing such application of fuzzy logic in ALD systems. Nevertheless, an FL based Preference Inferencing Agent seems to be an important component for any decision aid tool in the layout design (Ahmad, 2002, 2003; Karray *et al.*, 2001b; Raoot & Rakshit, 1993).

As such, the underlying concept in IDEAL's inferencing mechanism is the use of a Preference Inferencing Agent (PIA) comprising of fuzzy sets, rules and preferences for obtaining penalties and rewards in the layout fitness evaluation function for ranking and comparison purposes as well as for the automatic generation of layouts. The potential for utilizing FL arises from the fact that it provides a very natural representation of human conceptualization and partial matching. Indeed, the human decision-making process inherently relies on common sense as well as the use of vague and ambiguous terms. FL provides means for working with such ambiguous and uncertain terms (Negnevitsky, 2002; Nikolopoulos, 1997). Consequently, an FL based PIA is expected to deliver much of the flexibility in the automated layout design process that the layout design practitioners have always longed for. As such, we deem PIA as one of the core components, along with ILG, in tackling and automating the layout design process as well as in furthering the research in this important area. Further details of our vision and realization of the PIA are given in Chapter 5.

3.3.3 Preference Discovery and Validation Agent

The task of extracting knowledge from experts is extremely tedious, expensive, and time consuming. Furthermore, the subjective and dynamic nature of preferences in such domains as layout design frustrates the creation of an up-to-date knowledge base. In this regard, the importance of knowing decision-makers' needs and expectations through the quantitative analysis of behavioral data cannot be overemphasized. However, a traditional ES cannot automatically learn preferences or improve through experience.

Indeed, an automated learning mechanism could improve the speed and quality of knowledge acquisition as well as effectiveness and robustness of ES. Incidentally, Artificial Neural Networks (ANN) have been proposed as a leading methodology for such data mining applications (Vellido, 2002). ANN can especially be useful in dealing with the vast amount of intangible information usually generated in subjective and uncertain environments. The ability of ANN to learn from historical cases or decision-makers' interaction with layout alternatives could automatically furnish some domain knowledge and design rules, thus eluding tedious and expensive processes of knowledge acquisition, validation and revision (Marakas, 2002). Consequently, the integration of ANN with ES could enable the system to solve tasks that are not amenable to solution by traditional approaches (Negnevitsky, 2002).

Fortunately, the layout design problem renders itself to automatic learning of non-quantifiable and dynamic design rules from both superior layout designs and test cases. Furthermore, it is possible to automatically learn some decision-makers' preferences from their evaluation and manipulation of accepted or highly ranked layouts using some online ANN based validation agent. However, in the absence of core components like ILG and PIA, which would exploit the layout design preferences, an effective PDA could not be developed and tested. Consequently, we have given PDA a lower priority in developing IDEAL. Nevertheless, in Chapter 5, we have provided design and implementation of a small-scale prototype of PDA for demonstrating the viability of concept. In future, we intend enhance capabilities of our PDA and to employ Reinforcement Learning technology to complement ANN through incremental learning.

3.3.4 Knowledge Base

Knowledge is the primary raw material in an ES (Turban & Aronson, 2001). The conceptual model of the elicited knowledge needs to be converted to a format suitable for computer manipulation through a process called the Knowledge Representation (Marakas, 2002). The processes of knowledge elicitation and representation are not necessarily sequential. Typically, knowledge elicitation continues throughout the lifecycle of the system development and its usage because knowledge could be incomplete, inaccurate, and evolutionary in nature.

The knowledge of IDEAL consists of facts and heuristics or algorithms. It also contains the relevant domain specific and control knowledge essential for comprehending, formulating and solving problems. There are various ways of storing and retrieving preferences/rules including 'If-Then' production rules. The fact that representing knowledge in the form of such traditional production rules enhances the modularity of the system prompted us to adopt this approach. However, traditional and conventional logic based representation does not allow simple addition of new decision rules to the ES without any mechanism for resolving conflicts, thus resulting in inflexibilities that are not conducive to automated layout design systems (Chan & Lau 1997). This furnished another reason for our choice of fuzzy logic modeling preferences and building the inference engine for IDEAL.

3.3.5 Knowledge Acquisition Module

Knowledge acquisition is the accumulation, transmission, and transformation of problem solving expertise from experts or knowledge repositories to a computer program for the creation and expansion of the knowledge base (Turban & Aronson, 2001). It should be noted that knowledge acquisition is a major bottleneck in the development of an ES (Jackson, 1999). It is primarily due to mental activities happening at the sub-cognitive level that are difficult to verbalize, capture, or even become cognizant of while employing the usual cognitive approach of knowledge acquisition from experts (Marakas, 2002). Consequently, the task of extracting knowledge from an expert is extremely tedious and time consuming. It is estimated that knowledge elicitation through interviews generate between two and five usable rules per day (Jackson, 1999).

Knowledge could be elicited by domain experts or derived from the existing knowledge base. However, knowledge could be acquired automatically by the system itself using some machine learning mechanism. We intend to formulate our PDA in a manner that could provide knowledge about user preferences in a form readily usable by ILG and PIA. However, the automated knowledge acquisition has not been tackled rigorously in this thesis.

3.3.6 Explanation Facility

The ability to trace responsibility for conclusions to their sources is crucial to transfer of expertise, problem solving, and acceptance of proposed solutions (Turban & Aronson, 2001). The explanation unit could trace such responsibility and explain the behavior of the ES by interactively answering questions. For instance, an explanation facility enables a user to determine why a particular piece of information is needed and how intermediate or ultimate conclusions are obtained.

Explanation Facilities are very important from both developmental and marketing perspectives. For instance, the explanation facility is critical for both debugging of the knowledge base as well as user acceptance and adoption. These facilities may include user input help facility, design process information, and interrogation facilities. In its simplest form, an explanation facility could furnish the sequence of rules that were fired in reaching a certain decision. Indeed, the capability of an expert system to explain the reasoning behind its recommendations is one of the main reasons in choosing this paradigm over other intelligent approaches for the implementation of our concept.

Once again, a well-designed, interactive and effective user interface is an important ingredient in enabling a good explanation facility. In addition, incorporation of effective explanation capabilities is elusive without conducting a meticulously designed empirical study with actual users. However, such an extensive study is beyond the scope of this thesis. However, IDEAL contains a basic explanation capability through which experts can trace the sequence of rules that are used in arriving at certain conclusions. In the future, we intend to augment this explanation capability with even more informative and effective techniques.

3.3.7 User Interface

The user interface (UI) defines the way in which an ES interacts with the user, the environment, and such related systems as databases. The need for an interactive and user-friendly UI cannot be overemphasized and it is deemed to be an important factor in rendering the system easy to learn and easy to use. Indeed, "the interface is critical to the success of any information system, since to the end-user the interface is the system" (Head & Hassanein, 2002). Furthermore, research has shown that interface aesthetics as well as interactivity perform a larger role in users' attitudes than users would admit (Kuroso & Kashimura, 1995; Ngo, 2001). As such, the perceived usefulness of the interface, or users perception about the usefulness of the interface in a given work domain, plays an implicit role in longer-term user acceptance and performance (Ngo & Law, 2003; Preece *et al*., 1996; Shneiderman, 1997a). Accordingly, we strive for an interactive graphical user interface (GUI) for IDEAL.

Our GUI has the capability to accept input for the layout design from data files saved in text, csv, or Excel format (e.g. dimensions of packing space and modules as well as other parameters). It also has the provision for manual data entry or overriding of preferences from decision makers. Moreover, it enables fast and easy as well as informed and interactive manipulation of layout alternatives by the decision-maker. Some snapshots of Experts' User Interface and Knowledge Acquisition Modules as well as the prototype of end user interface are included in Appendix B for reference purposes.

Incidentally, our interface is still evolving. It is because IDEAL is still in the development stage and most of its existing functionalities are designed for developers. Consequently, some of its modules contain a higher degree of complexity to meet ecological requirements of system developers and experts. Indeed, experts operating in complex and dynamic decision-making ecologies prefer to have interfaces that are more complex, nevertheless, powerful (Burns & Hajdukiewicz, 2004). However, a prototype of an end-user interface has been developed, and tested, using the philosophy of Ecological Interface Design as well as various usability and Human-Computer Interaction guidelines. We employed a combination of digital and analog displays for increasing the efficacy of the interface. Furthermore, our design affords information about the context through various textual, graphical, analogical, and iconic references. Such an interactive interface could become the single most important factor to the eventual success of IDEAL.

Nevertheless, we intend to enhance the usability and interactivity of the interface in the near future. For instance, we could have a window showing one layout and another window showing the modules not included in the layout, enabling the decision maker to move modules in and out of the layout and/or rearrange them in the given layout while simultaneously observing changes in the fitness metrics used to rate that layout. In another mode of interaction, the user might be allowed to see a pair of highly ranked layouts for direct visual comparison and manipulation while observing the changes in fitness values in real time. Some mode of displaying contributions of various determinants of fitness in multi-criteria decision analysis as well as other experts' rating of a layout could augment both interactivity and efficacy of IDEAL. Indeed, IDEAL's interface affords intervention from decision-makers into the process of generating alternate layouts by modifying membership functions of preferences or weights in the fitness function etc. However, as IDEAL continues to evolve and remove constraints on what could be afforded in its various modes of interaction whould furnish creative ways in which they can support decision-makers' work.

3.4 Synergy of Intelligent Components

The proposed framework for IDEAL differs from a traditional ES by virtue of various intelligent components. Consequently, we deem it appropriate to elaborate the philosophy and synergic potential of such intelligent components, as these have been the primary focus of this research. This is because of our belief that these components furnish a significant amount of realizable automation in generating and manipulating superior layout alternatives by addressing the core issues in building the whole system. Furthermore, these components furnish a vehicle for carrying out further research in this direction. A somewhat detailed discussion of each intelligent component of IDEAL is provided in the following chapters.

The need for intelligent components arises from limitations of conventional systems design techniques that typically work under the implicit assumption of a good understanding of the process dynamics and related issues. Conventional systems design techniques fall short of providing satisfactory results for ill-defined processes operating in unpredictable and noisy environments such as layout decision analysis and design. Consequently, the use of such non-conventional approaches as Fuzzy Logic (FL), Artificial Neural Networks (ANN), and Genetic Algorithms (GA) is required.

The knowledge of strengths and weaknesses of these approaches could result in hybrid systems that mitigate limitations and produce more powerful and robust systems (Ahmad *et al.*, 2004b; Cordon *et al.*, 2004; Negnevitsky, 2002). Indeed, the potential of these technologies is limited only by the imagination of their users (Cordon *et al.*, 2004; Ruan, 1997).

Among the intelligent components of IDEAL, *Intelligent Layout Generator* (ILG) generates superior layout alternatives based on pre-specified and user-specified constraints and preferences as well as preference weights furnished by PIA. The *Preference Inferencing Agent* (PIA) incorporates the soft knowledge and reasoning mechanism in the inference engine. The *Preference Discovery*

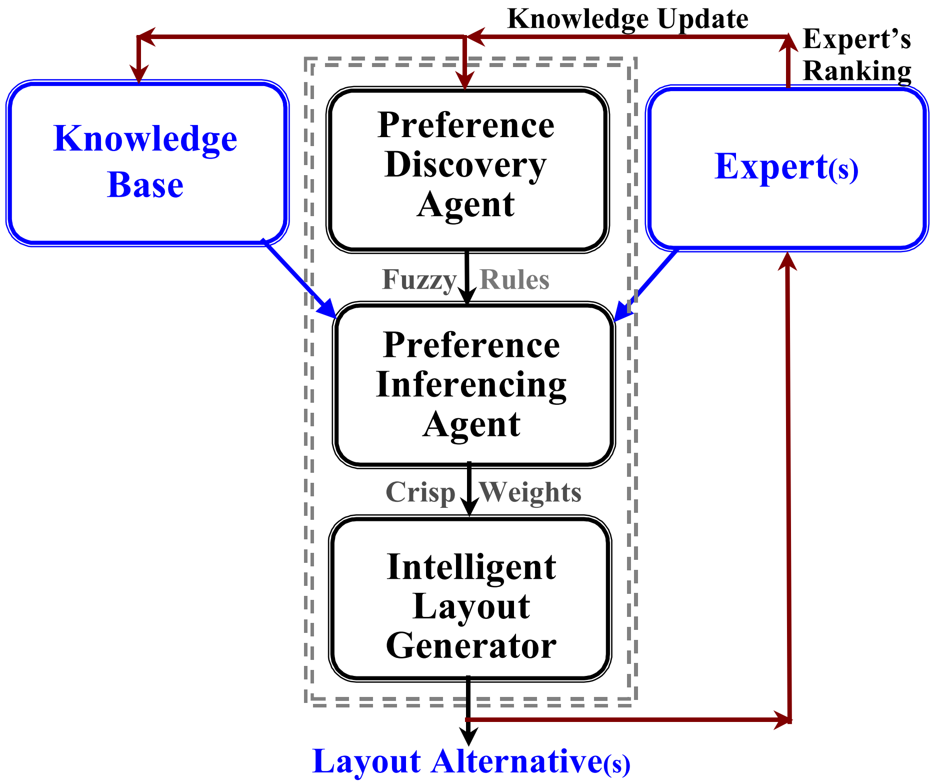


Figure 3-3: The Synergy of the Intelligent Components in IDEAL.

Agent (PDA) complements the ILG and the PIA by automatically discovering and refining some preferences.

The proposed synergy of the ILG, the PIA, and the PDA is shown in Figure 3-3. The PIA receives fuzzy preferences and rules from various sources including domain experts, the knowledge base and the PDA. These fuzzy preferences and rules are defuzzified by the PIA that, through its inferencing mechanism, furnishes crisp weights for use in the ILG. The ILG, in turn, generates superior layout alternatives for ranking and further manipulation by decision-makers. The layout alternatives generated by the ILG could be validated by the user or by the PDA. Consequently, the experts' ranking of layout alternatives serve as learning instances for updating and refining the knowledge-base, fuzzy rules, and preferences. Incremental learning technologies like Reinforcement Learning might prove useful here.

These intelligent components combine powers of the three main soft computing technologies representing various complementary aspects of human intelligence needed to tackle the problem at hand (Cordon *et al.*, 2004; Zha, 2003). The real power is extracted through the synergy of expert system with fuzzy logic, genetic algorithms, and neural computing, which improve adaptability, robustness, fault tolerance, and speed of knowledge based systems (Ahmad *et al.*, 2004c; Cordon *et al.*, 2004; Negnevitsky 2002; Zha, 2003).

We want to emphasize that these components have deliberately been designed to primarily have a generic character. The rationale behind this philosophy is our belief that a generic approach is more suitable in such subjective, uncertain, and dynamic problem domain as layout design that has applications in a diverse set of work domains. Consequently, a generic approach would result in minimal efforts from design engineer in adapting the system for any specific layout design problem.

3.5 Data Requirements

IDEAL requires various sorts of knowledge. These include the knowledge about *components*, such dimensions of packing space, number of modules, dimensions of modules, intrinsic utilities of modules, etc. Such knowledge is stored in a separate data file, which is distinct from the rule-base and may be seen as static knowledge.

In addition, IDEAL requires the knowledge about *preferences*, such as constraints, rules, inter-module flow relationships (which indicate adjacency requirements based on quantitative connectivity requirements), inter-module interaction relationships (which indicate adjacency requirements based on qualitative issues), etc. Such knowledge is stored in a rule-base in the form of production rules and may have both static and dynamic contents.

Furthermore, IDEAL requires knowledge about *procedures*, such as algorithms and heuristics. These include domain specific *control knowledge* such as attributes and applicability of various heuristics and metaheuristic search procedures and their control parameters. Such knowledge would facilitate good decisions about selection of procedures.

3.6 Summary

In this chapter, we have presented a framework for building an interactive Intelligent System for Decision Support and Expert Analysis in the Layout Design based on Soft Computing technologies. We have provided description of various components and data requirements of the system. Furthermore, we have explained the philosophy and synergy of the three intelligent components of the system. In the subsequent chapter, we provide details of design and implementation of an Intelligent Layout Generator with some promising placement heuristics.

Chapter 4

INTELLIGENT LAYOUT GENERATOR

4.1 Introduction

Endeavors towards automating the layout design process have a long history and numerous solution approaches, algorithms, and heuristics have been proposed in this direction. However, it is still a fertile research field due to the high degree of computational complexity and subjectivity involved. On one hand, computerized algorithms cannot substitute human intelligence, intuition, or erudition due to impediments in capturing subjective, qualitative, and political aspects of the work domain. On the other, computerized systems could increase the productivity and efficacy of the layout analyst through automated and fast generation, evaluation, and treatment of a large number of high-quality layout alternatives.

Consequently, the prime task in automating the layout design process is the generation of superior layout alternatives for consideration and manipulation by domain experts. As such, an Intelligent Layout Generator (ILG) for generating superior layout decision alternatives in an expedited manner is a core component in any automated decision support system for layout design. We deem the ILG as a stepping-stone in furthering the research in this area.

In this chapter, we present a Genetic Algorithms (GA) based approach for building such an ILG by employing various layout design heuristics, including some new ones that are fast and efficacious. The intelligence aspect comes from the employment of penalties/rewards or preference weights, furnished by a Preference Inferencing Agent, in the evaluation of a genetic fitness function.

It should be noted that we carried out preliminary experiments with various layout design problem formulation including QAP, QSC, and 2D-BPP. Furthermore, we employed several popular solution approaches including analytical and heuristic solution methodologies as well as such metaheuristics based search mechanisms as GA, SA, TS, NE, and RS, etc. Our preliminary studies resulted in the selection of 2D-BPP as the formulation for this thesis due to its more generic and natural characterization of the layout design problem. In addition, we adopted GA, in conjunction with some efficient placement heuristics, as a solution methodology due to its global scope and non-deterministic search mechanism as well as potential to furnish a diverse set of superior layout alternatives.

In short, these preliminary studies were the driving force in the selection of the approach we have employed in this thesis. This approach involves hybridization of the global search mechanism through GA and the local optimization through deterministic placement heuristics. Indeed, our approach has some innate characteristics, discussed later on, which are advantageous in providing effective decision support in layout design.

The rest of the chapter is organized as follows. Section 4.2 provides a general mathematical formulation of the problem at hand and some computational experience with LINGO^{TM}. Section 4.3 discusses the significance of Genetic Algorithms in generating superior alternate layouts. Section 4.4 describes the basic premise involved in metaheuristics based layout optimization approaches. Section 4.5 provides details about various GA operators and parameters as well as some effective and encompassing quantitative fitness evaluation functions. Section 4.7 provides some very promising decoding heuristics. Section 4.8 provides results of comparative studies of some existing and our proposed decoding heuristics. Section 4.9 provides some test cases for demonstrating the effectiveness of the research paradigm. Section 4.10 summarizes the chapter.

4.2 Mathematical Formulation

Here we provide a generic formulation of the two-dimensional oriented bin-packing problem (2D-BPP) where a finite number of rectangular modules M_i $(i = 1, 2, 3, \ldots, n)$ and m rectangular bins are given. The width and height of modules are w_i and h_i and all bins or packing spaces are rectangular in shape with standard width W and height H, respectively. An orthogonal packing pattern is desired that, by definition, involves a disjunctive placement of rectangles on the packing space in a manner that edges of each module M_i are parallel to x- and y- axes of the packing space. As all modules are oriented, rotations of modules are not allowed. The bottom-left corner of the bin is the reference point for the bin with coordinates $(x,y)=(0,0)$. The geometric center of a module M_i determines the spatial location of modules on the packing space for utility calculations, as depicted in Figure 2-1. The bottom-left corner of each module M_i is the reference point for the position of the module with coordinates (x_i,y_i). These determinants of module location, x_i and y_i are the decision variables besides the binary variable b_i that determines the inclusion or exclusion of module M_i in the packing configuration. Similarly, we assume that (X_c,Y_c) represent the reference point on the bin for the purpose of utility calculations. In this simplistic formulation, we assume that the mutual position of modules does not affect the utility of a packing pattern. All utilities can be calculated based on a module's distance from a pre-determined point of focus on the bin or the location of the module in the overall packing, as depicted in Figure 2-1.

Objective Function

The objective is to maximize the Total Utility of modules packed into the given packing spaces. The Euclidean distance between the point of focus, here the center, of the packing space and the center of a module M_i, denoted by δ_i, is given by:

$$\delta_i = \sqrt{(x_i + \frac{w_i}{2} - X_c)^2 + (y_i + \frac{h_i}{2} - Y_c)^2}$$

Equation 4-1

Alternatively, the Manhattan distance function, which is a very popular distance measure in when inter-module distances are small such as in a User Interface, can be employed. The Manhattan, rectilinear, distance measure would result in linear distance function and would be advantageous when some derivative based method is used for the exploration of the solution space. Indeed, a provision for users to readily select one of these two popular measures of inter-module distances would increase the scope of any ALD system (Ahmad, 2002; Mir and Imam, 2001). However, here we limit ourselves to the Euclidian distance measure for illustration and simulation purposes.

The goal for this simplified formulation is to *maximize* the *weighted sum of utilities of modules* minus *weighted loss due to unused area*. It could be mathematically expressed as follows:

$$Max. \quad \left[\sum_{i=1}^{n} u_i w_i h_i b_i - \zeta \left(mWH - \sum_{i=1}^{n} w_i h_i b_i \right) \right]$$

Equation 4-2

Here, u_i is a function of intrinsic utility (α_i) and the spatial location (x_i,y_i) of the module M_i in the packing space; ζ is the disutility of unused/waste space; and, m is the number of bins required to pack given modules (in situations where more than one identical packing spaces available). This expression could be simplified into the following:

$$Max. \quad \left[\sum_{i=1}^{n} (u_i + \zeta) w_i h_i b_i - \zeta mWH \right]$$

Equation 4-3

Boundary Conditions

A given module M_i lying within the boundary of the packing space should satisfy the following boundary constraints.

$$x_i \leq W - w_i \qquad \text{Equation 4-4}$$

$$y_i \leq H - h_i \qquad \text{Equation 4-5}$$

$$x_i, y_i \geq 0 \qquad \text{Equation 4-6}$$

These constraints can easily be adapted when there are multiple and identical packing spaces available by incorporating a binary decision variable b_i that is set to 1 when a module is included in the packing space and set to 0 when the module is not a part of a partial or final layout solution.

$$\left.\begin{array}{l} x_i \leq (W - w_i)b_i \\ y_i \leq (H - h_i)b_i \end{array}\right\} \quad \forall\, i = 1,2,...n. \qquad \text{Equation 4-7}$$

Where

$$b_i = \begin{cases} 1 & \text{if } M_i \text{ is included} \\ 0 & \text{elsewhere} \end{cases} \quad \forall\, i = 1,2,...n. \qquad \text{Equation 4-8}$$

No-Overlap Constraints

In order to ensure that a module M_i does not overlap with a module M_j, satisfying any one of the following four conditions would suffice for all *i, j =1,2, … n* and $i \neq j$.

1. The module M_i is completely on the left of the module M_j.
2. The module M_i is completely on the right of the module M_j.
3. The module M_i is completely below the module M_j.
4. The module M_i is completely above the module M_j.

The first condition can be modeled using the following constraint:

$$-x_i + x_j \geq w_i$$

A similar approach would result in the following four constraints corresponding to each of the four conditions listed above:

$$-x_i + x_j \geq w_i \qquad \text{Equation 4-9}$$

$$x_i - x_j \geq w_j \qquad \text{Equation 4-10}$$

$$-y_i + y_j \geq h_i \qquad \text{Equation 4-11}$$

$$y_i - y_j \geq h_j \qquad \text{Equation 4-12}$$

It can be seen that Expressions 4-9 and 4-10 are mutually exclusive. The same is true for expressions 4-11 and 4-12. If any one of these four constraints is satisfied then it is a sufficient condition to confirm a non-overlap (Ahmad, 2002).

Therefore, this either-or set of constraints can be transformed into the standard Linear Programming Formulation by subtracting a big constant D from the R.H.S. of these requirement type constraints:

$$\left.\begin{array}{l} x_i - x_j \geq w_j - D(1-d_1) \\ -x_i + x_j \geq w_i - D(1-d_2) \\ y_i - y_j \geq h_j - D(1-d_3) \\ -y_i + y_j \geq h_i - D(1-d_4) \\ \sum_{l=1}^{4} d_l \geq 1; \quad \sum_{l=1}^{2} d_l \leq 1 \quad and \quad \sum_{l=3}^{2} d_l \leq 1 \\ d_l = 0, 1 \qquad \forall\, l = 1,2,3,4 \end{array}\right\} \qquad \text{Equation 4-13}$$

As already mentioned, in order to ensure that a module M_i does not overlap with a module j, any one of the four conditions listed in expression 4-13 should be satisfied for all $i, j = 1,2, \ldots n$ and $i \neq j$. In the following, we provide proof for this axiom.

4.2.1 Simplified Single Bin Formulation

This formulation can be simplified for a case where layout for only one packing space or bin is being considered. This would result in the following formulation:

$$Max.\left[\sum_{i=1}^{n} (u_i + \zeta) w_i h_i b_i - \zeta WH\right]$$

Given that:

$$\begin{array}{l} u_i = \alpha_i (1 + \beta_i) \\ \beta_i = (\delta_{\max} - \delta_i) / \delta_{\max} \\ \left.\begin{array}{l} x_i \leq (W - w_i) b_i \\ y_i \leq (H - h_i) b_i \end{array}\right\} \qquad \forall\, i = 1,2,\ldots n. \end{array}$$

$$\left.\begin{array}{l} x_i - x_j \geq w_j - D(1-d_1) \\ -x_i + x_j \geq w_i - D(1-d_2) \\ y_i - y_j \geq h_j - D(1-d_3) \\ -y_i + y_j \geq h_i - D(1-d_4) \\ \sum_{l=1}^{4} d_l \geq 1; \quad \sum_{l=1}^{2} d_l \leq 1 \quad and \quad \sum_{l=3}^{2} d_l \leq 1 \end{array}\right\} \text{For all } i, j = 1,2, \ldots n \text{ and } i \neq j.$$

$$x_i, y_i \geq 0$$

$$b_i, d_l = 0,1 \quad \forall\, l = 1,2,3,4;\, i = 1,\ldots,n$$

We employed this simplified formulation in our comparative studies of placement algorithm as well as implementation of the ILG. However, in order to learn some insights about the structure of the problem, we employed such popular and powerful general-purpose optimization software as LINGO™ (Industrial Version 7.0). We used an Intel Xeon 3.06 GHz processor with 256MB of RAM under Windows XP. The computational experience is summarized in Table 4-1.

In these experiments, inputs were dimensions and utilities of modules and outputs were coordinates on the packing space where the modules were placed. The first column in Table 4-1 provides the size of layout design problems tested, varying *n* between 4 and 10. Furthermore, we treated coordinates *(x, y)* as both Integer (I) and Continuous(C) variables, as indicated in the second column. The third column contains the time spent by the LINGO in solving the problem. The fourth column depicts the quality of solution (global vs local optimum). The fifth column provides some comments on these solutions based on visual evaluation.

Size	Type	Time	Quality of Solution	Comments based on visual evaluation
4	C	9 min.	Local Optimum	Poor solution; Poor space utilization
4	I	5 min.	Local Optimum	Poor solution; Poor space utilization
6	C	32 min.	Local Optimum	Poor solution; Poor space utilization
6	I	28 min.	Local Optimum	Poor solution; Poor space utilization
10	C	> 7 days	No feasible solution	Several trials resulting in similar fate.
10	I	> 7 days	No feasible solution	Several trials resulting in similar fate.

Table 4-1: Computational Experience with LINGO for 2D-BPP.

These results were largely as per expectations as the problem is known to be strongly NP-Hard. It should be noted that constraints restricting the overlap are dynamic in character unlike constraints in normal non-linear optimization problems. These constraints depend on the current topology of the modules at any optimization stage (Mir & Imam, 2001). Any perturbation, however small it may be, in the current topology of the layout results in an entirely new problem and hence an entirely new solution space without any discernible change in the topology.

Moreover, even a small-scale problem required large amount of empty space for the movement of modules. In these experiments, a tighter packing space that was sufficient to comfortably accommodate all the modules always resulted in infeasible solution. A sample problem and its layout solution, obtained from LINGO, are shown in Table 4-2 and Figure 4-1 for reference. For this reason and because of other inherent characteristics of the problem, the typical methods of solving constrained non-linear optimization problems cannot be useful (Imam & Mir, 1989). In conclusion, the exact algorithms are unpromising; consequently, there is a need for some efficient module placement strategy that could effectively reduce the size of the search space to some tractable subset of solutions.

An unexpectedly thorny observation was the long time taken by LINGO in coming up with an initial feasible solution/layout even in cases where number of modules was very small and the size of the packing space was quite large compared to the size of individual modules. For test cases studied, it is observed that the time needed to obtain a layout (optimal or sub-optimal) in the 2D-BPP using LINGO is exceedingly prohibitive. In short, exact solution approaches are not promising. Conceivably, the whole exercise suggests the need for some efficient module placement strategies, which could effectively reduce the size of the search space to some tractable limit.

$n = 6; W = 10; H = 10; \zeta = 1;$ Objective = 491.1					
i	w_i	h_i	u_i	x_i	y_i
1	1.9	2.5	5	3.09	2.45
2	2	2.5	6	1.45	6.27
3	2	1	10	3.46	4.96
4	1	0.75	14	2.08	4.2
5	4	1.9	12	5.47	4.36
6	5	3.8	8	5.00	0.55

Table 4-2: Solved Example for 2D-BPP.

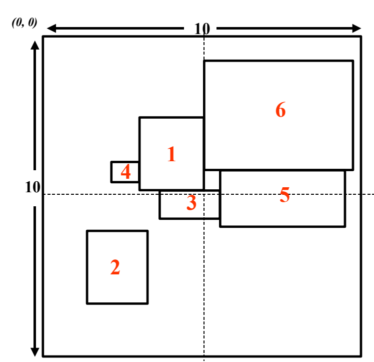


Figure 4-1: Solved Example for 2D-BPP.

The subsequent sections provide details of a Genetic Algorithm formulation and a range of proposed module placement strategies that are expected to solve large layout design problems reasonably fast. It should be noted that the given exact formulation can be easily and quickly adapted for such intelligent search techniques as Genetic Algorithms.

4.3 Genetic Algorithms in Layout Design

The high degree of subjectivity, multiplicity of evaluation criteria, and combinatorial complexity of the layout design problem indicates the difficulty in automating the layout design process. Intuitively, metaheuristics have been successfully employed and studies have demonstrated the efficacy and promise of such approaches (Mazumder & Rudnick, 1999; Lee & Lee, 2002; Martens, 2004; Tate & Smith, 1995; Wu *et al.*, 2002;). Among such metaheuristics, GA are the most frequently elected and researched ones (Ahmad *et al.*, 2004f; Dowsland *et al.*, 2002; Hopper & Turton, 2001; Tate & Smith, 1995). We have elected for GA in developing our ILG due to various inherent features, which are advantageous in procuring superior diverse layout alternatives. Incidentally, the diversity in layout alternatives is a critical factor in effective decision-making in the layout design.

Indeed, the freedom conferred by relatively relaxed requirements of GA in terms of problem formulation is also a big advantage. GA do not require much knowledge on the underlying rules or search space but simply a fitness function to evaluate how the fitness of solutions. Furthermore, GA may also be used for simultaneous and parallel optimization against several fitness metrics. As such, GA may be perceived as logically complementing the abilities of a knowledge-based system to reason about the application of different rules in dynamic, subjective, and uncertain scenarios, which is often the case with layout design problems.

4.4 Basic Premise

The core of our GA based approach is quite simple and involves treating the layout design as a packing problem. This process involves defining an ordering of modules (genotype) and a placement or decoding algorithm for obtaining the actual layout (phenotype) by placing modules in the given order (Ahmad *et al.*, 2004b; Dowsland *et al.*, 2002; Liu & Teng, 1999).

The efficiency and effectiveness of our approach is largely determined by the speed and effectiveness of the placement algorithm employed in decoding (Ahmad *et al.*, 2004b, 2004d, 2004f; Dowsland *et al.*, 2002; Jakobs, 1996; Liu & Teng, 1999; Wu *et al.*, 2003). Consequently, superior placement algorithms are critical for the usefulness of such an approach. However, existing placement algorithms lack the requisite efficiency and efficacy required in most layout design problem domains. These limitations become more pronounced when aesthetic values, such as symmetry, are among the determinants of layout utility. Consequently, an important step in designing and realizing an effective ILG is the development of some improved heuristics, especially ones that may provide layouts with higher aesthetic values.

4.5 Key Features

In addition to an efficient and effective decoding or placement heuristic, the GA based approach for solving the layout design problem involves determining several other problem-specific and generic critical features. The problem-specific decisions concern the fitness metric(s), initial solution/generation, encoding scheme, as well as the operators employed to navigate through the search space. Generic decisions comprise of the probabilities at which the search space navigation through crossover and mutation operators are applied, the population and generation sizes, and the stopping criteria.

4.5.1 Encoding Scheme

Our GA encoding scheme represents a layout using a sequence (or permutation when repetitions are not permitted) of modules. For example: *{12, 4, 11, 20, 9, 14, 2, 6, 13, 1, 15, 3, 18, 10, 7, 5, 19, 17, 16, 8}* shows a sequence of 20 modules to be placed in a given packing space. The total length l of a the sequence S, or chromosome, can be specified either by domain experts or through some algorithm based on a maximum number of modules that could be placed in a given packing space, amount of white space desired, etc. It has a significant role in determining the speed of the whole process. In our studies, we used a sequence length equal to the total number of modules, as we have not allowed leaving some modules out of the layout design. Such a representation of a layout solution as a sequencing problem permits the use of popular manipulation techniques for the searching the solution space. For instance, we may employ some order-based crossover operators instead of formulating some problem-specific operators that can only be applicable in one specific context.

4.5.2 Population Size

Population Size (P) also has significant role in determining the speed of the process. However, our preliminary experiments as well as earlier studies demonstrate that even a modest population of the size 10 to 20 could afford good results without expending computational resources unnecessarily (Ahmad, *et al.*, 2004d, 2004f; Tate & Smith 1995, Jakobs 1996). Consequently, we elected to employ a pre-specified and static population size during the evolution process. As such, the initialization step in the GA randomly generates *P* sequences of modules ($S_1, S_2, \ldots, S_P$) to create the initial population.

In our studies, we elected to employ a population size of 50. A larger initial population may be unnecessary since the GA automatically generates new members of the population in the process of searching the solution space. Alternatively, we can have a population pool that grows steadily with each generation (i.e. number of children slightly exceeds the decreased number of parents).

4.5.3 Genetic Operators and Parameters

In GA, genetic evolution of the population creates new solutions through genetic operations on individuals from previous generation. In the layout design context, genetic operations produce new layout solutions through evolution of existing solutions (crossover and mutation of individual layouts from previous generations). The means of performing these operations must be defined for the layout design problem. Since an order based encoding is employed, it must be ensured that valid chromosomes are engendered by these operations. A variety of genetic operators could be suggested for the GA.

We propose a new crossover and a couple of new mutation operator suitable for layout design problem. Among the published genetic operators, we limit ourselves to genetic operators similar to those employed by Tate & Smith (1995) and Jakobs (1996) for our application. These constitute a popular extract of possible operators. The final set of evolution operators (selection, crossover, mutation, and replacement) and parameters (population size, crossover rates, mutation rates, and termination criteria) are determined after some preliminary trials. Nevertheless, our preliminary studies as well as past studies show that the effectiveness of GA methodology remain largely insensitive to the exact values of such parameters (Ahmad, *et al.*, 2004d; Leung *et al.*, 2003; Tate & Smith, 1995).

Selection Operator

The selection operator selects individual layout solutions for genetic evolution. There is a diverse set of selection strategies available in the literature. However, a rank based elitist selection strategy, commonly known as biased Roulette Wheel selection, is one of the most common selection strategies with a bias towards selecting the fitter solutions (Negnevitsky, 2002). Selecting fittest individuals is also a popular strategy. Another common strategy is the random selection.

Intuitively, the rank based strategy should be more efficient as it gives more weightage to fitter solutions. However, as already mentioned, our ultimate goal is to develop an automated layout alternative generator as a decision and design aid for layout design. Such a system would be more effective when an array of superior but diverse layout alternatives available to decision makers. As such, a random selection strategy might be deemed more effective, if not efficient, for such systems. It is because a rank based selection might result in faster convergence to a few relatively inferior solutions.

However, our preliminary studies did not show any systematic difference resulting from the choice of selection strategy. Although we have allowed decision-makers to override the default option and employ others selection strategies, we adopted the biased Roulette Wheel selection strategy for our studies. The Roulette Wheel selection culls individual solutions with a probability given by:

$$p_i = \frac{f_i}{\sum_{j=1}^{P} f_j} > 0$$

where f_i is the fitness value of layout solution i Equation 4-14

In practice, the interval I = [0, 1) is divided into P subintervals such that each individual layout solution is assigned a sub-interval as follows:

$S_1 \leftrightarrow I_1 = [0, p_1)$,

$S_2 \leftrightarrow I_2 = [p_1, p_{1+}p_2)$,

:

$S_P \leftrightarrow I_2 = [1- p_P, 1)$.

Selection is done by generating a random number in the interval [0, 1) and the corresponding interval determines the individual layout. It can be seen that this is very much like spinning a roulette wheel where each individual layout has a segment on the wheel proportional to its fitness giving it the name Roulette Wheel selection.

Mutation Operator

In mutation, altering a single solution generates new individuals. In the context of the layout design problem, mutation results in diminutive changes in an existing layout. In itself, mutation amounts to random search of solution space with an incremental random modification of the existing layout and accepting it if there is an improvement. In the GA, crossover operator is the most efficient search mechanism. However, crossover itself does not guarantee the accessibility of the entire search space with a finite population size. The mutation operation makes the entire search space reachable, despite a finite population size.

The mutation rate is the probability of mutating a selected chromosome. If mutation rate is very low then there are higher chances of being trapped in a local optimum. However, if the mutation rate is very high then there would be too many random perturbations and offspring might lose their resemblance to the parents. Based on our preliminary studies, we selected the mutation rate to be high (more than 50%). The requirement of a diverse set of superior solutions entails such a higher mutation rate to ensure diversity in the population of layouts and prevent the population from becoming too homogenized. Indeed, these observations conform to results reported in past studies (Tate & Smith, 1995; Leung *et al.*, 2003). The following mutation operators are used in the proposed GA:

1. Tate and Smith (1995) proposed following set of mutation operators:
    a. Reverse the subsequence of the sequence in the mutating layout solution with random selection of the mutating solution and mutating subsequence.
    b. Add a new breakpoint resulting in splitting of a sequence or layout configuration by moving a random subsequence of modules to a new packing space or bin.
    c. Remove a breakpoint resulting in merger of two layout subsequences into one packing space or bin; i.e. consolidation of modules by reducing the number of bins.
2. Jakobs (1996) used the following set of mutation operators:
    a. Same as in 1(a).
    b. Exchange elements of two randomly selected layout subsequences.
3. Remove a subsequence of random length from the beginning of the sequence and append it at the end of the sequence.
4. Replace a randomly selected module with another randomly selected module from *0, 1, 2, … n*. Where the module ‘zero’ represents a dimensionless dummy module ensuring that the real number of modules in a sequence may even be reduced (thus reducing the number of modules placed in a given packing space). A tabu list may be maintained to avoid introduction of a recently removed module back to the layout for a specified number of mutation iterations. Such an approach has similarities with the Tabu search mechanism

discussed in Section 2.5 and would increase the probability of diversity among solutions. However, such a random swapping of modules might not be feasible due to come proximity or usability constraints. Nevertheless, such prohibited moves would result in the debased fitness value of the resulting solution and thus reduce the probability of being selected for further evolution.

It should be noted that the first set of mutation operators is applicable only for multiple packing space layout design problem. While, the second and third set of mutation operators could be applied in both single and multiple packing space layout design problem.

Crossover Operator

During crossover, one or more offsprings are derived from two or more parents. In the layout design context, crossover results in combining parts of two existing layouts in order to generate a new layout. In this regard, an appropriate set of crossover operators as well as crossover rate is needed. A higher crossover rate, or probability, permits a more extensive exploration of the solution space and reduces the chances of trapping in a false optimum. On the other hand, a lower crossover probability enables exploitation of superior solutions, or search points, in the existing population. In view of the highly subjective nature of most layout design applications, we selected a crossover rate of 20% in order to exploit existing solutions more.

The following crossover operators are used on two parents (say S_j and S_k) selected randomly based on their ranks in the population.

1. Tate and Smith (1995) Crossover consists of following steps:
    i. Fill each position in the offspring layout by randomly selecting a gene present at the same position from the first or second parent layout (resolving conflicts to guarantee that the result was in fact a sequence).
    ii. Insert leftover genes in order (or in random order) to fill in the blanks (unresolved conflicts).
    iii. Borrow breakpoints set randomly from one of the parents.
2. Jakobs (1996) Crossover consists of following steps:
    i. Copy q elements of the sequence S_j at a random position p in the new sequence S_{new}.

It should be noted that $l \leq p, q \leq n$.

ii. Fill up the remaining elements of S_{new} with other elements of S_k in the same order.

3. Append a randomly selected subsequence from one parent to another, overwriting the trailing subsequence in the second parent (possibly, with an additional breakpoint) and removing any duplication, if necessary.

It should be noted that the first crossover operator is applicable only for multi-bin layout design problem. While, the second crossover operator is applicable only for a single bin layout design problem. Furthermore, the second crossover is a sort of a drastic mutation operator.

Replacement Strategy

A traditional GA generates *P* offspring layouts before sorting out the poor ones by selection. We argue that module placement strategies are computationally very costly; consequently, we propose that GA sort out the worst individual after a new offspring layout is created on an ongoing basis. As a result, fitter offspring could influence the layout solution quality (an approach similar to the one used by Jakobs, 1996). This strategy results in a *steady state* or incremental GA as opposed to a *generational* GA where multiple offspring are created to replace the current population.

4.5.4 Fitness Function

The most taxing and application specific task in employing GA is the definition of fitness function(s) (Al-Sultan *et al.*, 1996). The fitness function is used to differentiate between a *superior* and an *inferior* layout solution. It should be selected with great prudence, as the GA will converge on layout solutions deemed *fit* against the fitness function employed. In the layout design context, the best, though somewhat impractical, approach would be to let experts or users judge the fitness through visual evaluation (Ahmad *et al.,* 2004b; Lok & Feiner, 2001). However, one of the important objectives in developing IDEAL is to minimize user inputs once the preliminary preferences have been identified through various means. We discuss details of our fitness metrics in Section 4.6.

4.5.5 Termination Criteria

We opted to terminate the GA iterations when the improvement in the fitness of a new population over the preceding population is less than a certain value (say 0.1%) for a pre-specified number of iterations. Typically, we terminated the GA after 1,000 to 5,000 iterations, as experience revealed that the layout fitness obtained after few thousand iterations is only marginally different from those obtained by 50,000+ iterations. Moreover, some marginal gain in fitness with regard to some rigid and myopic fitness metric offer little meaning and practical advantage in such subjective and uncertain domains. Nevertheless, the termination criteria can readily be changed by the user of IDEAL. Furthermore, experience has shown that very large number of GA iterations is more likely to converge on few closely related layout alternatives, defeating the objective of having a diverse set of superior layout alternatives.

4.6 Fitness Evaluation Metrics

As already noted, the layout design problem involves such a plethora of subjective and uncertain considerations that no single objective could solely be used to generate layout alternatives. However, automated layout design systems require some fitness quantification and evaluation mechanism in order to guide the search to superior solutions (Ligget, 1992). We, therefore, propose the use of some hybrid fuzzy-genetic fitness function that would combine multiple objectives in terms of rewards/penalties arising from various layout design considerations. As such, various determinants of the layout utility are combined through some crisp weights or Significance Parameters (SP) to penalize deviation from the desired values or Preference Parameters (PP). These significance and preference parameters may be determined by the layout planners or through the PIA using the existing knowledge. The development of such a multi-attribute layout fitness function is dealt with in Section 5.4. As a preliminary research model, we envisaged the following major categories of design preferences/parameters as determinants of layout fitness:

a. Intrinsic Utility of a module
b. Inter-Module Interaction
c. Space Utilization
d. Qualitative Fitness or Aesthetic Appeal

Despite all the subjectivity and uncertainty involved in calculating the intrinsic utility of a module, the inter-module interaction, and the space utilization, we classify these as quantitative measures of layout fitness. The rationale is that these measures may be quantitatively captured in an automated or semi-automated fashion with relative ease, given that the required data is complete and known with certainty. Nevertheless, the evaluation of aesthetic appeal cannot be readily measured using some tangible factors. Consequently, we classify determinants of the layout aesthetics as qualitative measures of layout fitness. Here, we describe these fitness measures in details.

4.6.1 Intrinsic Utility of Module

Intrinsic utility of a module is the utility a module brings when it is included in a layout design. Although it forms an important component of the total fitness of the layout, for simplicity sake and without any loss of generalization, we ignore inherent utility of a module in our prevailing discussions. It is because our subsequent comparative analyses involve single bin scenarios where all modules need to be included in a packing configuration. As such, the intrinsic utility of a module would only constitute a constant term in the total fitness.

4.6.2 Inter-Module Interaction

The inter-module interaction is an important determinant of layout fitness in many layout design work domains. For instance, this inter-module interaction may define the material, personnel, or information flow from one facility to another in the facility layout design context. In such scenarios, it will be an important factor in determining the material, personnel, and information transfer costs of operations. For instance, it is estimated that about 25% of operating costs in manufacturing facilities are due to materials handling (Tompkins *et al.*, 2002). Another instance would be the wire length and routing in VLSI circuit design that is usually done after macrocell placements. These wire lengths and routings may deteriorate the circuit performance through such nuisances as crosstalk and signal delays (Youssef *et al.*, 2003).

Indeed, we consider inter-module interaction as an important determinant of layout fitness. Consequently, IDEAL has been equipped with functionalities for modifying these inter-module interactions through editing values in an *interaction matrix* (also termed as connectivity matrix or flow matrix in some layout design work domains). The interaction matrix provides the interaction between all pairs of modules. An element of this matrix is denoted by $f_{i,j}$ and represents the flow between any two modules M_i and M_j.

The $f_{i,j}$ may or may not be same as $f_{j,i}$, i.e. the interaction matrix may also be asymmetric. Consequently, IDEAL has been equipped with the flexibility in formulating both symmetric and asymmetric interaction matrices. For elaboration purposes and without any loss of generalization in our comparative analyses, we assume that the inter-module interaction among all modules is identical and unity. Furthermore, we calculate it as the sum of mutual distances between geometric centers of all pairs of modules or the Total Inter-Module Distances (*IMD*). This specific case of our general inter-module interaction constitutes a popular fitness evaluation metric in the literature.

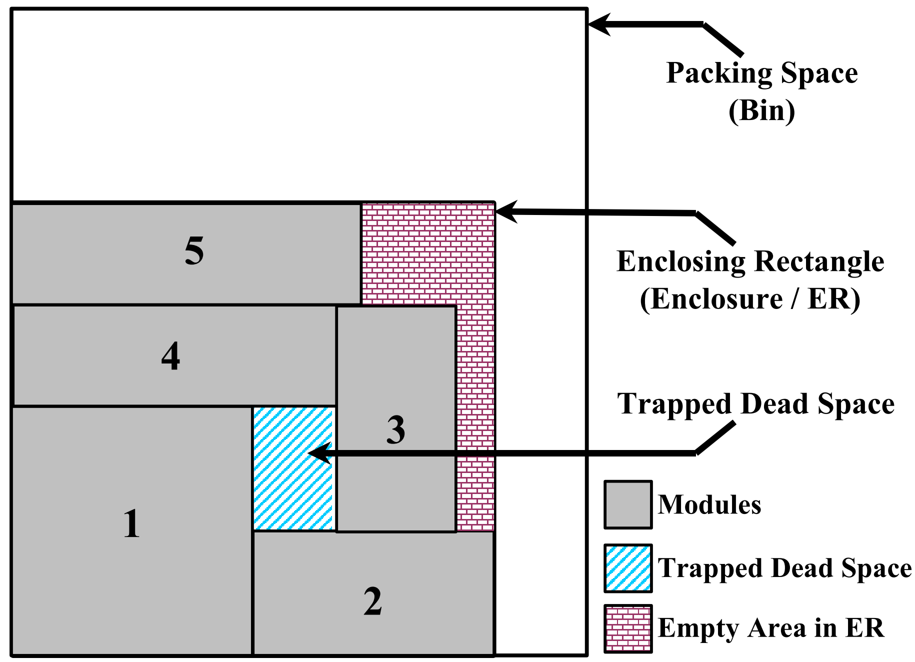


Figure 4-2: Basic Bin-Packing Terminology

4.6.3 Space Utilization

Space utilization is among the more popular layout design fitness metrics. Apparent reasons for such popularity may include the relative ease in automating the quantification of layout fitness. Nevertheless, here too the inherent subjectivity and uncertainty involved in layout designing often complicates the issue. In order to facilitate the foregoing discussion, we have elaborated some basic terminology used here in Figure 4-2.

In Figure 4-2, the large rectangle represents the bin or packing space. Within this packing space, we find another rectangle, which is the smallest rectangle that fully encompasses the other five smaller rectangular modules placed in the layout. We term this smallest rectangle that encompasses the whole layout as Enclosing Rectangle (*ER*). In the given packing configuration, we see some empty spaces that are surrounded by modules on all sides. We term such an empty space as Dead Space.

With this basic terminology, we proceed to the definition of fitness metrics. Indeed, there are numerous measures of space utilization available in the literature including the height of the packing, the module tightness, the amount of dead space, the amount of usable spare space, etc. Evidently, all these measures of space utilization have merits and demerits, as revealed by the following discussion, and no single measure can be deemed as truly and fully capable of capturing the space utilization in a given layout configuration. Nevertheless, we have employed the Packing Height (*HT*), the Contiguous Remainder (*CR*), and the Module Tightness (*MT*) for our comparative evaluation of decoding heuristics in terms of space utilization.

The underlying principles for selecting these metrics involved rationality of the concept, consistency and robustness, feasibility and ease of implementation, as well as popularity within the scientific community. Indeed, these metrics, along with inter-module interaction, are found to provide a rational regime for gauging the space utilization. Nevertheless, our experiments also reveal that these metrics capture different rational notions of space utilization, as discussed in the following paragraphs.

Height (*HT*)

The Height (*HT*) of the packing pattern is the most widely used measure of space utilization (Ahmad *et al.*, 2004d; Hopper & Turton, 2001; Leung *et al.*, 2003). It is based on the notion that a lower packing height implies availability of more spare space for further placement of modules. Nevertheless, this measure of space utilization has serious shortcomings. These shortcomings can be illustrated from Figure 4-3 where both layout topologies have identical heights. Accordingly, the two packing patterns have the same fitness value in terms of *HT*. However, layout *B* might be deemed more fitting in terms of space utilization. In short, some packing pattern might be deemed superior to other packing patterns of the same *HT* in terms of other objectives like sum of distances from some focal point, inter-module interactions, etc.

Apparently, the ease in quantifying height of a layout pattern in an automated manner has been the major contribution to its popularity (Dowsland *et al.*, 2002; Jakobs, 1996; Hopper & Turton, 2001). Consequently, we have included *HT* as one of the measure of space utilization for the ease of comparison with earlier heuristics. In computational terms, the negative of *HT* is employed as the GA fitness function.

In order to facilitate use of *HT* in some hybrid GA fitness metric for MCDM, we need to normalize it against some suitable benchmark value. Towards this end, we introduce the notion of "Benchmark Packing Height" (*BPH*). We calculate the *BPH* as ratio of the Total Module Area and the Width of the packing space, as shown in the following Equation:

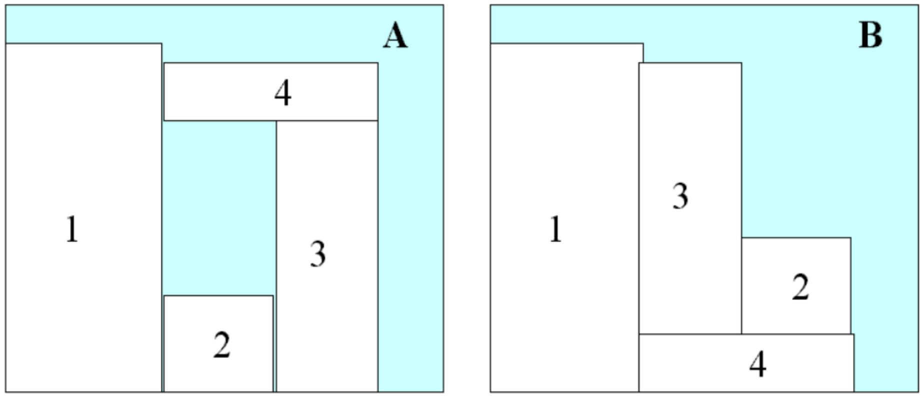


Figure 4-3: Two Layout Patterns with the Same Height

$$BPH = \frac{\sum_{i=1}^{n} w_i h_i}{W} \qquad \text{Equation 4-15}$$

In reality, it is a lower bound on the packing height, which is possible only if all the modules are in a manner that there is no dead space or spare space in the enclosing rectangle (i.e. 100% *MT*). Thus, the actual packing height in any practical problem is usually greater than *BPH*. Using this *BPH,* we may calculate a normalized measure of packing height ($\hat{H}T$) in the following Equation:

$$\hat{H}T = \frac{BPH - HT}{BPH} \times 100 \qquad \text{Equation 4-16}$$

This normalization scheme was adopted after experimentation with the behavior of GA with several other normalizations. With the above normalization, the Packing Quality will be 100% in a hypothetical supreme condition. In all practical cases it will be less than 100 and may even be negative for extremely poor packing. In IDEAL, user has an option to readily select either *HT* or its normalized type $\hat{H}T$.

Contiguous Remainder (*CR*)

The research literature has long been proposing the Contiguous Remainder (*CR*) or the 'reusable trim loss' as a more appropriate measure of space utilization (Jakobs, 1996; Liu & Teng, 1999). The *CR* refers to the largest contiguous vacant portion of the packing space available for additional module placements (Ahmad *et al*, 2004d, 2004f; Jakobs, 1996; Liu & Teng, 1999). In other words, *CR* is the empty area on a given bin or packing space outside the edges of the boundaries created by the packed modules in a layout configuration, as depicted in Figure 4-4. Conceivably, a larger value of *CR* implies that more space is available for further placements.

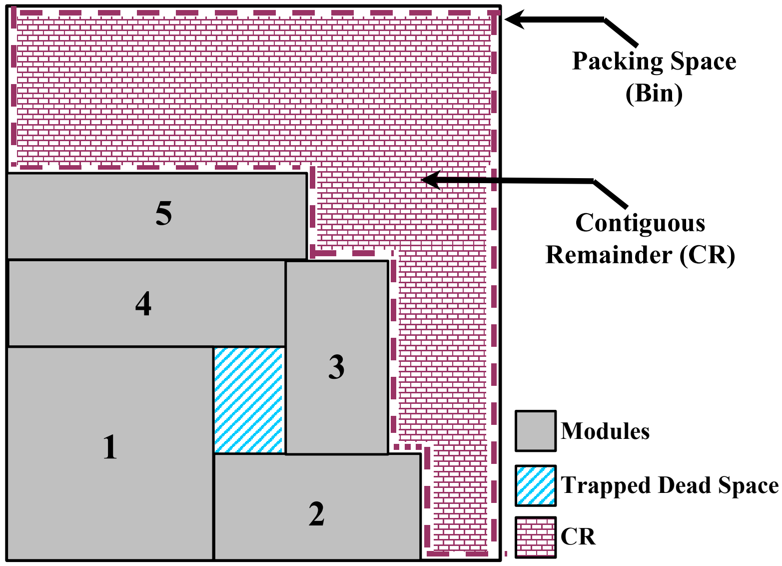


Figure 4-4: Elaboration of the concept of Contiguous Remainder

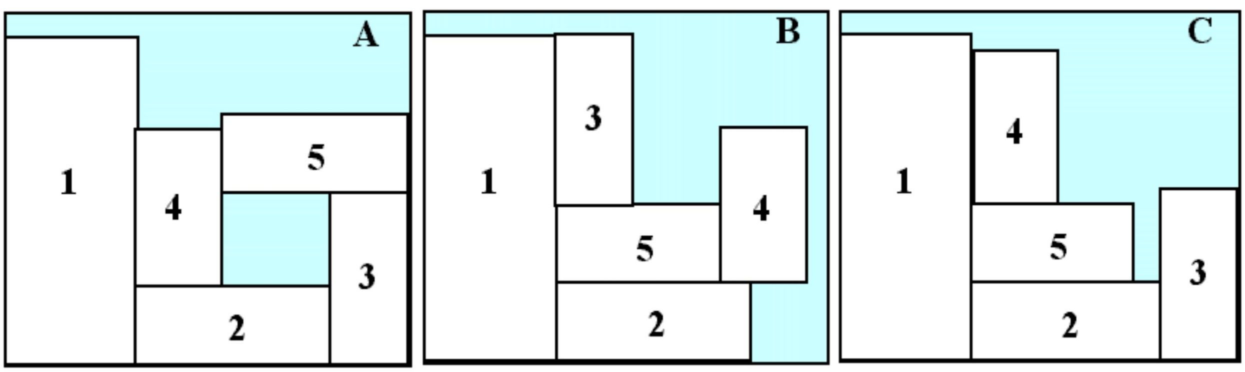


Figure 4-5: Three Layout Patterns with the Same Height

For further elaboration and comparison purposes, we consider Figure 4-5 where three layout *A, B,* and *C* have exactly the same *HT*. However, the *CR* in the layout *A* is less than that of in layout *B* and *C*. Note that *CR* does not include the dead spaces surrounded from all sides, as is the case with layout *A* where some space is surrounded by modules 2, 3, 4, and 5.

Despite widespread recommendations for the adoption of *CR* as a more appropriate measure of space utilization, we are not aware of any research literature that actually employs *CR* for comparative studies. Apparently, the conceptual and computational rigor involved in automatically quantifying the value of *CR* hampered the adoption of *CR* as more viable measure of space utilization by the scientific community.

Indeed, in many instances, *CR* proves to be a better measure of space utilization than *HT*. For instance, the layout pattern *B* in Figure 4-3 has larger *CR* than the pattern *A*. Similarly layouts *B* and *C* in Figure 4-5 have higher *CR* than layout *A*. However, *CR* alone does not capture sufficient information regarding space utilization. For instance, Figure 4-5 has layouts *B* and *C* that have identical *CR*; however, some layout experts might prefer to rate the space utilization of the layout *C* higher than that of *B* as some part of *CR* in layout *B* consists of a thin slice of space that might not be useful in a given layout design scenario. Nevertheless, our preliminary experiments have consistently shown *CR* as a better measure of space utilization than *HT*.

The Contiguous Remainder can be calculated by using the following expressions:

$$CR = Page\ Area - Total\ Module\ Area - Trappped\ Dead\ Space \qquad \text{Equation 4-17}$$

$$CR = H \times W - \sum_{i=1}^{n} w_i h_i - Dead\ Space \qquad \text{Equation 4-18}$$

A dual of *CR* is the White Space Level (*WSL*). Both *CR* and *WSL* in effect provide the same measure of the quality of the packing; however, the *WSL* is a normalized function and suits the GA and MCDM paradigm more than the *CR*. The *WSL* may also be termed as normalized *CR* or $C\hat{R}$ and is calculated as follows:

$$C\hat{R} = WSL = \frac{CR}{\sum_{i=1}^{n} w_i h_i} \times 100 \qquad \text{Equation 4-19}$$

The Trapped Dead Space is an important measure of space utilization in itself as well as in calculation of other metrics as *CR* and *WSL*. Its calculation however is not straightforward. An algorithm was developed for IDEAL since no algorithm for the exact calculation of the trapped dead space or the contiguous remainder was found in the published literature. IDEAL calculates the exact dead space by detecting the trapped spaces through a digital scanning of the packing created at any instance when a module is placed. This algorithm keeps track of all areas occupied by the placed modules and thus finds the trapped dead spaces as the areas not occupied by any module.

Module Tightness (*MT*)

In addition to *HT* and *CR*, we incorporated Module Tightness as one of our fitness determinants for space utilization. We define Module Tightness (*MT*) as the difference between Area of Enclosing Rectangle (*AER*) and total space used by modules ($\sum_i^n w_i h_i$) and is expressed as a percentage of

AER. The Module Tightness is a measure of how tightly the modules have been packed with as little trapped dead space as possible. A higher value of *MT* implies better space utilization. Here, Enclosing Rectangle (*ER*) refers to the smallest rectangle that circumscribes the whole layout. For illustration purposes, we use the Figure 4-6 that shows two topologies of a 16-module problem. In these topologies, the packing space is shown in dotted lines and the enclosing rectangle is shown in solid lines representing smallest rectangle circumscribing the whole layout. If *AER* is the area of the Enclosing Rectangle then *MT* can mathematically be expressed as follows:

$$MT = 1 - \left[\left(AER - \sum_{i=1}^{n} w_i h_i \right) / AER \right] \times 100 \qquad \text{Equation 4-20}$$

In Figure 4-6, all the space in layout *A* circumscribed by Enclosing Rectangle is utilized in packing the modules and there is no dead space inside this Enclosing Rectangle. Consequently, the *MT* for the layout *A* is 100%, representing the highest degree of space utilization. However, it is not true in the layout pattern *B* shown in Figure 4-6. As such, the layout *B* has an *MT* of 81.2%, signifying that about 81% of space inside the Enclosing Rectangle has been utilized.

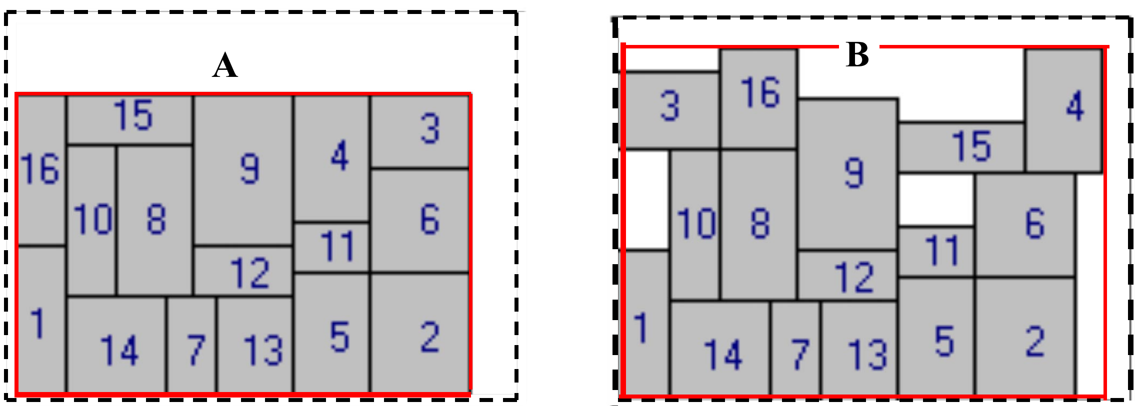


Figure 4-6: The Enclosing Rectangle – the smallest rectangle circumscribing the whole packing.

Hybrid Metrics

We want to emphasize that all these measures (namely *HT*, *CR*, *MT, IMD*) capture some aspects of space utilization. For instance, Figure 4-7 depicts two layouts with almost identical values of *HT*, *CR*, and *MT*. Nevertheless, one layout might be deemed superior to other by the decision maker. Accordingly, it should be left to the user to select an appropriate measure of space utilization, possibly as a combination of multiple measures. Consequently, our system (IDEAL) provides a very

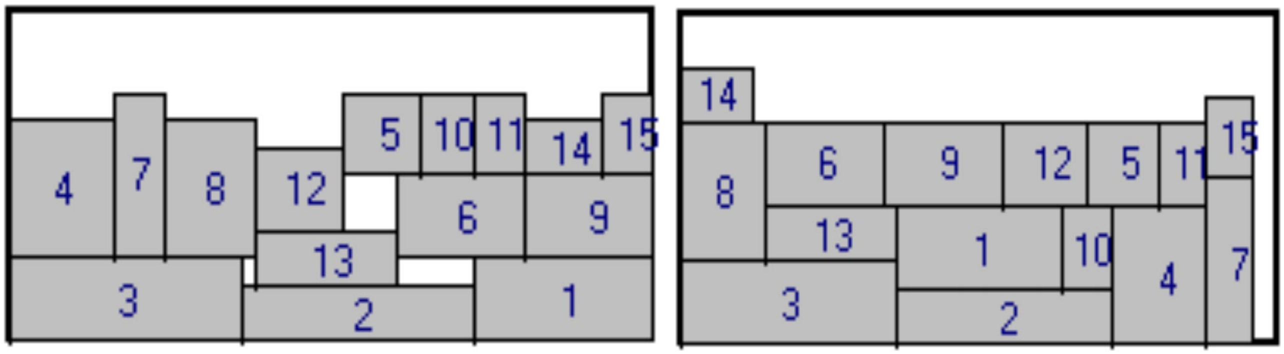


Figure 4-7: Layouts with almost Identical *HT*, *MT*, and *CR*

visible and usable way of selecting and combining these measures through the setting of appropriate weights or significance parameters. Alternatively, IDEAL may employ preference weights generated by PIA.

4.6.4 Aesthetic Appeal

Aesthetic values are subjective measures of layout quality. Such values cannot easily be defined in specific terms and usually depend on users' personal judgments. As the saying goes: beauty lies in the eye of beholder; the perceived aesthetic appeal of a given layout may be rated differently by different people. Consequently, we classify aesthetic appeal of a layout as a qualitative measure of fitness. It should be noted that GA are also known to be promising search strategy when fitness functions involve qualitative decision variables (Azadivar & Wang, 2000; Azadivar & Tompkins, 1999). However, to the best of our knowledge, no earlier study has compared computerized layout design algorithms in terms of ability to generate solutions with higher aesthetic appeal.

In addition to aforementioned quantitative fitness metrics, we employed subjective visual evaluation of layout alternatives by two experts with decades of researching and practicing experience in LD. We provided those experts with the top 10 layouts generated by each algorithm and asked them to rate those based on their personal judgment. It means that experts ranked 70 randomly ordered alternatives, as we were initially comparing seven placement algorithms. Experts had neither any knowledge about the method used for generating those alternatives nor they were under any time constraint for furnishing those ratings.

In addition, we have formulated some methods and robust measures for quantifying the aesthetic value of a layout in an automated fashion. However, the details of such qualitative fitness metrics as cohesion, distribution, and density, etc. are tackled in Section 5.3.

4.7 Decoding Algorithms

As discussed in Section 2.5.3, existing decoding algorithms lack the requisite efficiency and efficacy. Such shortcomings are more pronounced when layout evaluation criteria include such aesthetic values of the layout as cohesion, proportion, distribution, density, etc. – tackled in greater detail in Section 5.3. Such aesthetic values are considered among the key determinants of layout utility and are generally ignored in the research literature in layout design. In this section, we outline some new, efficient, efficacious, and robust placement algorithms for constructing the actual layouts with higher aesthetic contents.

These placement algorithms work with an ordering of modules obtained through some non-deterministic and evolutionary metaheuristic-based approach, which is GA in case of IDEAL. These new module placement algorithms are inspired by the fact that for any given packing space the number of modules at hand for placement is a small integer. Moreover, if we confine our placement possibilities only to the corners of 'in-place' modules then for a particular module there exist at most $O(n)$ possible locations. Accordingly, the combinatorial complexity should not pose a significant problem if some intelligent and fast pseudo-exhaustive exploration is carried out in a hierarchical manner for enhancing the space utilization and the layout quality.

It should be noted that the flexibility requirements in any ALD system calls for several efficient and effective heuristics available to the decision-maker. Consequently, we strived to develop several efficacious placement heuristics. Nevertheless, we want to make it explicit that the primary motivation in our quest for improved heuristics was our desire to generate layouts with both higher aesthetic contents and better space utilization. Consequently, we were willing to make a tradeoff in speed in order to get improved quality. Nevertheless, comparative studies have shown that proposed algorithms are more efficient in terms of overall speed of the metaheuristic-based layout optimization than other existing heuristics.

We compare these proposed placement algorithm with other popular and efficient existing placement strategies, such as the Bottom-Left or BL (Jakobs 1996), the Improved-BL or IBL (Liu & Teng, 1999), and the Bottom-Left Fill or BLF (Hopper & Turton, 2001; Chazelle, 1983) that have drawn considerable attention of the research community. Indeed, the BL property, where no module in a layout can be shifted further to the left and to the bottom of the packing space, has been a basis of numerous placement heuristics Nevertheless, the BL, the IBL, and the BLF algorithms are among the more popular deterministic placement heuristics. Evidently, the popularity of these heuristics stems from the simplicity of concept, ease of implementation, and speed of operation. Nevertheless, heuristics like BL, IBL, and BLF are not suitable for cases where the packing space is binding in both horizontal and vertical dimensions.

The decoding heuristics we have proposed also retain the BL property. However, these algorithms provide significant improvements over BL, IBL, and BLF. Our comparative evaluation demonstrates the superiority of the new placement algorithm in terms of efficiency, efficacy, robustness, and aesthetic appeal of the outcome.

4.7.1 The Minimization of Enclosing Rectangle Area (MERA) Algorithm

The first placement algorithm is Minimization of Enclosing Rectangle Area (MERA). Its title is inspired by the underlying notion where a reduction of the rectangular area of the packing pattern (*AER)* is sought during all placement decisions with a bias term favoring lower placements. The optimization part in the placement strategy is not an extensive or expensive optimization but a sort of a heuristic refinement – a pseudo-exhaustive search. Such a hierarchical optimization scheme facilitates improvement in space utilization as well as quality of layouts. This pseudo-exhaustive method is outlined here.

In the pseudo-code for MERA, index *A* corresponds to four corners of an in-place module (M_L) and index *B* corresponds to four corners of an in-coming module (M_K). The Step 2 proceeds by investigating the placement prospects for all four corners of an in-coming module at all four corners of all in-place modules. The second term in the composite *newOBJ* is meant to bias placement to the bottom-left position in the layout, which is a general packing preference in various placement heuristics or LD contexts such as bin-packing.

The computational cost of BL-algorithm is $O(n^2)$. It follows from the fact that every in-coming module can be shifted a maximum of i times as every shift is restricted by one of the $i-1$ previously placed modules or by the boundaries of the bin. However, in case of MERA, each in-coming module can be placed at a maximum of $16(i-1)$ corner points (a very weak upper bound) where $i-1$ modules are previously in place. As such, theoretically the MERA algorithm also has the same $O(n^2)$ cost as for BL and IBL (Jakobs, 1996; Liu & Teng, 1999). Moreover, an increase in the computational complexity would be considered quite rational and acceptable if significant improvements in terms of both quantitative and qualitative fitness metrics are realized, as demonstrated by comparative analyses in Section 4.8.

Let:

Blocks = No. of Modules at hand for Placement

Nplaced = No. of Modules Already Placed

ER = Enclosing Rectangle

AER = Area of Enclosing Rectangle

y_i = y-coordinate of the bottom-left corner of i^{th} module.

$newOBJ = AER + (y_i + \text{width of } ER)/2$

1) Place module 1 at the bottom-left corner of the packing space

2) Set *OBJ* to a big value

3) FOR $K = 2$ to *Blocks*

    FOR $L = 1$ to *NPlaced*

      FOR $A = 1$ to 4

        FOR $B = 1$ to 4

            Place corner B of M_K on corner A of M_L

            Calculate the *newOBJ*

            IF *newOBJ* is less than *OBJ* THEN

                OBJ = *newOBJ*

                Save placement of module M_K

            ENDIF

    END *B*

   END *A*

END *L*

END *K*

4) Stop if no room for more modules.

4.7.2 Minimization of Enclosing Rectangle Area under Gravitational Attraction

The MERA module placement heuristic can easily be adapted for various commonly encountered situations through simple modifications of the objective function. One such application is the optimization of some measure of inter-module interactions. For instance, minimization of total inter-module distance in a facility layout involving material handlings or a VLSI layout involving macrocell interconnects. Such scenarios mimic gravitational attraction among modules. As such, we call it the Minimization of Enclosing Rectangle Area under Gravitational Attraction (MERAG). Following the familiar reasoning approach, we can establish that the theoretical computational complexity of MERAG is also $O(n^2)$.

As mentioned, the MERAG decoding or module placement heuristic works in a similar way to MERA but differs in the minimization objective employed for making placement decisions. Here again, a composite minimization objective is employed for placement stipulations that include the *Area of the Enclosing Rectangle* plus the *Weighted Sum of All Module Distances from a Single Point* as depicted in following Equation:

$$\sum_i \sum_j \alpha_{i,j}\, d_{i,j} + weight \times AER \qquad \text{Equation 4-21}$$

This is analogous to gravitational pull towards a center or focal point with a force proportional to the distance between the focal point and modules. Here, *weight* is a user-defined value to permit the user to change the weightage given to *AER* in these placement decisions (the default value in IDEAL is unity). Whereas, $\alpha_{i,j}$ is the connectivity or interaction between the Module M_i and the Module M_j.

The values of $\alpha_{i,j}$ could also be manipulated by the user to control the module placement decisions with respect to the focal point. The ability to control $\alpha_{i,j}$ in the range of [-1, 1] provides another level of user control on the inter-module interaction resulting from relative positions of pairs of modules (i.e. mutual distances between modules). For instance, a larger value of $\alpha_{i,j}$ would tend to bring the Module M_i and the Module M_j closer to each other. Similarly, a negative value of $\alpha_{i,j}$ would tend to keep the Module M_i and the Module M_j distant from each other. However, as a default, we have kept all values of $\alpha_{i,j} = 0$ for all pairs of modules except when $i=0$, representing a dummy module of zero dimensions and zero utility to be served as a focal point or center of gravity in the packing space, where $\alpha_{0,j} = 1$. As before, the user has control over deciding which modules should be closer to the focal point by controlling $\alpha_{0,j}$. Nevertheless, all our experimental results reported in this Chapter are based on default values of all $\alpha_{i,j} = 0$, without any loss of generalization.

It should be noted that there is no bias term added to the minimization objective function for making placements at the bottom-most and left-most positions preferable. It is because we also want to breakaway from the traditional goal of minimizing the packing height, as the case with existing packing heuristics. We reiterate that our motivation was to procure superior solutions not only in terms of space utilization but also in terms of aesthetic contents. Furthermore, as discussed earlier, *HT* is not always a good choice for measuring the space utilization. For instance, *CR* is a more viable fitness measure of space utilization, as indicated by several researchers. As results in Section 4.8 show, MERAG without the bottom-left bias provides solutions with higher aesthetic contents and *CR*.

Let:
Blocks = No. of Modules at hand for Placement
NPlaced = No. of Modules Already Placed
newOBJ = Sum of Inter-module Distances
AER = Area of Enclosing Rectangle

1) Place module 1 at the bottom-left corner of the page
2) Set *OBJ* to a big value
3) FOR K = 2 to *Blocks*
    FOR L = 1 to *NPlaced*
        FOR A = 1 to 4
            FOR B = 1 to 4
                Place corner B of M_K on corner A of M_J
                Calculate the *newOBJ* = $\sum_{i=1}^{i=K} \sum_{j=1}^{NPlaced} \alpha_{i,j}\, d_{i,j} + weight \times AER$
                IF *newOBJ* is less than *OBJ* THEN
                    OBJ = *newOBJ*
                    Save placement of module M_K
                ENDIF
            END B
        END A
    END L
END K
4) Stop if no room for more modules.

4.7.3 Minimization of Enclosing Rectangle Area under Magnetic Attraction

Just like MERAG, we can readily adapt MERA to conform to a situation where every module potentially interacts with every other module in the layout. Such a situation mimics magnetic attraction of modules to specific focal points. As such, we call it the Minimization of Enclosing Rectangle Area under Magnetic Attraction (MERAM) strategy. This algorithm also works in a similar way to MERA but differs in the minimization objective. In MERA, the objective is to minimize the area of the Enclosing Rectangle; whereas, in MERAM a composite minimization objective is used that includes the *Area of the Enclosing Rectangle* plus the *Sum of Inter-module*

Distances. This is as if all the modules apply force of attraction on each other with force proportional to the distance between a pair of modules. The composite objective is shown in the following Equation:

$$\sum_{i=1}^{n-1} \sum_{j=1}^{n} \alpha_{i,j} d_{i,j} + weight \times AER \qquad \text{Equation 4-22}$$

Here *weight* is a user-defined value and $\alpha_{i,j}$ is the connectivity or interaction between the Module M_i and the Module M_j and is assumed unity for all pairs (i.e. $\alpha_{i,j} = 1$ for all pairs of modules); however, $\alpha_{i,j}$ could be manipulated by the user to control the placements of modules with respect to each other. Consequently, this technique would allow users to incorporate preferences regarding relative positions of modules. For instance, if a pair of modules is required to be closer to each other then positive $\alpha_{i,j}$ ($\alpha_{i,j} > 0$) could be used. Alternatively, a negative $\alpha_{i,j}$ ($\alpha_{i,j} < 0$) would imply that proximity between a particular pair of modules is undesirable. Similar to BL and MERA, we can establish that the theoretical computational complexity of MERAM is $O(n^2)$.

4.8 Comparative Evaluation of Decoding Algorithms

In order to test and validate the efficiency, efficacy, and robustness of our placement algorithms in producing layout of higher aesthetic contents, we employed both automated capturing of quantitative measures as well as visual evaluations by experts in layout design. We employed some randomly generated and some benchmark problems from the existing research literature for these comparisons.

A computer program was written in Visual BASIC to implement the BL, IBL, BLF, MERA, MERAG, and MERAM as well as the GA based optimization component including various qualitative and quantitative fitness evaluation functions. In comparative studies, all aspects are kept identical except the module placement strategy. Consequently, the reported results do not account for any interactions between the module placement strategy and some GA parameters, which is in itself a good research direction. Furthermore, our implementation of IBL and BLF does not involve rotation of modules, as our intended work domain that involves only oriented modules. Nevertheless, IDEAL supports the rotation of modules, if needed. The computer program is used for comparative analyses on Intel Xeon 3.06GHz processor and 256MB of RAM under Windows XP.

Our initial quantitative analyses are centered on packing height, module tightness, contiguous remainder and inter-module distances. Furthermore, three facility layout design researchers and practitioners were asked to provide subjective rating of some layout alternatives in terms of symmetry. These experts have decades long experience in teaching, researching, and practicing in layout design applications. These experts had no knowledge of the algorithm/method used for generating these alternatives. Furthermore, they did not have any indication of fitness metrics/values used by us. In addition, these experts were under no time constraint for furnishing their ratings. All three experts have decades long experience in teaching, researching, and practicing in layout design applications. These ratings were on a scale of 1-10 with a higher score representing higher aesthetic value perceived by the expert. We want to emphasize that a layout quality rating of 10 represents a highly symmetric layout configuration, which usually cannot be achieved for problems consisting of randomly generated unequal modules or when modules dimensions have high variability. Consequently, we found that a Layout Quality rating of around 5 implies that the layout alternative is quite superior, in terms of symmetry, for the given problem instance. Furthermore, experts were given only the top 10 layout alternatives, in terms of *CR*, for subjective rating. This approach meant that experts remained keenly interested in our experiments. Rating a large number of layout alternatives might have diminished their interest and, subsequently, the reliability of their subjective estimates. Nevertheless, such a practice confounds the measure of layout quality with the *CR*. In short, a rating close to 5 on a scale of 1-10 should not been seen as inferior even if modules dimensions permit a higher degree of symmetry.

We used several benchmark problems consisting of 25-, 49-, 50-, 97- and 100-modules from the literature for our comparative studies – shown in Appendix C (with two instances each for 25- and 50-module cases). We initially employed a Random Search approach for our comparative studies by generating 100 random sequences of modules. As already mentioned, Random Search and Naive Evolution are among the most effective search strategies, though not at par with GA or SA, for layout design problems.

The relative performance of the BL, IBL, BLF, MERA, MERAG, and MERAM placement strategies for 100 random sequences of each benchmark problem instance is depicted in Table 4-3 to Table 4-9. In these tables, the first column shows the fitness metric against which the performance of various algorithms is measured as well as the optimal or benchmark values of those performance metrics. Since it is not easy to verifiably come up with an optimal value of IMD, we have used the IMD of a layout alternative that has optimal *HT*, *MT*, and *CR* as a reference value. The second column shows placement heuristics that are being compared for the given fitness metric. The third column shows the number of instances a particular algorithm performed better than others in 100 Random Search (RS) iterations. Columns 4 to 7 represent the best outcome, the worst outcome, the mean fitness, and the standard deviation of fitness values, respectively, for the corresponding algorithm. For ease of comprehension, the best outcome in each column is shown in boldface.

Results show that MERA and its adaptations, MERAG and MERAM, outperform the existing algorithms by wide margins. The proposed algorithms generate superior outcomes in terms of the packing Height or *HT* (the lower the better), the Module Tightness or *MT* (the higher the better), the Contiguous Remainder *CR* (the higher the better), the Inter-Module Distances or *IMD* (the lower the better) and the layout Quality Rating *QR* (the higher the better). The performance gains are more pronounced for larger problems. This superior performance can be shown as statistically significant using means and standard deviations.

More specifically, results indicate that MERA is more suitable for generating superior layout alternatives in terms of *HT*, *MT*, and layout quality *QR*. Notably, shortcomings of BL, IBL, and BLF discussed in Chapter 2 were so overwhelming that often we were required to increase the height of the packing space for providing these algorithms sufficient room to generate feasible solutions.

In addition to MERA, its adaptations MERAG and MERAM generally fared significantly better than BL, IBL, and BLF in terms of *CR* and *IMD*. However, in comparison to MERA, MERAG and MERAM seem to be more appropriate for generating superior layout alternatives with *CR* and *IMD* as fitness metrics. Furthermore, MERAG and MERAM also provided layout alternatives that were generally rated higher in *QR* by experts.

To elaborate more on these results, we consider the 100-module problem (A100) for which performance comparisons are presented in Table 4-9. These performance comparisons are depicted in Figure 4-8 to Figure 4-11 as a percentage distance from the optimal or benchmark values for visual comparison purposes. In Figure 4-8 to Figure 4-11, the first picture from top represents the number of times each algorithm outperformed all other algorithms for the given fitness measure in 100 RS iterations. The second picture represents the best outcome obtained from each algorithm. The third picture shows the mean performance of each algorithm. Likewise, the fourth picture depicts the worst outcome from each algorithm.

In terms of *HT*, the optimal value was 100 units and the outcome closest to the optimal value was generated by MERA. Furthermore, MERA generated superior outcomes more frequently than any other algorithm. Although MERAG and MERAM generated much superior outcomes in comparison to BL or IBL, these adaptations of MERA do not seem to outperform BLF in terms of *HT*. These observations may be visually appraised in Figure 4-8.

Indeed, MERAG and MERAM are not geared towards generating solutions with lower *HT*, as there is no bias term used for giving priority to a lower placement position in their composite fitness objectives. Nevertheless, a simple adaptation of the respective composite fitness objective in MERAG and MERAM, by adding a bias towards a lower placement position, may provide layout solutions with improved *HT*. Furthermore, our preliminary studies indicate that even reducing the *weight* of the *AER* in the composite fitness objective results in layout alternatives with significantly reduced *HT*. Nevertheless, we have not tested this notion in a scientific manner. Similar observations can be made with regard to *MT* and can be visually appraised using Figure 4-9. However, performance of MERAG and MERAM with regard to *MT* may also be improved by adding a bias term or by changing the *weight* assigned to *AER* in their respective composite fitness function.

In terms of *CR*, MERA and both of its adaptations perform significantly better than the existing algorithms, as can be seen from Table 4-9 as well as Figure 4-10. In particular, MERAM frequently provided solutions better than other algorithms. Nevertheless, MERA has also provided very good solutions as well as the best outcome. Moreover, MERAG also provided solutions that were frequently superior to those provided by the existing algorithms.

Similarly, in terms of *IMD*, the proposed algorithms outperform the existing ones. In particular, MERAG provided superior solutions very frequently. We used the *IMD* of a layout alternative that has optimal *HT*, *MT*, and *CR* as a reference value, which is 563,000 units. Indeed, it constitutes a very realistic benchmark value of *IMD*, which is not necessarily optimal. The performance of all algorithms relative to this benchmark value may be visually appraised using Figure 4-11. It is interesting to observe that the proposed algorithms exceed this reference by several percentage points.

Above all, the proposed algorithms provided solutions that received relatively very high subjective *QR* rating from experts. Indeed, procurement of solutions with higher aesthetic value was the motivation behind developing these new algorithms. However, as observed, the proposed algorithms also provide superior solutions against other fitness measures.

Fitness Objective	Algorithm	Wins	Best	Worst	Mean	Std. Dev.
HT **(Optimal = 15)** **The Lower the Better**	**BL**	5	**17**	22	19.56	1.12
	IBL	8	**17**	22	18.96	1.10
	BLF	29	**17**	**21**	18.48	0.92
	MERA	**58**	**17**	**21**	**18.42**	0.91
	MERAG	0	23	30	26.62	1.77
	MERAM	0	23	30	26.65	1.60
MT **(Optimal = 100%)** **The Higher the Better**	**BL**	5	88.24	68.18	77.05	4.39
	IBL	7	88.24	68.18	79.43	4.53
	BLF	28	88.24	64.10	80.76	3.82
	MERA	**46**	**90.50**	**71.43**	**81.91**	4.43
	MERAG	4	88.89	53.05	71.41	6.74
	MERAM	10	88.89	63.03	78.29	5.53
CR **(Optimal = 600)** **The Higher the Better**	**BL**	0	577	477	525.70	22.74
	IBL	0	579	459	538.45	25.32
	BLF	2	587	501	553.09	17.88
	MERA	14	597	506	562.12	16.70
	MERAG	40	595	**528**	570.89	16.42
	MERAM	**44**	**598**	522	**575.08**	13.79
IMD **(Reference = 9558)** **The Lower the Better**	**BL**	0	9010.55	11008.92	10008.77	443.21
	IBL	0	8733.58	10545.81	9702.29	444.79
	BLF	0	8665.30	10479.1	9760.64	383.11
	MERA	0	8518.17	10580.62	9470.92	426.84
	MERAG	**54**	**6649.22**	9118.25	**7629.36**	417.29
	MERAM	46	7009.07	**8337.42**	7668.98	297.82
QR **(Scale: 1-10)** **The Higher the Better**	**BL**	0	3.0	1.75	2.55	0.44
	IBL	0	5.25	2.5	3.18	0.89
	BLF	0	3.75	2.5	3.23	0.45
	MERA	**5**	**5.5**	**2.75**	**3.98**	0.81
	MERAG	2	4.75	1.75	3.38	1.02
	MERAM	3	4.75	2.25	3.80	0.76

Table 4-3: Comparison of Decoding Heuristics for 100 random sequences of Problem **H25**

Fitness Objective	Algorithm	Wins	Best	Worst	Mean	Std. Dev.
HT (Optimal = 15) The Lower the Better	**BL**	3	18	28	21.92	1.94
	IBL	7	18	26	21.40	2.02
	BLF	25	**17**	**24**	19.55	1.38
	MERA	**65**	**17**	**24**	**19.31**	1.31
	MERAG	0	24	30	27.26	1.62
	MERAM	0	23	30	27.36	1.59
MT (Optimal = 100%) The Higher the Better	**BL**	1	83.33	53.57	69.23	5.99
	IBL	5	83.33	57.69	70.93	6.56
	BLF	12	90.50	62.50	78.26	5.33
	MERA	**40**	90.52	62.50	**78.39**	5.04
	MERAG	13	85.71	58.82	74.79	5.74
	MERAM	29	**90.91**	**65.93**	77.89	4.65
CR (Optimal = 600) The Higher the Better	**BL**	0	573	454	529.78	23.19
	IBL	6	584	470	547.13	21.14
	BLF	7	590	509	563.45	14.52
	MERA	**38**	593	526	**569.84**	11.86
	MERAG	31	**598**	530	567.03	12.97
	MERAM	18	587	**544**	568.08	9.11
IMD (Reference =9048) The Lower the Better	**BL**	0	8300.1	11678.4	10102.7	619.37
	IBL	0	8585.9	9604.0	9604.0	465.35
	BLF	0	8254.69	10411.4	9496.0	460.93
	MERA	0	8369.7	10255.7	9285.7	405.21
	MERAG	**71**	**7141.3**	8760.1	**7885.1**	332.31
	MERAM	29	7290.3	**8715.1**	8045.0	309.8
QR (Scale: 1-10) The Higher the Better	**BL**	0	2.75	1.25	1.88	0.54
	IBL	0	3.5	2.0	2.80	0.44
	BLF	0	4.25	2.25	3.28	0.57
	MERA	4	5.0	3.0	4.13	0.60
	MERAG	1	5.5	**3.25**	3.83	0.76
	MERAM	**5**	**6.0**	3.0	**4.33**	0.99

Table 4-4: Comparison of Decoding Heuristics for 100 random sequences of Problem **J25**

Fitness Objective	Algorithm	Wins	Best	Worst	Mean	Std. Dev.
HT **(Optimal = 60)** **The Lower the Better**	**BL**	1	78	116	90.8	7.53
	IBL	0	73	106	88.2	6.6
	BLF	20	70	94	77.0	4.43
	MERA	**43**	**68**	**93**	**76.91**	5.13
	MERAG	7	70	100	79.8	4.8
	MERAM	29	70	94	77.3	4.22
MT **(Optimal = 100%)** **The Higher the Better**	**BL**	1	77.1	51.9	66.8	5.43
	IBL	0	83.8	56.8	68.6	5.14
	BLF	15	85.9	64.0	78.5	4.31
	MERA	**40**	**90.1**	**65.8**	79.8	5.08
	MERAG	9	87.4	64.5	77.5	4.36
	MERAM	35	87.4	65.1	**80.2**	4.1
CR **(Optimal = 4800)** **The Higher the Better**	**BL**	0	4313	3073	3841.3	247.81
	IBL	0	4456	3611	4085.9	175.44
	BLF	6	4591	4171	4403.03	95.88
	MERA	29	4653	4281	4522.1	74.92
	MERAG	21	4658	4254	4506.3	80.35
	MERAM	**44**	**4685**	**4348**	**4545.6**	63.85
IMD **(Reference = 73600)** **The Lower the Better**	**BL**	0	79208.6	107309.3	91311.7	5565.9
	IBL	0	75610.0	96652.0	85878.7	3838.6
	BLF	3	71581.5	88750.4	80338.8	3425.3
	MERA	5	69330.9	85086.8	78213.9	2978.7
	MERAG	**70**	**67883.1**	**80514.2**	**73741.0**	2797.5
	MERAM	22	67979.2	83308.6	75893.7	2836.4
QR **(Scale: 1-10)** **The Higher the Better**	**BL**	0	3.25	1.25	2.1	0.56
	IBL	0	3.5	2.25	2.88	0.4
	BLF	0	4.75	1.5	3.58	1.07
	MERA	3	6.5	**3.75**	5.35	0.75
	MERAG	2	6.75	**3.75**	4.85	1.12
	MERAM	**5**	**7.0**	3.5	**5.48**	1.13

Table 4-5: Comparison of Decoding Heuristics for 100 random sequences of Problem **H49**

Fitness Objective	Algorithm	Wins	Best	Worst	Mean	Std. Dev.
HT **(Optimal = 55)** **The Lower the Better**	**BL**	1	66	99	79.0	5.66
	IBL	1	68	91	77.21	4.73
	BLF	15	65	79	69.2	3.12
	MERA	**81**	**64**	**78**	**68.7**	3.17
	MERAG	0	74	117	84.6	5.54
	MERAM	0	76	93	83.9	3.60
MT **(Optimal = 100%)** **The Higher the Better**	**BL**	1	83.49	56.22	70.19	4.88
	IBL	1	81.03	60.55	71.72	4.27
	BLF	11	85.16	70.45	80.60	3.52
	MERA	**67**	**86.73**	**71.36**	**81.04**	3.57
	MERAG	4	82.98	60.55	74.07	4.56
	MERAM	16	82.96	66.93	76.33	3.19
CR **(Optimal = 2500)** **The Higher the Better**	**BL**	0	1848	365	1404.6	248.84
	IBL	0	2025	976	1600.4	191.66
	BLF	2	2101	1738	1805.7	95.79
	MERA	29	2251	1835	2051.1	88.89
	MERAG	19	2256	1386	1992.1	157.62
	MERAM	**50**	**2287**	**1756**	**2084.9**	109.54
IMD **(Reference =103000)** **The Lower the Better**	**BL**	0	106743.9	132059.6	114929.0	4289.85
	IBL	0	102312.3	118215.9	110207.2	3149.32
	BLF	0	99970.8	115237.4	108215.1	2588.71
	MERA	0	99220.2	111803.2	106558.0	2307.54
	MERAG	**84**	**91988.2**	**103857.1**	**96853.3**	2338.76
	MERAM	16	93559.0	105806.7	100271.7	2507.35
QR **(Scale: 1-10)** **The Higher the Better**	**BL**	0	3.0	1.5	2.1	0.54
	IBL	0	3.0	2.0	2.25	0.35
	BLF	0	5.25	2.5	4.08	0.97
	MERA	**4**	6.0	3.25	4.70	1.03
	MERAG	**4**	**6.5**	**3.5**	**4.85**	1.20
	MERAM	2	6.0	3.25	4.68	0.99

Table 4-6: Comparison of Decoding Heuristics for 100 random sequences of Problem **A50**

Fitness Objective	Algorithm	Wins	Best	Worst	Mean	Std. Dev.
HT **(Optimal = 15)** **The Lower the Better**	**BL**	3	19	26	21.54	1.96
	IBL	5	18	30	22.57	2.47
	BLF	5	**17**	23	19.89	1.00
	MERA	**87**	**17**	**21**	**19.24**	0.95
	MERAG	0	25	40	28.83	2.56
	MERAM	0	26	31	27.97	1.22
MT **(Optimal = 100%)** **The Higher the Better**	**BL**	1	80.53	58.85	71.48	5.35
	IBL	3	85.00	51.00	68.59	7.11
	BLF	3	89.21	69.33	78.58	4.39
	MERA	**68**	**92.31**	**74.73**	**81.80**	4.02
	MERAG	9	84.07	58.85	76.35	4.67
	MERAM	16	87.18	70.35	78.29	3.16
CR **(Optimal = 600)** **The Higher the Better**	**BL**	1	573.0	463.0	515.4	18.77
	IBL	1	568.0	483.0	536.6	15.33
	BLF	1	574.0	513.0	550.5	7.91
	MERA	**64**	577.0	**544.0**	**562.7**	6.69
	MERAG	16	**583.0**	526.0	554.7	11.12
	MERAM	17	580.0	530.0	556.9	8.76
IMD **(Reference = 36670)** **The Lower the Better**	**BL**	0	35217.2	45429.7	39956.7	1634.18
	IBL	0	35196.6	42107.4	38272.3	1510.05
	BLF	0	35485.4	42457.8	39547.4	1081.42
	MERA	0	34644.4	40339.8	37473.7	1067.66
	MERAG	**87**	**30704.1**	**34141.4**	**32114.4**	743.72
	MERAM	13	30932.3	35528.8	33165.5	884.16
QR **(Scale: 1-10)** **The Higher the Better**	**BL**	0	3.25	1.0	2.18	0.83
	IBL	0	3.25	2.0	2.75	0.39
	BLF	0	4.0	2.5	2.83	1.09
	MERA	**5**	6.0	**3.25**	**5.0**	0.82
	MERAG	4	**6.5**	**3.25**	4.85	1.37
	MERAM	1	5.25	3.0	4.33	0.66

Table 4-7: Comparison of Decoding Heuristics for 100 random sequences of Problem **J50**

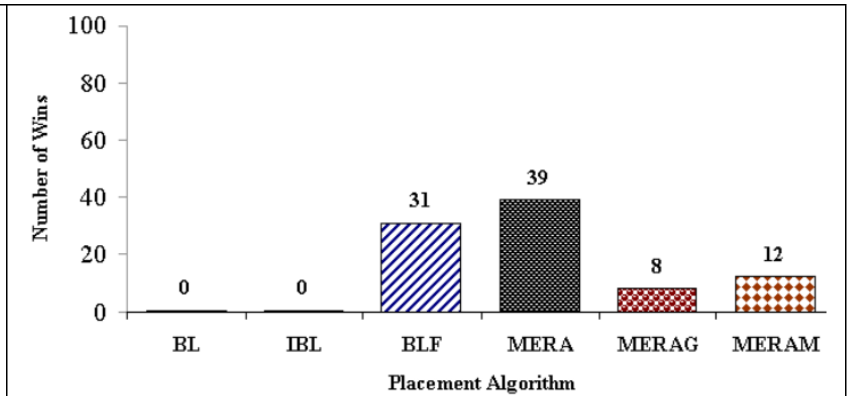
Fitness Objective	Algorithm	Wins	Best	Worst	Mean	Std. Dev.
HT **(Optimal = 120)** **The Lower the Better**	**BL**	0	170	256	197.49	16.92
	IBL	0	152	240	180.38	13.66
	BLF	13	136	186	154.92	9.48
	MERA	34	136	**176**	152.20	9.60
	MERAG	14	135	182	154.96	10.56
	MERAM	**39**	**134**	182	**151.28**	9.68
MT **(Optimal = 100%)** **The Higher the Better**	**BL**	0	70.64	46.91	61.22	4.93
	IBL	0	79.01	50.04	66.94	4.86
	BLF	9	88.30	64.56	77.81	4.68
	MERA	31	89.42	**69.10**	80.21	4.96
	MERAG	18	89.47	66.82	79.0	5.27
	MERAM	**42**	**90.08**	66.82	**80.83**	5.09
CR **(Optimal = 7200)** **The Higher the Better**	**BL**	0	5023.0	763.0	3398.0	825.08
	IBL	0	5897.0	2111.0	4643.5	606.82
	BLF	1	6856.9	5321.8	6237.1	276.78
	MERA	22	6868.0	5017.0	6460.9	253.30
	MERAG	26	6795.0	**5881.0**	6497.3	204.31
	MERAM	**51**	**6928.0**	5762.0	**6596.0**	200.94
IMD **(Reference = 502700)** **The Lower the Better**	**BL**	0	535922.4	850390.9	675013.1	55187.0
	IBL	0	481388.8	686003.3	606174.2	40633.8
	BLF	0	477175.4	582486.8	524828.1	19899.6
	MERA	0	456947.5	546825.2	502645.6	18813.6
	MERAG	**89**	**397695.3**	**515612.9**	**455439.8**	21629.4
	MERAM	11	433334.5	525795.4	479824.5	20728.5
QR **(Scale: 1-10)** **The Higher the Better**	**BL**	0	2.25	1.25	1.75	0.35
	IBL	0	2.75	1.75	2.08	0.37
	BLF	0	4.25	2.25	3.33	0.71
	MERA	3	5.5	2.75	4.23	0.74
	MERAG	**4**	**5.75**	**3.5**	**4.5**	0.76
	MERAM	3	**5.75**	3.0	**4.5**	0.97

Table 4-8: Comparison of Decoding Heuristics for 100 random sequences of Problem **H97**

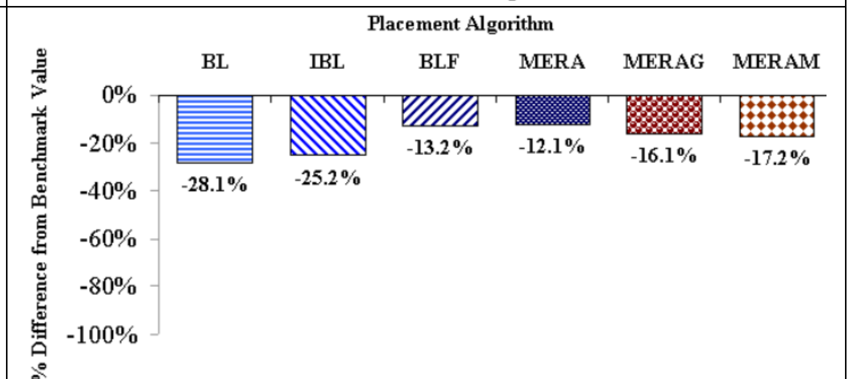
Fitness Objective	Algorithm	Wins	Best	Worst	Mean	Std. Dev.
HT **(Optimal = 100)** **The Lower the Better**	**BL**	0	128.1	160.2	142.45	6.97
	IBL	0	125.2	158.3	139.78	6.50
	BLF	31	113.2	131.8	121.72	3.49
	MERA	**39**	**112.1**	**128.3**	**118.88**	3.23
	MERAG	8	116.1	134.1	124.09	3.60
	MERAM	12	117.4	129.8	123.14	2.68
MT **(Optimal = 100%)** **The Higher the Better**	**BL**	0	77.52	62.05	69.99	3.39
	IBL	0	79.55	62.73	71.30	3.27
	BLF	10	87.81	75.74	81.76	2.27
	MERA	**71**	**88.76**	**77.37**	**83.81**	2.23
	MERAG	7	85.61	74.95	80.54	2.26
	MERAM	12	85.44	76.66	81.13	1.81
CR **(Optimal = 5000)** **The Higher the Better**	**BL**	0	3112.4	705.2	2069.2	2069.2
	IBL	0	3208.0	1045.1	2415.0	2415.0
	BLF	10	4163.72	3238.07	3764.85	200.34
	MERA	19	**4546.0**	3572.2	4042.1	175.79
	MERAG	6	4315.7	3252.8	3994.7	198.62
	MERAM	**65**	4521.8	**3866.9**	**4196.9**	119.49
IMD **(Reference = 563000)** **The Lower the Better**	**BL**	0	582589.4	696861.1	632092.7	22983.7
	IBL	0	566760.9	656799.0	606410.5	18531.7
	BLF	0	533895.3	591047.9	565016.8	11682.3
	MERA	6	518156.9	575072.4	551050.4	10414.5
	MERAG	**79**	**499098.1**	**559710.5**	**533375.1**	12315.8
	MERAM	15	511928.0	571375.5	543029.4	12414.3
QR **(Scale: 1-10)** **The Higher the Better**	**BL**	0	2.25	1.25	1.75	0.39
	IBL	0	3.25	1.75	2.43	0.50
	BLF	0	3.5	2.5	2.95	0.35
	MERA	3	5.0	3.0	4.05	0.71
	MERAG	**6**	**5.75**	**3.75**	**4.65**	0.61
	MERAM	1	4.50	2.75	3.85	0.52

Table 4-9: Comparison of Decoding Heuristics for 100 random sequences of Problem **A100**

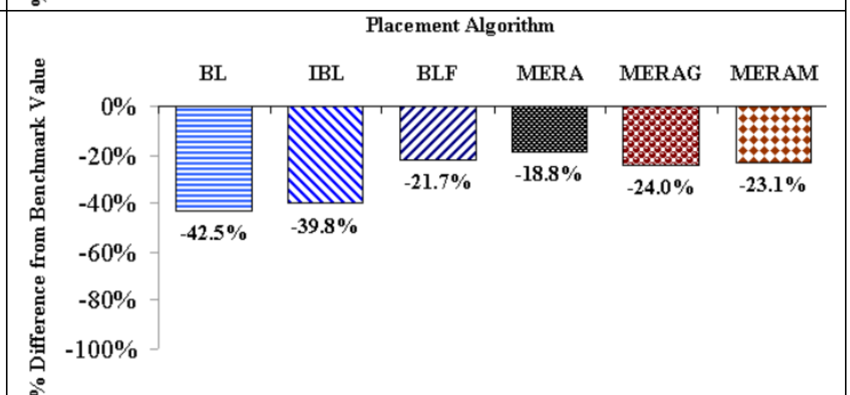
Number of Wins



Best outcome
(% diff. from optimal value)



Mean outcome
(% diff. from optimal value)



Worst outcome
(% diff. from optimal value)

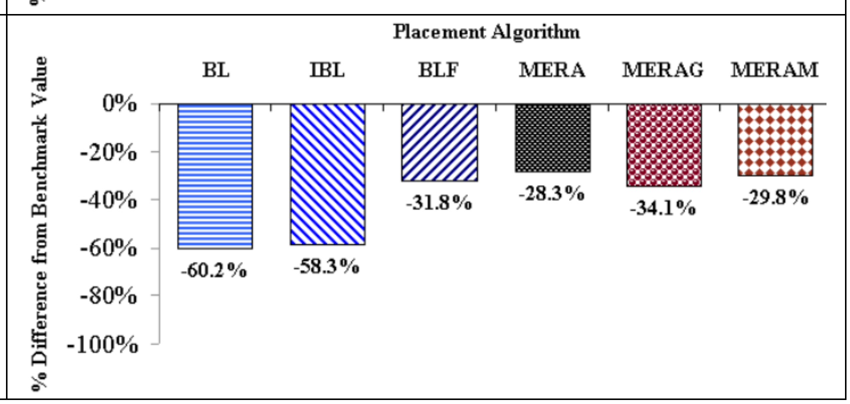
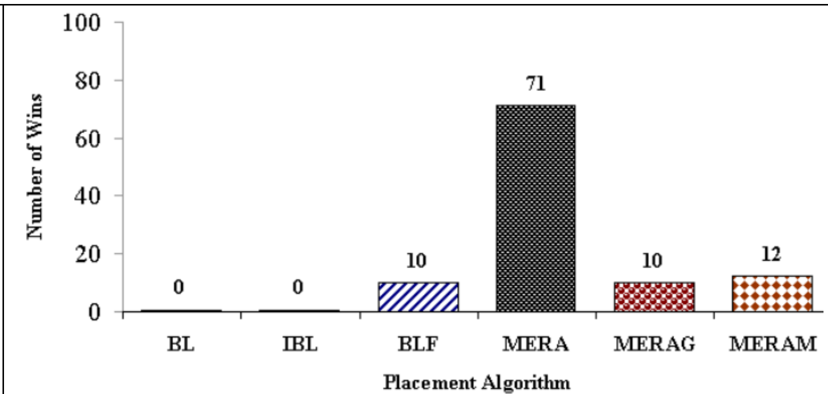
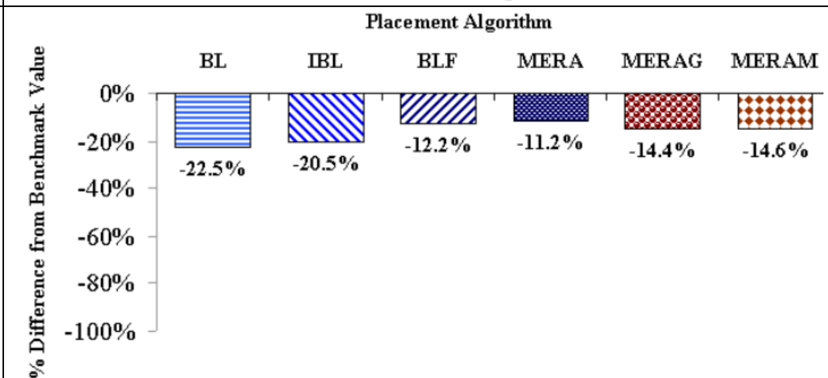


Figure 4-8: Performance comparison of algorithms w.r.t. *HT* for 100-module problem (A100)

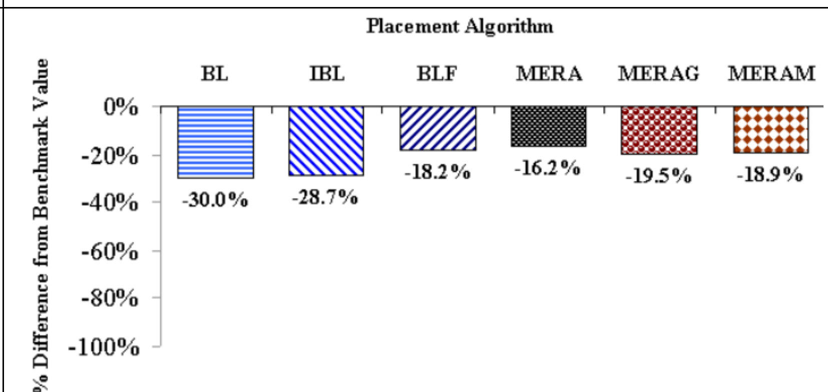
Number of Wins



Best outcome
(% diff. from optimal value)



Mean outcome
(% diff. from optimal value)



Worst outcome
(% diff. from optimal value)

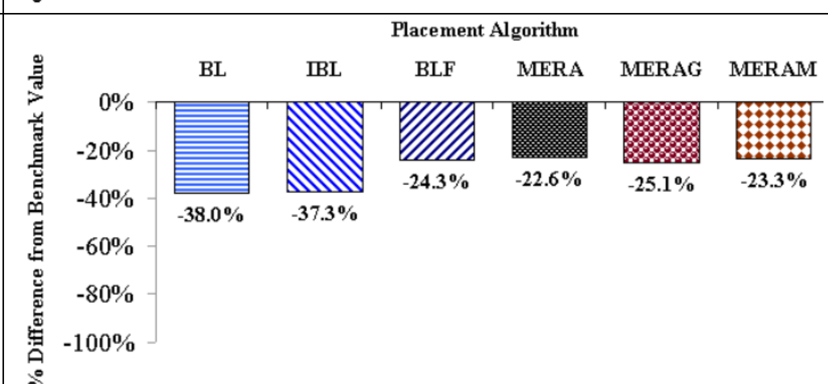
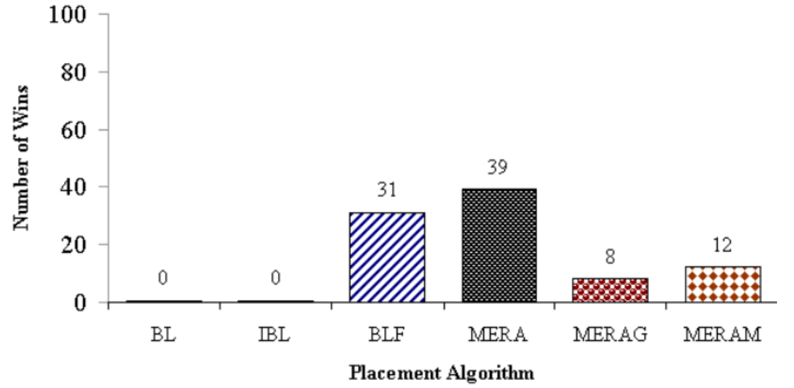
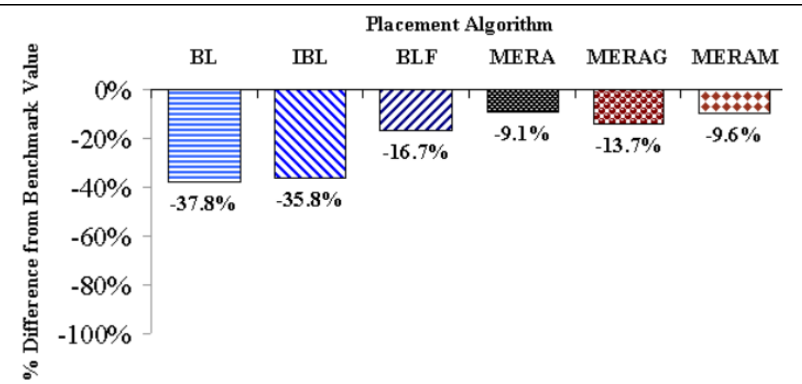


Figure 4-9: Performance comparison of algorithms w.r.t. *MT* for 100-module problem (A100)

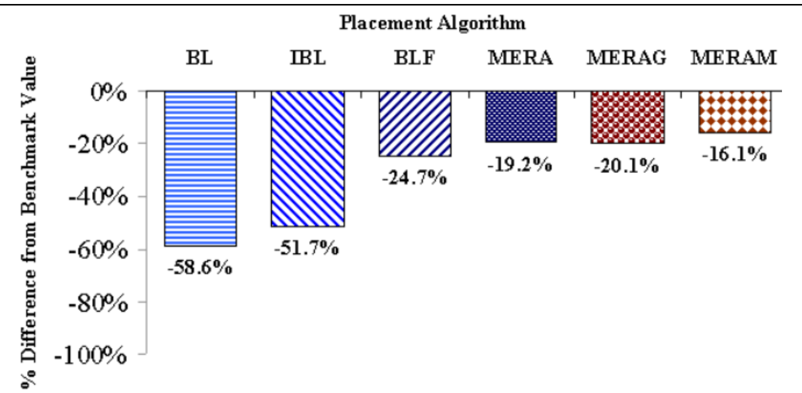
Number of Wins



Best outcome (% diff. from optimal value)



Mean outcome (% diff. from optimal value)



Worst outcome (% diff. from optimal value)

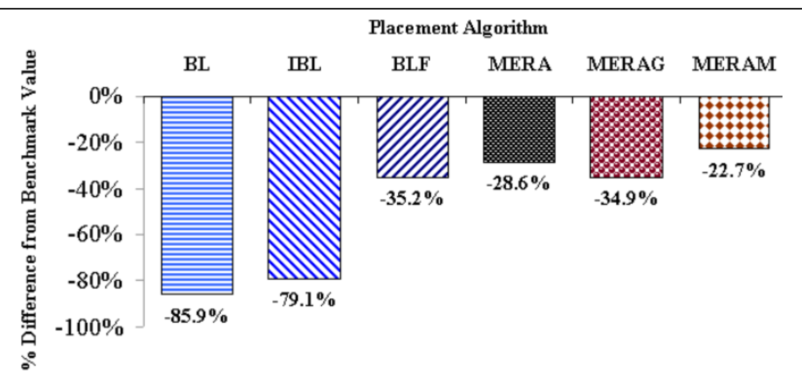


Figure 4-10: Performance comparison of algorithms w.r.t. *CR* for 100-module problem (A100)

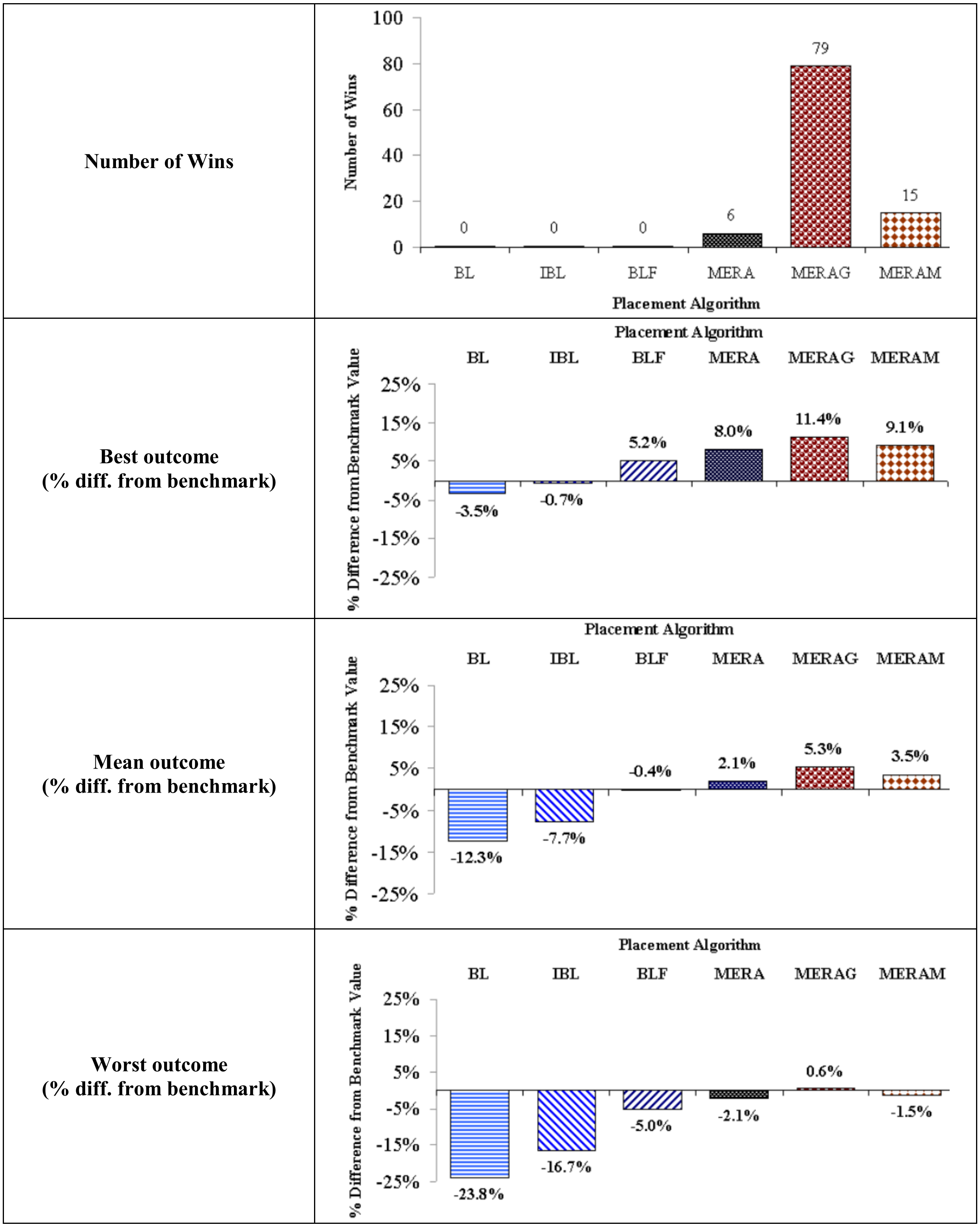


Figure 4-11: Performance comparison of algorithms w.r.t. *IMD* for 100-module problem (A100)

Before embarking on comparing the execution speed of algorithms, we reiterate that proposed algorithms were primarily motivated by the need for achieving improved layout quality and space utilization. As such, speed of execution was of secondary importance. Nevertheless, we have also compared the speed of these algorithms. The average CPU time taken by BL, IBL, BLF, MERA, MERAG, and MERAM for 100 random sequences of various problems is shown in Table 4-10.

		Problem Size (number of modules)				
		10	**15**	**25**	**50**	**100**
Placement Heuristic	**BL**	0.00019	0.00279	0.00705	0.01716	0.4152
	IBL	0.000208	0.00372	0.00889	0.02273	2.5713
	BLF	0.00465	0.00967	0.04355	0.1741	3.427
	MERA	0.00841	0.01271	0.05207	0.2863	13.361
	MERAG	0.00916	0.01881	0.09322	1.0249	61.011
	MERAM	0.00613	0.00973	0.03971	0.2997	16.127

Table 4-10: Average Time Elapsed (in seconds) per 100 iterations with *HT* as fitness metric.

Apparently, the average time taken by MERA and its adaptations increases significantly with the increase in the problem size. However, this observation is quite deceiving from various pragmatic perspectives as explained in the following. First, our experience has demonstrated that only a few random module sequences often furnished layouts with proposed algorithms alone that were superior to the best obtained after a whole cycle of BL+GA, IBL+GA. Likewise, a few random sequences were sufficient to produce layout alternative either superior or comparable with those obtained after the whole cycle of BLF+GA, as evident from results in Table 4-11.

Indeed, on a time-sharing multi-tasking system, measuring the running time of an algorithm has little meaning, because the CPU is being shared among many applications and the CPU elapsed time really measures the amount of competition among various applications running on the system (Mir & Imam, 2001). In addition, the execution time also depends on a range of factors like the CPU speed, size and speed of the main memory, size and speed of the cache memory, the size and speed of bus, computational efficiency of the code, computational efficiency of the programming language used, etc. (Mir & Imam, 2001). Visual BASIC is known to rank poorly in terms of computational efficiency when compared to such high-level programming languages as C, C++, FORTRAN, etc. (Mir & Imam, 2001). We opted for Visual BASIC mainly because of its capabilities in designing the interface. Indeed, some preprocessing of input data might also speed up the process significantly. For instance, converting all data to integer values through some scaling operations would allow more efficient integer operations compared to expensive floating-point computations.

Consequently, we believe that the real test of speed of an algorithm is the efficiency in quickly generating superior layout alternatives and not in quickly generating layout alternatives that may be inferior or offer little advantage. Indeed, MERA, MERAG, and MERAM not only resulted in better quality alternatives but also provided those faster than the ones obtained with several thousand iterations of GA with other existing placement algorithms like BL, IBL, and BLF (i.e. BL+GA, IBL+GA, and BLF+GA). Results from GA based optimization process for a 100-module benchmark problem (A100) are reported in Table 4-11. Once again, we have compared these algorithms based on *HT, MT, CR, IMD* and *QR*.

In Table 4-11, the first column shows the fitness metric used for contrasting the performance of all six algorithms, which are shown in the second column. The third column shows the fitness value of layout alternative obtained by a given algorithm for a given fitness metric. In order to readily gauge the relative performance of these algorithms, we have shown the percentage difference between the fitness of the layout alternative and the optimal or benchmark fitness value in parentheses. Evidently, the performance of the proposed decoding heuristics is significantly superior to that of existing ones when used in tandem with metaheuristics like GA.

Interestingly, even the performance of MERAG+GA and MERAM+GA is significantly better than that of BLF+GA for *HT* and *MT* as fitness functions. This is in contrast with the relative performance of these algorithms when Random Search was used. Indeed, it provides evidence that all three proposed decoding heuristics are more effective than the existing ones.

Furthermore, the superiority and diversity of layout alternatives obtained through proposed heuristics make any relatively higher computational cost compared to BL, IBL, and BLF a worthwhile trade-off. Moreover, the performance of BL, IBL, and BLF is known to deteriorate dramatically with the increase in the problem size as can be seen from Table 4-3 through Table 4-11 and as demonstrated by a series of earlier studies (Ahmad *et al.*, 2004d, 2004f; Jakobs 1996, Liu & Teng 1999, Hopper & Turton 2001, Wu *et al.* 2002).

Objective	Technique	Fitness (% diff. from Optimal/Ref.)
HT (Optimal = 100) The Lower the Better	**BL+GA**	123.5 (-23.5%)
	IBL+GA	118.6 (-18.6%)
	BLF+GA	112.2 (-12.2%)
	MERA+GA	**106.9 (-6.9%)**
	MERAG+GA	108.1 (-8.1%)
	MERAM+GA	107.9 (-7.9%)
MT (Optimal = 100%) The Higher the Better	**BL+GA**	78.8 (-21.2%)
	IBL+GA	83.5 (-16.5%)
	BLF+GA	89.75 (-10.25%)
	MERA+GA	**95.4 (-4.6%)**
	MERAG+GA	92.7 (-7.3%)
	MERAM+GA	93.1 (-6.9%)
CR (Optimal = 5000) The Higher the Better	**BL+GA**	3432 (-31.4%)
	IBL+GA	3905 (-21.9%)
	BLF+GA	4235 (-11.3%)
	MERA+GA	4709 (-5.8%)
	MERAG+GA	**4811 (-3.78%)**
	MERAM+GA	4718 (-5.64%)
IMD (Reference = 563000) The Lower the Better	**BL+GA**	553459.5 (+1.7%)
	IBL+GA	521419.6 (+7.4%)
	BLF+GA	483010.3 (+14.2%)
	MERA+GA	450759.9 (+19.9%)
	MERAG+GA	**429224.7 (+23.8%)**
	MERAM+GA	450216.1 (+20.1%)
QR (Scale: 1-10) The Higher the Better	**BL+GA**	1.5
	IBL+GA	1.75
	BLF+GA	3.5
	MERA+GA	4.5
	MERAG+GA	**5.25**
	MERAM+GA	4.5

Table 4-11: Comparison of Decoding Heuristics with GA for Problem **A100**

In contrast, MERA and its adaptations result in significantly higher performance improvements for larger problems furnishing another cogent reason for resorting to such approaches. Furthermore, the computational cost of MERA is not truly prohibitive in efficiently procuring an outcome better than that generated by BL, IBL, and BLF, as discussed earlier.

A review of relevant literature shows that BL-Fill (BLF) is among the most superior, if not the superior, module placement algorithms (Hopper & Turton, 2001). The BLF algorithm has a time complexity of $O(n^3)$ and results in a relatively slower process both in terms of average computational time for each evaluation as well as the number of evaluations required to achieve superior results.

A comparison of the best solutions obtained through 100 random sequences of benchmark problems shows that the performance of MERA is better than BLF. For instance, MERA results in the best outcome with a relative distance of 16.3% from optimal *HT* for a 50-module problem (ref. Table 4-6). In contrast, BLF results in the best outcome having a relative distance of 18.1% from the optimal. In terms of *MT*, the best outcome obtained with MERA is 13.3% from the optimal and the one obtained with BLF is 14.9% from the optimal. In terms of *CR*, the best outcome obtained with MERA is 10% from the optimal and the one obtained with BLF is 12% from the optimal. In terms of *IMD*, the best outcome obtained with MERA is superior to the one obtained with BLF, too.

Likewise, the relative distance between the best solution in terms of *HT* by MERA and the optimal solution is 12.1% for 100-module problem against 13.2% by BLF (ref. Table 4-9). In terms of *MT*, the best outcome obtained with MERA is 11.2% from the optimal against 12.2% with BLF. In terms of *CR*, the best outcome obtained with MERA is 9.1% from the optimal against 16.7% with BLF. In terms of *IMD*, the best outcome obtained with MERA is superior to the one obtained with BLF.

In addition, the GA based optimization approach reveals that the advantage MERA and its adaptations have over BLF translates into significant gains through metaheuristic search procedures, as can be seen from Table 4-11 and Figure 4-12 to Figure 4-14. Contrasting these values reveals that MERA, MERAG, and MERAM are more effective than BLF. Furthermore, about 50,000 evaluations of BLF in a metaheuristics search results in layouts that are either inferior or, at best, comparable to those obtained in less than 1000 evaluations of MERA. Consequently, MERA is an efficient and effective placement algorithm for layout optimization.

Moreover, we compared layouts generated by BL, IBL, BLF, MERA, MERAG, and MERAM for the sequence in which modules are ordered with respect to Decreasing Length (DL). Such ordered sequences are known to produce large performance improvements in space utilization (Ahmad *et al.*

2004f; Burke *et al.*, 2004; Jakobs 1996; Hopper & Turton 2001; Wu *et al.* 2002). The relative performance of all six algorithms for a DL sequence is summarized in Table 4-12.

Objective	Technique	Fitness (% diff. from Optimal/Ref.)
HT (Optimal = 100) The Lower the Better	**BL+DL**	123.8 (-23.8%)
	IBL+DL	128.2 (-28.2%)
	BLF+DL	115.2 (-12.2%)
	MERA+DL	**107.9 (-7.9%)**
	MERAG+DL	120.7 (-20.7%)
	MERAM+DL	113 (-13.0%)
MT (Optimal = 100%) The Higher the Better	**BL+DL**	80.3 (-19.7%)
	IBL+DL	77.6 (-22.4%)
	BLF+DL	89.2 (-10.8%)
	MERA+DL	**92.1 (-7.9%)**
	MERAG+DL	82.5 (-17.5%)
	MERAM+DL	88.0 (-12.0%)
CR (Optimal = 5000) The Higher the Better	**BL+DL**	3356.6 (-32.8%)
	IBL+DL	3821.7 (-23.6%)
	BLF+DL	4287.5 (-14.2%)
	MERA+DL	**4678.6 (-6.3%)**
	MERAG+DL	4399.4 (-12.0%)
	MERAM+DL	4595.3 (-8.1%)
IMD (Reference = 563000) The Lower the Better	**BL+DL**	543561 (+3.45%)
	IBL+DL	570023 (-1.25%)
	BLF+DL	546666 (+2.90%)
	MERA+DL	**523992 (+6.93%)**
	MERAG+DL	540347 (+4.2%)
	MERAM+DL	541458 (+3.83%)
QR (Scale: 1-10) The Higher the Better	**BL+DL**	1.5
	IBL+DL	2.25
	BLF+DL	2.75
	MERA+DL	**6.5**
	MERAG+DL	5.5
	MERAM+DL	5.0

Table 4-12: Comparison of Decoding Heuristics for DL sequence for Problem **A100**

Notably, this single sequence of MERA+DL produces layouts that are superior to the best obtained by BL+GA and IBL+GA as can be seen from Table 4-11 and Table 4-12. It implies that only one evaluation of MERA is enough to beat such existing algorithms as BL and IBL by wide margins. Outcomes of such an ordered sequence for 100-module problem (A100) are shown in Appendix D for visual comparison purposes.

Another interesting observation is that an increase in the number of GA iterations for BL and IBL may provide some marginal improvement against rigid fitness measures. However, the layout Quality Rating or *QR*, as measured by experts' subjective ranking, deteriorates, as evident from Table 4-11. Furthermore, increasing the number of GA iterations is more likely to furnish a population that consists of identical or closely related solutions. Indeed, such closely related set of layout alternatives lacks the requisite diversity and does not provide a genuine range of superior alternatives to decision-makers. Consequently, the resulting superior quality and diversity of layout decision alternatives obtained through MERA, MERAG, and MERAM makes any higher computational cost a worthwhile trade-off.

In short, the MERA algorithm and its adaptations result in more efficient, robust, and superior layout optimization than the existing algorithms. It demonstrates that MERA, somehow, captures the dynamics of the layout design problem more aptly.

The GA convergence rate of decoding heuristics for a 100-module problem (A100) with *CR* as fitness measure is shown in Figure 4-12 and Figure 4-13. It can be seen that BL, IBL, and BLF start with inferior solutions and converge to solutions that are comparable to the starting solution obtained from MERA, demonstrating the efficiency and effectiveness of MERA.

Similarly, the convergence rate of placement algorithms for the same 100-module problem (A100) with *HT* as fitness measure is shown in Figure 4-14. Once again, it can be seen that MERA converges to solutions that are more superior. Whereas, BLF seems to perform marginally better than MERAG and MERAM. This is in league with our observation with RS, where BLF seemed to provide somewhat better solutions with respect to packing height than those obtained from MERAG and MERAM. However, as already discussed, the best outcome from a GA cycle with MERAG or MERAM is significantly superior, in terms of *HT*, to that obtained from a GA cycle with BLF, as can observed from Table 4-11.

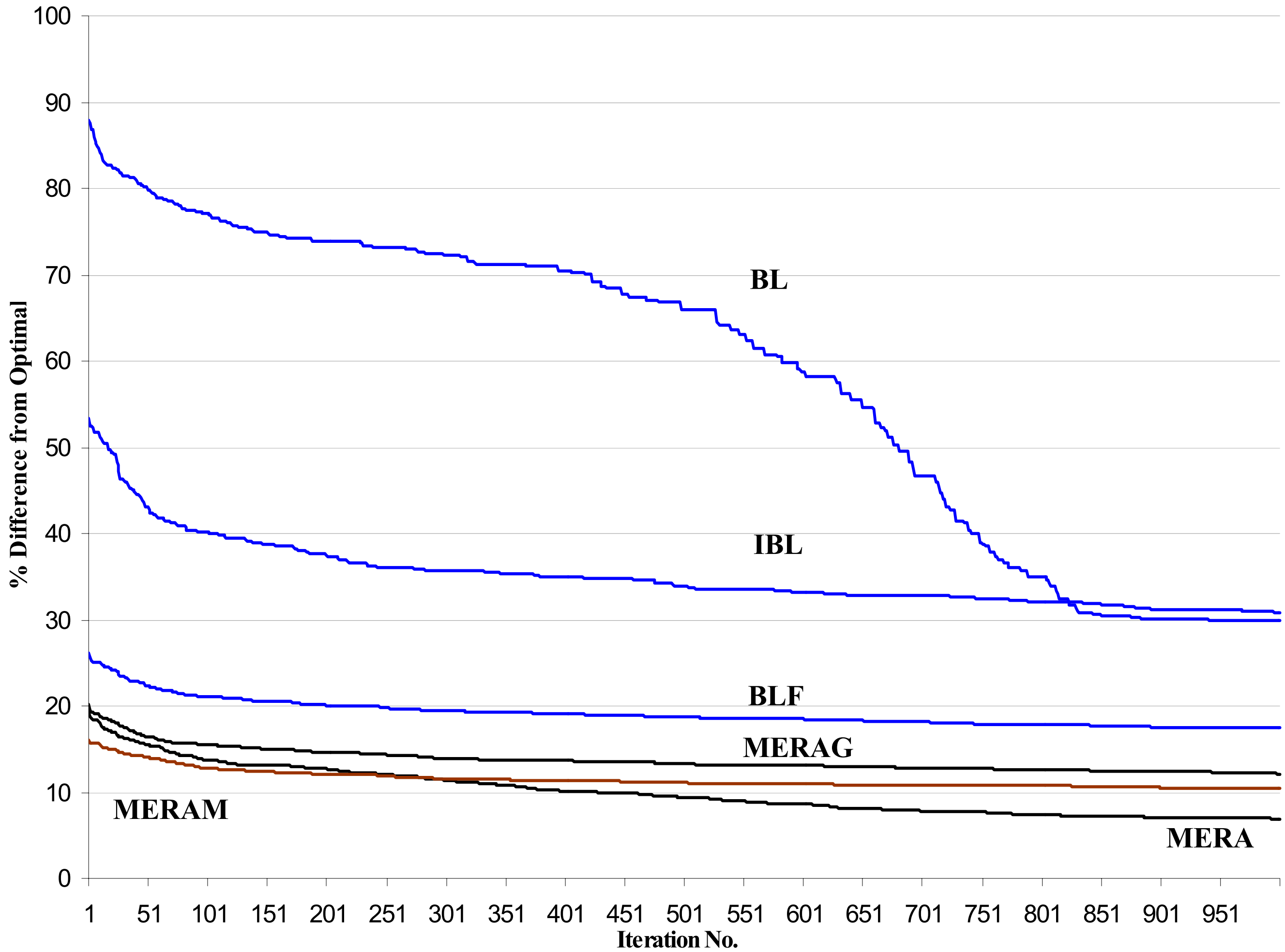


Figure 4-12: GA Convergence (average *CR*) for the 100-module problem (A100)

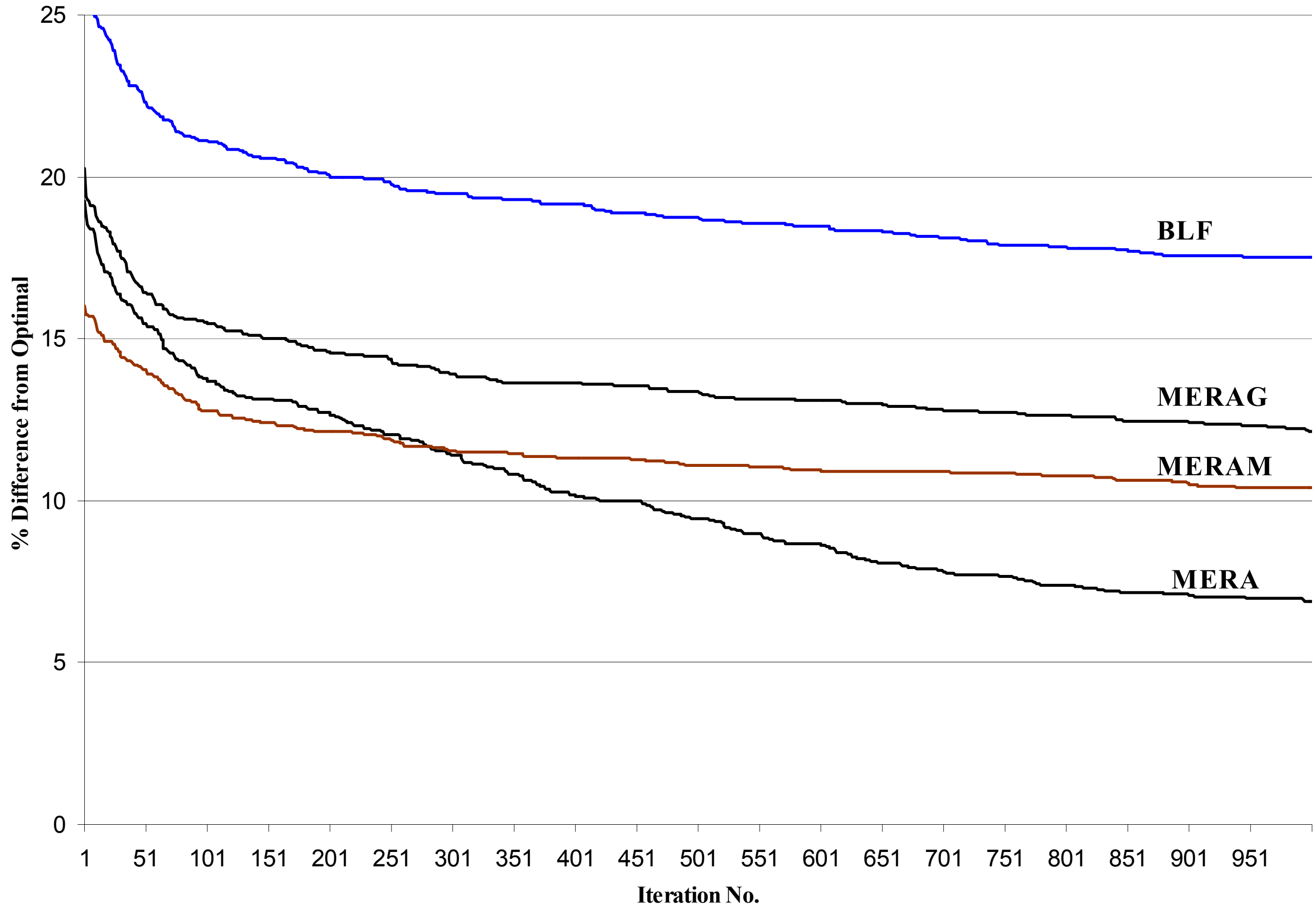


Figure 4-13: GA Convergence (average *CR*) for the 100-module problem (A100)

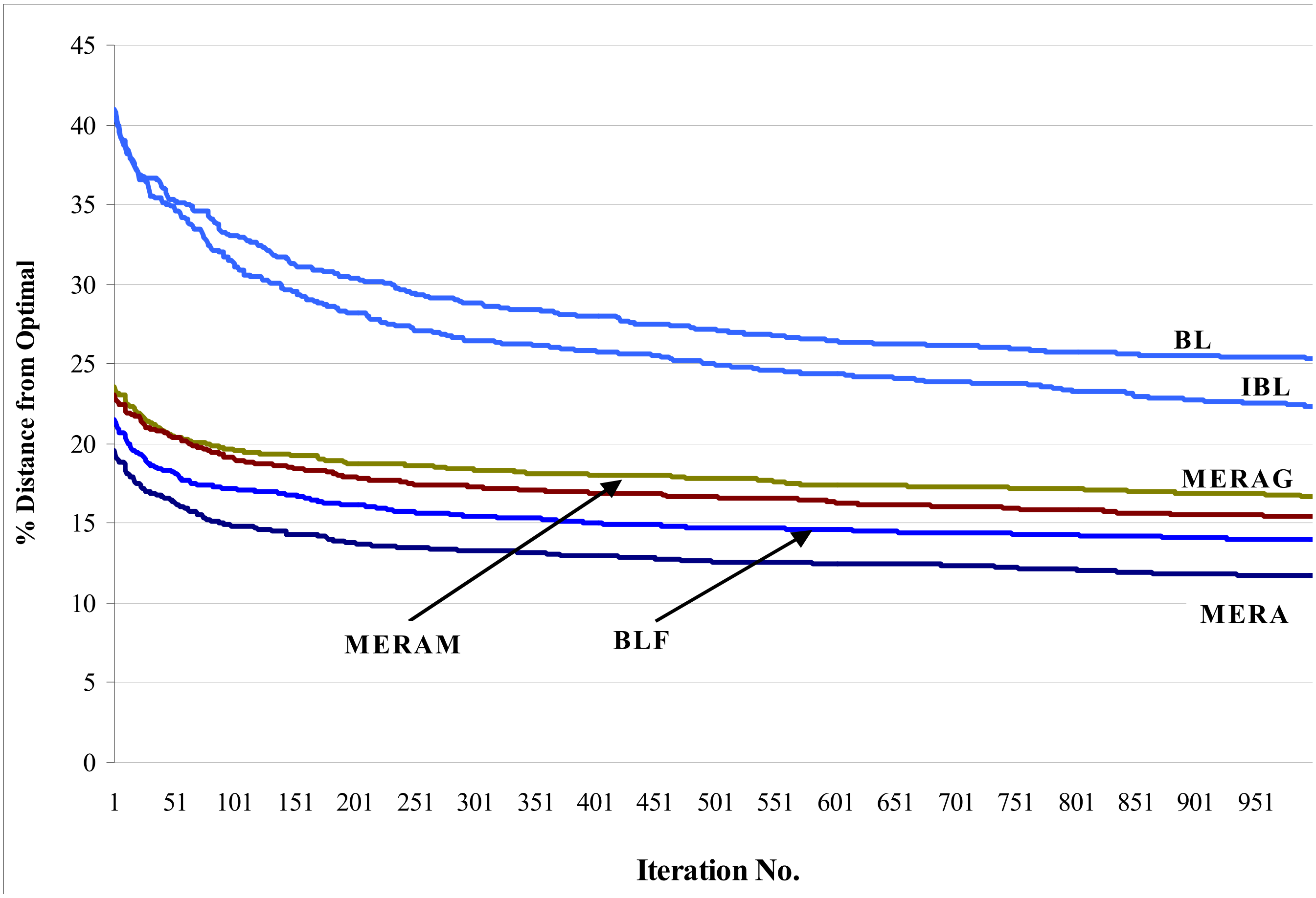


Figure 4-14: GA Convergence (average *HT*) for the 100-module problem (A100)

The comparison of MERA, MERAG, and MERAM is a bit tricky enterprise. It can be seen that MERA provides layouts superior in *HT* and *CR* more frequently than MERAG or MERAM. The same is true in terms of subjective layout quality rating *QR*. However, it should be noted that MERAG and MERAM seem to be more appropriate for applications where inter-module interaction calls for compact packing while minimizing the total inter-module distance, such as the wiring length in some VLSI layout design. In such a scenario, the layout design using MERA would require an auxiliary fitness evaluation and optimization mechanism that could debase the efficiency of the overall process. One effective methodology would be to employ metaheuristics in conjunction with MERA; however, such metaheuristics require a large number of evaluations even at the onset of the process. In contrast, only few random sequences of MERAG could provide a very good, if not the best, solution. However, we want to emphasize that our preliminary studies show that MERA is more consistent in furnishing layouts with higher space utilization and layout quality.

4.9 Bin-Packing Case Studies

Here, we present few test cases to demonstrate the effectiveness of IDEAL and the proposed decision-making paradigm for layout design. Ironically, there is not much literature available on benchmark problems that involve layout design using modules that are unequal in size, fixed in shape, fixed in orientation, and involve subjectivity and uncertainty in placement preferences.

In order to test the viability of IDEAL, we generated several layout alternatives for a 25-module problem using various algorithms. This 25-module problem was procured from a packing industry and has been included in Appendix E. We gave those alternatives to an expert for getting subjective ratings based on space utilization and layout symmetry as well as any possible manipulation and refinement of those layouts. The expert have more than 20 years of teaching, researching, and practicing experience in layout design applications. The expert neither had knowledge of algorithms used to generate these alternatives nor had any information about the fitness metrics used to evaluate these layouts. Results of those evaluations were used in the training of PDA, as well, as discussed in Section 5.6. Few interesting instances of this exercise are presented here to demonstrate the efficacy of IDEAL.

4.9.1 Case I

The layout alternative presented in Figure 4-15 received a rating of 85 out of 100 from the expert. It can be seen that this packing topology has higher symmetry as well as space utilization. The values of fitness metrics as gauged by IDEAL are consistent with this high rating by the expert. For instance, *CR* shows 100% space utilization and *MT* is about 91%. The Symmetry of Count comes out to be 94%, the Symmetry of Cohesion comes out to be 75%. In addition, the sum of inter-module distances is around 4780 (against a benchmark value of 5150), which is quite superior.

It shows that fitness metrics used in IDEAL have high correlation with the experts' subjective rating. Furthermore, this layout was obtained using a single random iteration with MERAM and demonstrates the ability of the heuristic to efficiently produce layout with higher space utilization as well as aesthetic contents.

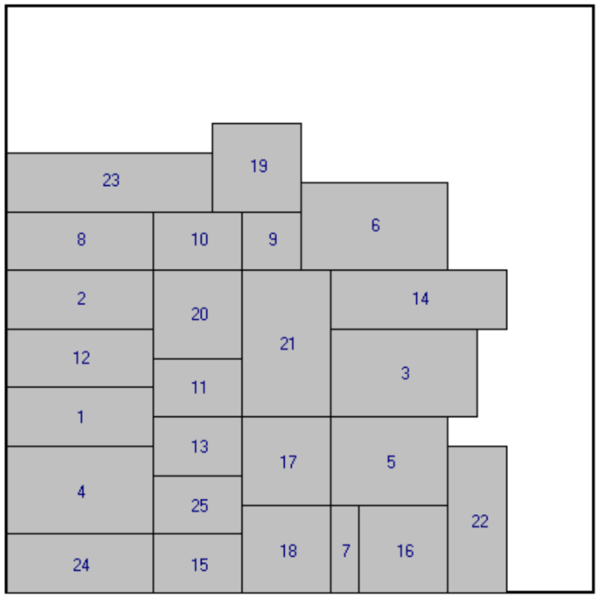


Figure 4-15: Case I – Layout Alternative

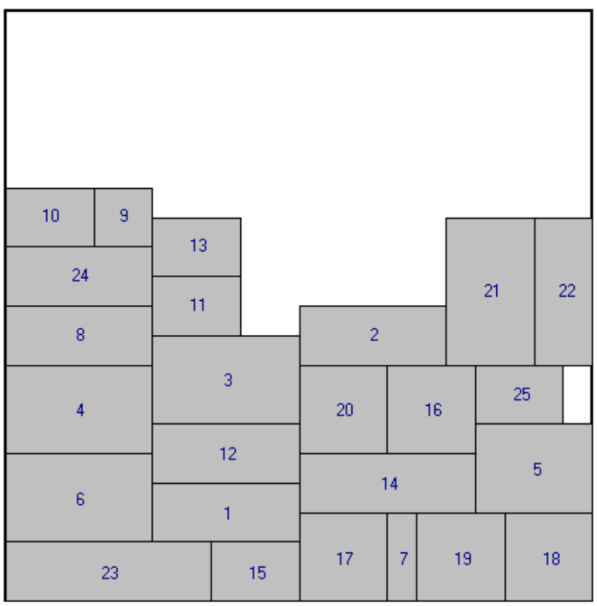


Figure 4-16: Case II – Layout Alternative

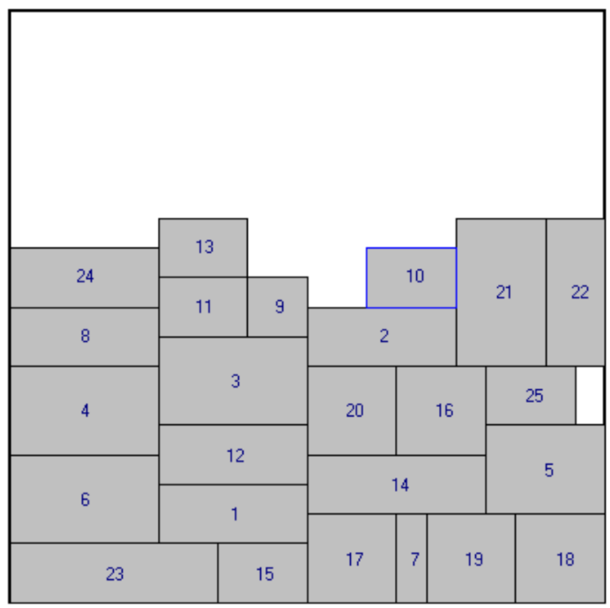


Figure 4-17: Case II – Refined Layout

4.9.2 Case II

The layout alternative presented in Figure 4-16 received a rating a rating of 75 out of 100 from the expert. It can be seen that this packing topology can easily be modified to achieve higher symmetry as well as space utilization. For instance, we moved module-9 on top of module-3 and module-10 on top of module-2.

These simple moves improved the fitness of the layout significantly. For instance, the Symmetry of Density increased from 26% to 96% while Module Tightness increased from 83% to about 93%, Furthermore, the height of the packing pattern came closer to the optimal height as well as the sum of inter-module distances has been reduced from 5438 to 5114. The refined layout is shown in Figure 4-17. Apparently, DM has given this alternative higher rating based on these intuitive observations.

4.9.3 Case III

The layout alternative presented in Figure 4-18 received a rating a rating of 70 out of 100 from the expert. Apparently, the layout shown in Figure 4-18 does not seem to be a superior outcome in terms of symmetry or space utilization. However, once again, the higher rating by the expert is a reflection on the fitness potential of the layout alternative following few simple manipulations. It can be seen that the modified topology shown in Figure 4-19 has higher symmetry as well as space utilization.

It involved the following manipulations: move the module-5 to the bottom-right corner of the bin; move the module-23 on top of modules 5 and 18; move the module-11 to the right of the module-12; move modules 7, 17, and 21 on top of moduel-23; shift modules 1, 4, and 8 downwards and swap position of modules 1 and 4; move module-14 to the right of module-10. All these nine moves took less than 2 minutes to complete and naturally followed each other. It resulted in a superior outcome with the Module Tightness at around 87%, Symmetry of Density at around 92%, Symmetry of Distribution at around 88%, as well as significant decrease in the Height of the packing pattern. The resultant layout subsequently received a subjective a rating of 90 out of 100 by the DM.

4.9.4 Case IV

The layout alternative presented in Figure 4-20 received a rating a rating of 75 out of 100 from the expert. Once again, the higher rating by the expert is a reflection on the fitness potential of the layout alternative following few simple manipulations. It can be seen that the modified topology shown in Figure 4-21 has higher symmetry as well as space utilization.

It involved the following moves: move module-21 to the right of module-11; move module-17 on top of module-21; move modules 16 and 20 on top of module-21; move module-1 on top of modules 17 and 22; move module-4 on top of module-1; move module-8 on top of module-4. All these six moves took less than one and a half minute to complete and naturally followed each other. It resulted in a superior outcome with Module Tightness at around 90%, Symmetry of Distribution at around

87%, as well as significant decrease in the Height of the packing pattern. When this resultant pattern in Figure 4-21 was given to DM, it received a subjective rating of 85 out of 100.

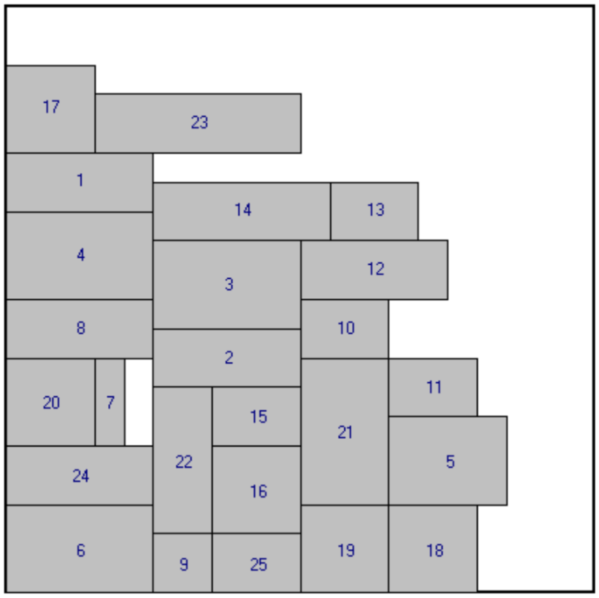


Figure 4-18: Case III – Layout Alternative

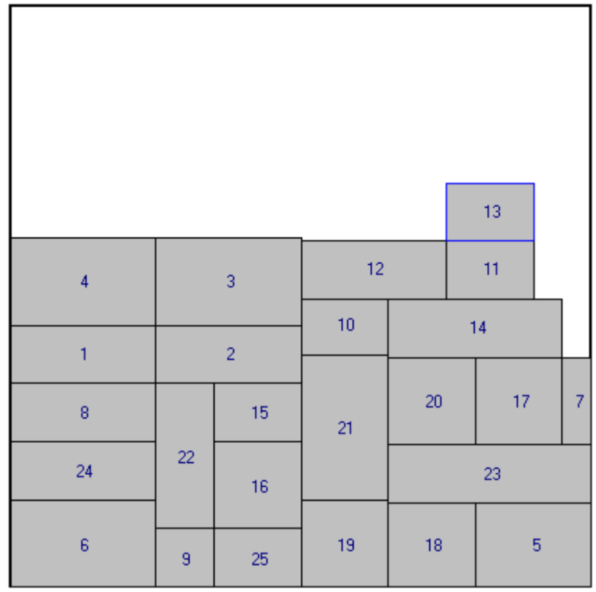


Figure 4-19: Case III – Refined Layout

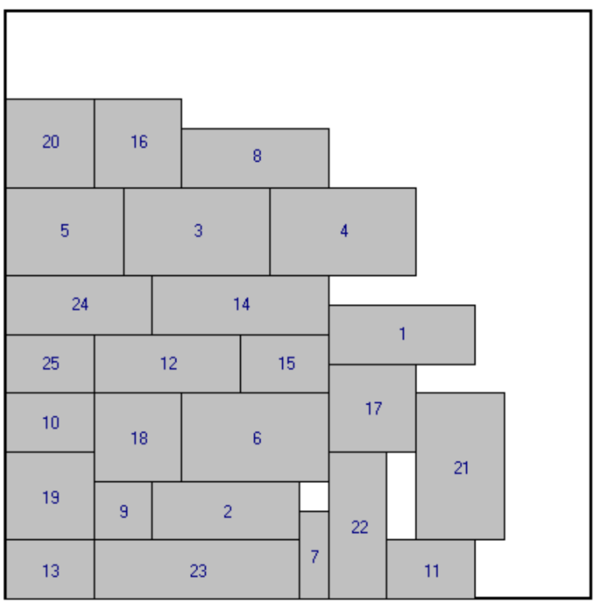


Figure 4-20: Case IV – Layout Alternative

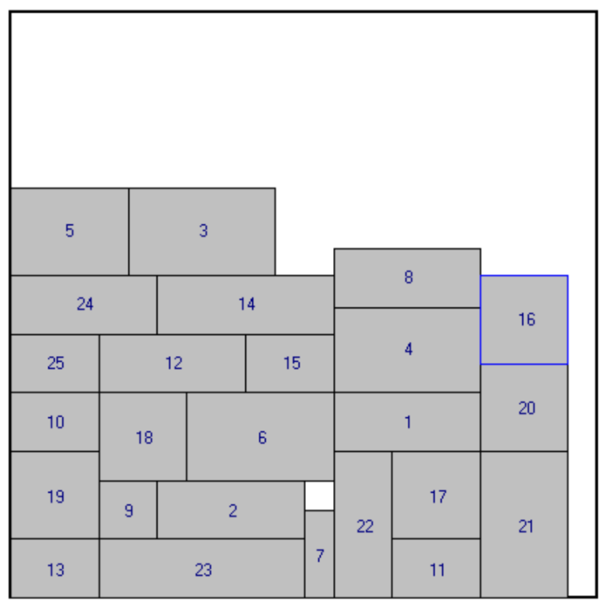


Figure 4-21: Case IV – Refined Layout

4.9.5 Case V

The packing pattern Figure 4-22 also requires few simple and swift moves to significantly enhance the space utilization as well as layout symmetry, as shown in Figure 4-23.

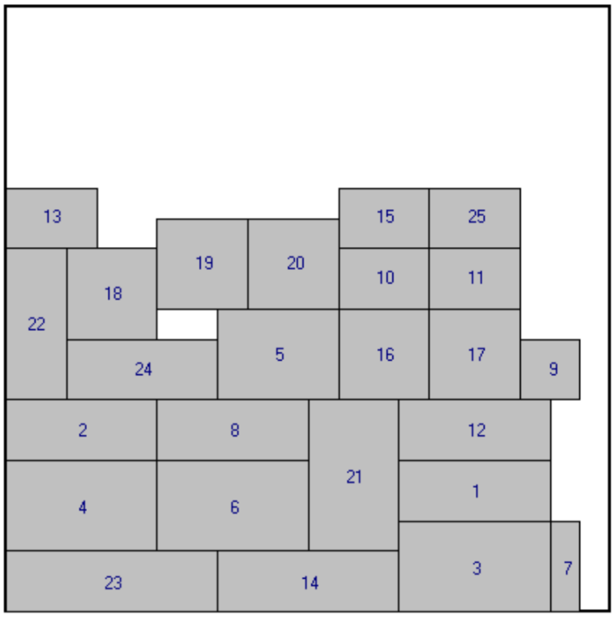


Figure 4-22: Case V – Layout Alternative

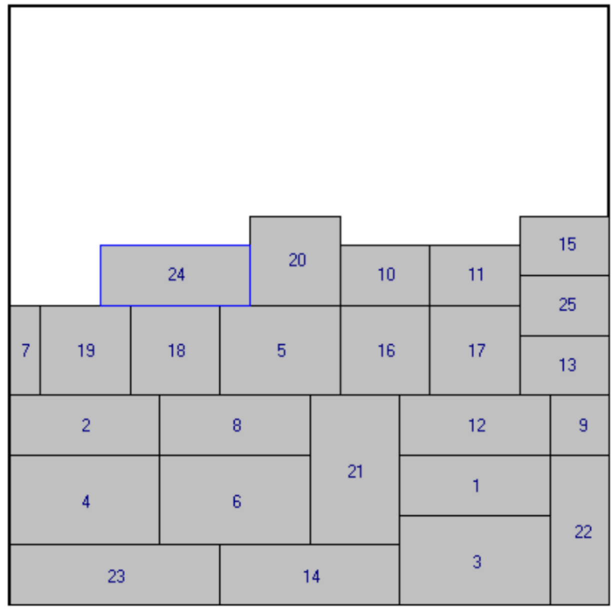


Figure 4-23: Case V – Refined Layout

4.10 Summary

In this Chapter, we employed a 2D-BPP formulation of the layout design problem and described our vision and implementation of an Intelligent Layout Generator that is capable of employing preferences and parameters furnished by experts or some intelligent inferencing mechanism. We proposed some new placement algorithms and demonstrated the promise those algorithms hold in efficiently generating superior and diverse layout alternatives. The hierarchical optimization approach is realized through a GA based metaheuristic search hybridized with deterministic placement heuristics, which act both as a decoder and as a tuner of the layout solutions. Our hybridized optimization approach, as investigated in this thesis, not only achieved speed and efficacy but also superior quality layouts. Furthermore, we have formulated various encompassing quantitative determinants of layout utility. In the subsequent chapter, we provide our vision and implementation of the inferencing mechanism for reasoning with available knowledge and furnish preference weights and parameters to ILG.

Chapter 5

PREFERENCE MODELING, INFERENCING, & DISCOVERY

5.1 Introduction

The brain of an expert system is the Inference Engine that contains general algorithms and functionalities for manipulating, and reasoning with, the knowledge stored in the knowledge base (Turban & Aronson, 2001). Here we describe the design and implementation of a fuzzy technology based prototypical Preference Inferencing Agent (PIA) for performing inferencing and reasoning tasks in IDEAL.

The core concept involves employing a PIA comprising of fuzzy sets, rules, and preferences in obtaining penalties and rewards for the hybrid fitness evaluation functions as well as various critical parameters for ILG and PDA. The primary benefit of fuzzy rule-based system is that its functioning mimic more of human expert rules. The traditional rigid and myopic fitness functions do not serve well in such complex, subjective, and uncertain problem domains as layout design. Indeed, multi-criteria fitness functions are deemed more appropriate for automatic generation, evaluation, and comparison of layout alternatives. However, IDEAL has provisions for decision-maker to specify Significance Parameter (SP) and Preference Parameter (PP) in both crisp and fuzzy manner, thereby increasing the flexibility and the ease with which decision-makers may creatively adapt their preferences.

The rest of the chapter is structured as follows. Section 5.2 outlines benefits of employing fuzzy logic in various forms in layout optimization as well as our fuzzy preference modeling and inferencing approach. Section 5.3 presents some qualitative fitness evaluation metrics. Section 5.4 describes our multi-criteria fuzzy decision making approach. Section 5.5 elaborates the working of PIA through an example. Section 5.6 discusses pros and cons of a neuro-based expert system enabling automated discovery of preferences and our vision of a multi-layer perceptron based backpropagation network. Section 5.7 summarizes the chapter.

5.2 Fuzzy Technology in Layout Design

As already mentioned, the knowledge pertinent to layout design and analysis is usually imprecise, incomplete, inconsistent and vague. The abstract, subjective, and uncertain nature of most layout design guidelines and rules render fuzzy technology an excellent candidate for the implementation of the inference engine. As a modeling and inferencing tool, FL provides ability to work with incomplete or inconsistent information. (Konar, 2001; Negnevitsky, 2002). Indeed, there are various formalisms available for dealing with missing information (Liu *et al.*, 1997; Tresp *et al.*, 1994). However, the formalism available in FL for tolerating or predicting missing information has demonstrated to be more robust and tractable than other formalisms (Negnevitsky, 2002). Studies have shown that just a few fuzzy rules may provide better results than a few thousand rules of other common formalisms (Berthold & Huber, 1998).

Our objective in developing a generic research framework, which might be adapted to various layout design applications, intensifies the need for employing robust preference modeling and reasoning methods. It is our belief that FL furnishes a more robust formalism through its powerful capabilities in linguistic modeling, inferencing, and analysis under uncertainty. The PIA described here translates the domain-dependent knowledge into domain-independent parameters for the use by ILG. Consequently, it brings a good deal of flexibility in the automated layout design process.

Furthermore, FL simplifies the knowledge acquisition process by facilitating elicitation of expert's opinion and readily transforming those in the suitable fuzzy functions (Cox, 1999). The simple linguistic formalism of preferences in FL also facilitates utilization of experts' creativity in an easy, flexible, efficient, and informed manner. Consequently, the ability of FL to realize a complex non-linear input-output relationship as a synthesis of multiple simple input-output relations offers great promise (Kavcic 2002).

As noted, these favorable characteristics of FL have drawn attention from some researchers in the layout design. The fuzzy modeling and inferencing techniques have successfully been applied to placement decisions in general layout design problem and this body of literature is growing fast (Ahmad *et al.*, 2003; Ahmad *et al.*, 2004b; Aiello & Enea, 2001; Badiru & Arif, 1996; Deb & Bhattacharyya, 2004; Dweiri & Meier, 1996; Evans *et al.*, 1987; Grobelny, 1987a; Grobelny, 1987b; Karray *et al.* 2000b; Kang *et al.*, 1994; Kim *et al.*, 2001; Raoot & Rakshit, 1993; Raoot & Rakshit, 1991; Soltani & Fernando, 2004; Youssef *et al.*, 2003a; Whyte & Wilhelm 1999a, 1999b; Zha & Lim, 2003). In short, the application of FL is not a new inquiry in the LD field. Nevertheless, we want to synergistically employ FL in its various efficacious application modes along with other powerful soft computing techniques.

As already mentioned, most research employing FL in layout design has used FL as a linguistic modeling tool. However, some literature employing FL in layout design as an analytical tool also exists (Karray *et al*., 2000a; Tam *et al.*, 2002). Nevertheless, the literature employing FL in LD as an algorithmic tool is very meager (Khan *et al.*, 2002; Youssef *et al*., 2003). The PIA described in this chapter employs FL as both linguistic and analytical tools. Nonetheless, we plan to employ FL as algorithmic tool in future by drawing on some fuzzy rules for skipping less promising placement moves, thus expediting the overall layout generation process.

One of the foremost requirements in the use of FL is the determination of fuzzy Membership Functions (MF) through experts' knowledge. The MF completely defines fuzzy sets. Typically, MF used in fuzzy knowledge-based systems are the triangular and trapezoidal functions as those provide an adequate representation of experts' knowledge and significantly simplify the computational process (Ahmad *et al.*, 2003, 2004c; Negnevitsky, 2002; Saletic *et al.*, 2002; Triantaphyllou & Lin, 1996). Nevertheless, the choice of MF is based more on personal preference than any mathematical justification (Keely, 1997). In Figure 5-1, we have shown a triangular MF, which may mathematically be expressed in the following. It has three parameters '*a*' (minimum), '*b*' (middle) and '*c*' (maximum) that determine the shape of the triangle.

$$\textit{Triangular } MF(x;a,b,c) = \max\left[\min\left(\frac{x-a}{b-a},1,\frac{c-x}{c-b}\right),0\right] \qquad \text{Equation 5-1}$$

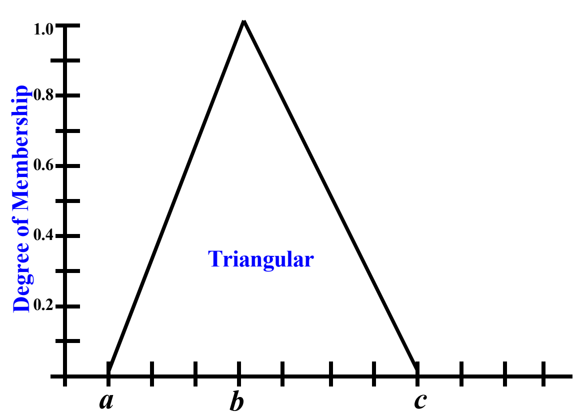


Figure 5-1: A Triangular Membership Function.

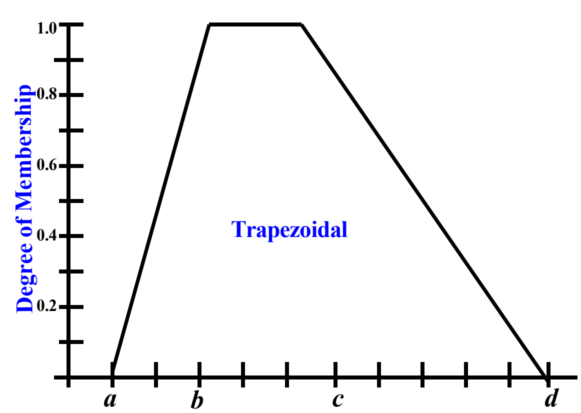


Figure 5-2: A Trapezoidal Membership Function.

A trapezoidal MF is depicted in Figure 5-2, which may mathematically be expressed by the following:

$$Trapeziodal\ MF(x;a,b,c,d) = \max\left[\min\left(\frac{x-a}{b-a},1,\frac{d-x}{d-c}\right),0\right] \qquad \text{Equation 5-2}$$

For simplicity of implementation, experimentation, and interpretation sake, we have primarily employed triangular MF in our research prototype. However, the augmenting the prototype with the use of trapezoidal MF is not a difficult task.

5.2.1 Preference Modeling

Most linguistic rules in layout design consist of two parameters, the significance and the preference. A *Significance Parameter* (SP) tells 'how important' certain criterion is for the overall fitness of the layout. Whereas, a P*reference Parameter* (PP) tells 'how much' of a certain aspect/criteria should be incorporated in the layout generation (Giegel & Loui, 2001).

Significance Parameter

In a broad sense, an SP determines the weight that would be assigned to a specific fitness metric in a hybrid fitness function (Ahmad *et al.*, 2004b; Evans, 1987). We have modeled SP with fuzzy MF shown in Figure 5-3. These mimic the idea used in REL charts for signifying proximity stipulations. These involve MF for *A* (**A**bsolutely important), *E* (**E**specially important), *I* (**I**mportant), *O* (**O**rdinarily important), and *U* (**U**nimportant). The difference between classical REL and our approach is that here we do not have an MF for X or 'not desirable' as this scenario could easily be modeled by setting SP to unity and PP to zero.

Preference Parameter

As indicated, a preference parameter (PP) tells how much decision makers prefer to incorporate a certain aspect/criterion in layout alternatives (Ahmad *et al.*, 2004b; Evans, 1987; Giegel & Loui, 2001). In a broad sense, any departure from the PP of a specific measure would result in penalizing the hybrid fitness function, depending on the weight or SP assigned to that measure. Membership functions of PP for various considerations would depend on decision-makers' preference. These membership functions will be used by PIA for generating preference weights for ILG. Alternatively, IDEAL furnishes decision-makers' with an ability to provide absolute values for these parameters.

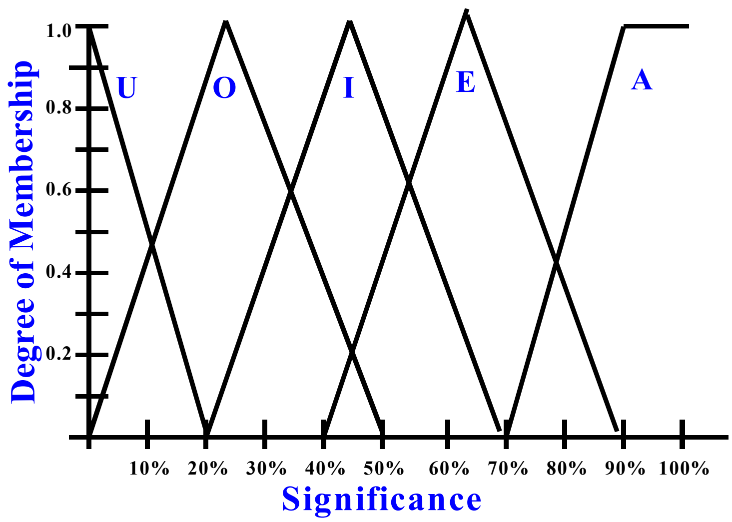


Figure 5-3: Membership Functions for the Significance Parameter.

5.2.2 Inferencing Mechanism

A typical fuzzy decision-making system (FDMS) accepts fuzzy and/or crisp preferences and transforms those into crisp weights, using the fuzzy rules present in the knowledge base, for employment in some layout fitness evaluation function (Karray & de Silva, 2004). One possible methodology for measuring the utility/fitness of the layout based on both tangible as well as intangible criterion would involve the development of a composite fitness function, comprising of weighted sums of utilities arising from various design issues. Weights in such a fitness function correspond to preferences provided by experts and must be calculated/inferred for further use (Ahmad *et al.*, 2003, 2004b; Geigel & Loui, 2001; Triantaphyllou *et al.*, 1998). Consequently, fuzzy inferencing can be described as a process of mapping a given input to an output by employing the theory of fuzzy sets. Incidentally, there is a variety of fuzzy inferencing mechanisms available in the literature. However, the Mamdani-style inference method is the most popular technique for capturing experts' knowledge, sanctioning a more intuitive and human-like description of expertise (Negnevitsky 2002).

The configuration of PIA, as depicted in Figure 5-4, comprises of four main components. These include the Fuzzification component, the Knowledge Base, the Rule Evaluation unit, and the Defuzzification unit. Here we discuss roles of these units within the context of the Mamdani-style fuzzy inferencing mechanism employed in PIA. The first step in Mamdani style fuzzy inferencing mechanism is to fuzzify all the crisp inputs and determine the degree to which these inputs belong to

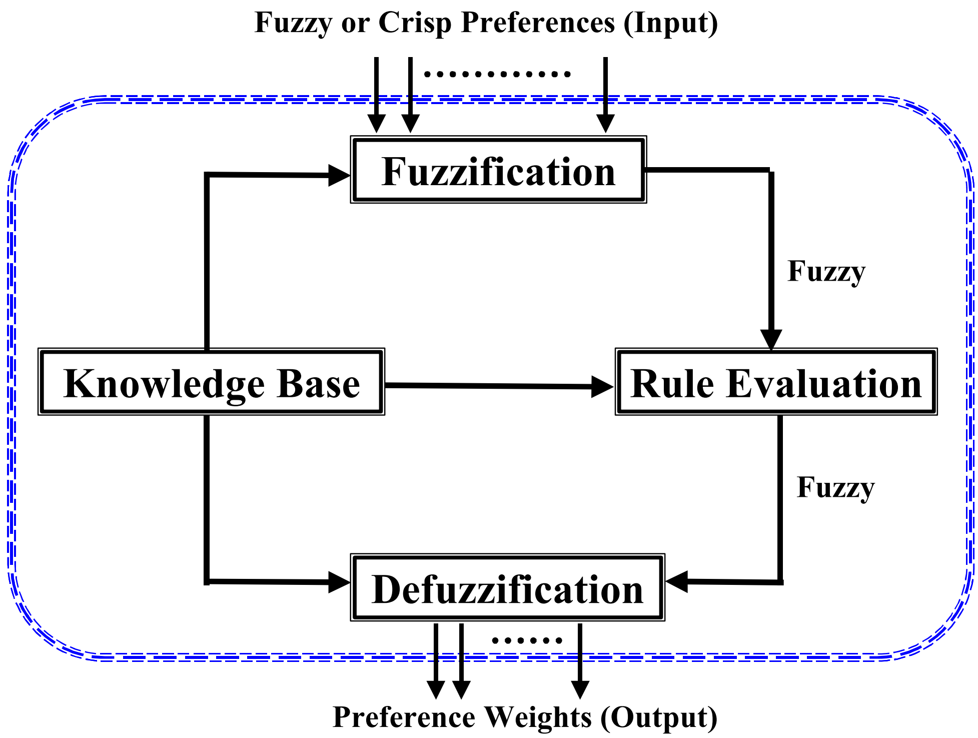


Figure 5-4: Preference Inferencing Agent (PIA).

each of the appropriate fuzzy sets. The *Fuzzification* component measures the values of input and output, transfers the range of values into a corresponding universe of discourse, and converts them into natural language. The *Knowledge Base* contains the experts' knowledge of the application domain and the control rules of the process. These rules may be fuzzy or crisp. The MF for fuzzy rules are also determined by experts based on their knowledge of the problem domain.

The *Rule Evaluation* unit applies decision-making logic that simulates the experts' decision-making process based on a fuzzy concept. This step requires taking the fuzzified inputs, and applying them to antecedents in the fuzzy rules (Karray & de Silva, 2004). The results of antecedent evaluation are then applied to the membership function of the consequent. The consequent membership function is 'clipped' or 'scaled' to the level of the truth-value of the rule antecedent. We adopted a scaling method in our rule evaluation as it retains the shape of the fuzzy MF.

The rule evaluation is followed by the *Aggregation* step involving unification of the outputs of all rules (Negnevitsky, 2002). The input to the aggregation process is the clipped or the scaled consequent membership functions from the rule evaluation module and the output is one fuzzy set for each output variable.

The fuzziness facilitates in evaluating rules in the layout design using FL; however, the final output of the PIA has to be a crisp number so that it could be used in some fitness function in ILG. This process of converting fuzzy values to crisp values is called defuzzification. The most prevalent technique for *defuzzification* in expert systems is the *centroid* technique where a vertical line carves the aggregate fuzzy set into two equal masses (Karray & de Silva, 2004). We have also employed this centroid method for defuzzification purposes in our system. This way vague linguistic rules are used in realizing important and useful crisp values for evaluation and generation of superior layout alternatives. These crisp values are used in ILG in the form of preference weights of a hybrid layout fitness function. The fuzzy inferencing mechanism is further elaborated in Section 5.5 with an example.

5.3 Fitness Metrics

As discussed in Chapter 4, we classify layout fitness metrics into two broad categories, namely Quantitative Fitness Metrics and Qualitative Fitness Metrics. We reiterate that we have termed some fitness metrics as quantitative only because the fitness values those metrics are designed to capture might be defined objectively with relative ease under highly certain and predictable scenarios, as opposed to such intrinsically subjective fitness measures as aesthetic values of a layout plan.

5.3.1 Quantitative Fitness Metrics

Various effective quantitative fitness evaluation metrics have already been mentioned in Chapter 4. These include Packing Height, Contiguous Remainder, Module Tightness, and Inter-Module Interaction. All these metrics capture some notion of space utilization and mutual positioning or interaction of modules in the given spatial configuration. The incommensurable attributes of layout fitness may be combined in some kind of hybrid fitness evaluation function through normalized values of fitness metrics (Triantaphyllou *et al.*, 1998). The Module Tightness (*MT*) as described in Section 4.5.4 already takes a normalized form in which *MT* may vary from 0 to 100 with a higher value representing a better space utilization. Furthermore, normalization of fitness metrics like Packing Height (*HT*) and Contiguous Remainder (*CR*) against some appropriate standard or benchmark value is also described in Section 4.5.4.

5.3.2 Qualitative Fitness Metrics

As already indicated, the quality and fitness of a layout solution is determined by a range of tangibles and intangibles. Indeed, terms like 'quality' and 'aesthetic value' are intrinsically subjective and prone to different interpretations by different people. The key to an understanding and judging aesthetics in layouts is to gain an understanding of what salient features are determining layout aesthetics in layouts. In this thesis, we describe a set of intuitively selected fitness metrics for gauging aesthetic contents in layouts.

To the best of our knowledge, no earlier study has compared layout design algorithms in terms of the ability to generate solutions with higher aesthetic contents. Consequently, there is a relative dearth of fitness metrics capturing the aesthetic value of a layout in an automated manner. It should be noted that some qualitative fitness metrics have been employed in some past studies, especially those involving evaluation of a computer interface design. However, such comparisons were realized using visual evaluation of the layout and experts' enumeration and estimation of the values of those metrics. In addition, such studies involved very small-scale problems and fitness values were calculated through visual evaluation and physical counting (Ngo & Law, 2003).

We deem automating the qualitative/aesthetic evaluation of a layout configuration as an intricate undertaking. One reason is that different criteria represent different perspectives and are very much likely to be competing and conflicting. Furthermore, various attributes often involve different and incommensurable measurement units (Aouni *et al.*, 2005; Triantaphyllou *et al.*, 1998; Triantaphyllou & Lin, 1996). Nevertheless, a key motivation in developing IDEAL is to minimize user inputs once the preliminary preferences have been identified. Consequently, some means of measuring these qualitative values of the layout in an automated, but swift, objective, and less resource-intensive manner is another crucial step in extending the research in the layout design automation (Ahmad *et al.*, 2003, 2004b; Head & Hassanein, 2002). Such an automatic evaluation of layouts requires quantifying highly subjective layout design guidelines, which come from experts' opinions that are subjective and uncertain.

We would like to point that some automated tools are available for fitness evaluation in some subjective domains somewhat related to layout design, for instance, e-Store structures (Head & Hassanein, 2002). However, such existing tools are far from being able to automate the process of qualitative fitness evaluation reasonably well (Ahmad *et al.*, 2003; Head & Hassanein, 2002; Ngo & Law, 2003). It is primarily due to the inability of such tools to incorporate many of the important but subjective and uncertain layout design guidelines and system related characteristics into the evaluation mechanism.

To further the research in this direction, we formulated and tested several empirical, nevertheless, informed qualitative measures of layout fitness. Our extensive testing resulted in narrowing down to four more effective and relatively robust qualitative fitness measures. Our preliminary studies have shown that these fitness metrics may be used in predicting expert's subjective rating of the layout reasonably well. For instance, layout fitness gauged by these metrics have a significant correlation with experts' subjective rating of the layout for a given layout optimization algorithm. Here we describe these qualitative fitness metrics.

It should be noted that, as a convention within the scope of this thesis, our discussions about the aesthetic contents of a layout configuration are with reference to the Enclosing Rectangle and not the packing space itself. Towards this end, the Enclosing Rectangle is divided into four quadrants and symmetry is determined by contrasting those four quadrants with each other. This turned out to be a good tactic for small- and medium-scale problems. However, it is not as much effective in large-scale problems because the probability of all four quadrants of Enclosing Rectangle being comparable in terms of following metrics becomes high. One possible solution strategy would be to divide the Enclosing Rectangle in more sections determined through some fuzzy rules using the number of modules and the size of Enclosing Rectangle. However, we have not implemented such a scheme, yet.

As such, our comparative analyses are based on partitioning the Enclosing Rectangle into four quadrants. Furthermore, we have tried to ensure that these normalized metrics always remain in the range of 0 to 1 (alternatively, 0% to 100%). However, this required us to have a large value of denominator in mathematical formulations of these metrics for normalization purposes. Such an approach resulted in less sensitive measures of symmetry. Employing a smaller denominator in these formulations made these very sensitive to aesthetic contents of the layout. However, such formulations occasionally resulted in negative values, contrary to the requirement of all normalized values lying in the range of 0 to 1. Consequently, we have tried to employ only mathematically appropriate formulations. However, subjectivity and uncertainty in gauging these measures of symmetry may be used as a rationale for having metrics that occasionally provide negative fitness values. In such a case, the fitness may be arbitrarily forced to 0.

Cohesion

Cohesion is the extent to which modules on each side of vertical and horizontal axes of a layout configuration have same aspects ratios (*AR*). Towards this end, we divide the Enclosing Rectangle into four quadrants and find out the difference between the maximum *AR* (*max* AR_k) and the minimum *AR* (*min* AR_k) in each quadrant *k*. If we denote this quantity as $cohesion\_D_k$ then it can be mathematically expressed as follows

$$cohesion\_D_k = \max AR_k - \min AR_k \cdots \{for\ the\ k^{th}\ Quadrant\} \qquad \text{Equation 5-3}$$

If we let the sum of all $cohesion\_D_k$ (for all four quadrants) be *cohesion_D* then:

$$cohesion\_D = \sum_{k=1}^{4}\{\max AR_k - \min AR_k\} \qquad \text{Equation 5-4}$$

Or

$$cohesion\_D = \sum_{k=1}^{4} cohesion\_D_k \qquad \text{Equation 5-5}$$

Now, a measure of total deviations (*cohesion_D_total*) can be found by using the sum of root mean square values of pairwise differences in $cohesion\_D_k$ as follows:

$$cohesion\_D\_total = \sqrt{\sum_{k=1}^{3}\sum_{l=k+1}^{4}\left(cohesion\_D_k - cohesion\_D_l\right)^2} \qquad \text{Equation 5-6}$$

We divide this measure of total deviation or *cohesion_D_total* by $\sqrt{3} * cohesion\_D$ to get a normalized measure of cohesion. The presence of $\sqrt{3}$ in the denominator arises from the fact that each quadrant is compared with three other quadrants in terms of cohesion to get the overall cohesion of the packing pattern:

$$cohesion = \left(1 - \frac{\sqrt{\sum_{k=1}^{3}\sum_{l=k+1}^{4}\left(cohesion\_D_k - cohesion\_D_l\right)^2}}{\sqrt{3} * \sum_{k=1}^{4}\left(cohesion\_D_k\right)}\right) * 100 \qquad \text{Equation 5-7}$$

This measure of *Cohesion* may vary from 1 to 100 with a higher value representing a higher degree of cohesion and better aesthetic contents.

Balance

Balance is defined as the difference between total weighting of components on each side of vertical and horizontal axes. A measure of how the weight of a page is distributed is very important in determining the aesthetical quality of a layout configuration. Towards this end, we divide the Enclosing Rectangle into four quadrants and find out the difference between the maximum Area (*max* $Area_k$) and the minimum Area (*min* $Area_k$) in each quadrant k. It represents the difference of size between the largest and the smallest module in the k^{th} quadrant of the Enclosing Rectangle. If we denote this quantity as $balance\_D_k$ then it can be mathematically expressed as follows:

$$balance\_D_k \quad = \quad \max Area_k - \min Area_k \qquad \text{Equation 5-8}$$

Let the sum of all $balance\_D_k$ be $balance\_D$ then:

$$bal\_D \quad = \quad \sum_{k=1}^{4} \{\max Area_k - \min Area_k\} \qquad \text{Equation 5-9}$$

Or

$$balance\_D \quad = \quad \sum_{k=1}^{4} balance\_D_k \qquad \text{Equation 5-10}$$

Now, a measure of total deviations $balance\_D\_total$ can be found by using the following sum of root mean square values of pairwise differences in $balance\_D_k$ as follows:

$$balance\_D\_total \quad = \quad \sqrt{\sum_{k=1}^{3} \sum_{l=k+1}^{4} \left(balance\_D_k - balance\_D_l\right)^2} \qquad \text{Equation 5-11}$$

We divide this measure of total deviation or $balance\_D\_total$ by $\sqrt{3} * balance\_D$ to get a normalized measure of balance. The presence of $\sqrt{3}$ in the denominator arises from the fact that each quadrant is compared with three other quadrants in terms of balance to get the overall measure of balance of the packing pattern:

$$Balance \quad = \quad \left(1 \; - \; \frac{\sqrt{\sum_{k=1}^{3} \sum_{l=k+1}^{4} \left(balance\_D_k \; - \; balance\_D_l\right)^2}}{\sqrt{3} \; * \sum_{k=1}^{4} \left(balance\_D_k\right)}\right) * 100 \qquad \text{Equation 5-12}$$

This measure of *Balance* may vary from 1 to 100 with higher value representing a higher degree of balance and better aesthetic value of the layout.

Distribution/Count

Distribution is the extent to which modules are equally divided, or distributed, in a layout design. This symmetry of distribution depends on the count of modules present in each quadrant of the Enclosing Rectangle. Towards this end, we find out the pairwise difference or deviation between the number of modules present in a quadrant k ($Count_k$) and the number of modules present in a quadrant l ($Count_l$) using the following mathematical expression:

$$Count\_D_{k,l} \quad = \quad Count_k - Count_l \qquad \text{Equation 5-13}$$

Now, a measure of total deviations in all quadrants *Count_D_total* can be found by using the sum of root mean square values of pairwise differences in $Count\_D_{k,l}$ as follows:

$$Count\_D\_total \quad = \quad \sqrt{\sum_{k=1}^{3} \sum_{l=k+1}^{4} \left(Count_k - Count_l\right)^2} \qquad \text{Equation 5-14}$$

We divide this measure of total deviation or *Count_D_total* by $\sqrt{3} * \sum_{k=1}^{4}\left(Count_k\right)$ to get a normalized measure of balance. The presence of $\sqrt{3}$ in the denominator arises from the fact that each quadrant is compared with three other quadrants in terms of balance to get the overall measure of balance of the packing pattern:

$$Count \quad = \quad \left(1 \quad - \quad \frac{\sqrt{\sum_{k=1}^{3} \sum_{l=k+1}^{4} \left(Count_k - Count_l\right)^2}}{\sqrt{3} * \sum_{k=1}^{4}\left(Count_k\right)}\right) * 100 \qquad \text{Equation 5-15}$$

This measure of *Count* could vary from 1 to 100 with higher value representing a higher degree of uniformity in the distribution of modules and, in turn, a better aesthetic value of the layout.

Density

Density is the extent to which the percentage of module area on entire layout configuration is uniform. Towards this end, we divide the Enclosing Rectangle into four quadrants and find out the pairwise difference or deviation between the sum of areas of modules in a quadrant k (AQ_k) and sum of areas of modules in a quadrant l (AQ_l) using the following mathematical expression:

$$Density\_D_{k,l} \quad = \quad AQ_k - AQ_l \qquad \text{Equation 5-16}$$

Now, a measure of total deviations in all quadrants *Density_D_total* can be found by using the following sum of root mean square values of pairwise differences in *Density_D*$_{k,l}$ as follows:

$$Density\_D\_total \quad = \quad \sqrt{\sum_{k=1}^{3} \sum_{l=k+1}^{4} \left(AQ_{k,l}\right)^2} \qquad \text{Equation 5-17}$$

Or

$$Density\_D\_total \quad = \quad \sqrt{\sum_{k=1}^{3} \sum_{l=k+1}^{4} \left(AQ_k - AQ_l\right)^2} \qquad \text{Equation 5-18}$$

We divide this measure of total deviation or *Desity_D_total* by $\sqrt{3} * \sum_{k=1}^{4} AQ_k$ to get a normalized measure of cohesion. The presence of $\sqrt{3}$ in the denominator arises from the fact that each quadrant is compared with three other quadrants in terms of cohesion to get the overall cohesion of the packing pattern:

$$Density \quad = \quad 100 * \left(1 \quad - \quad \frac{\sqrt{\sum_{k=1}^{3} \sum_{l=k+1}^{4} \left(AQ_k - AQ_l\right)^2}}{\sqrt{3} * \sum_{k=1}^{4} AQ_k} \right) \qquad \text{Equation 5-19}$$

This measure of *Density* could vary from 1 to 100 with higher value representing a higher degree of uniformity in the amount of space occupied by modules and, in turn, a better aesthetic value of the layout.

Effectiveness of Qualitative Metrics

In order to determine the effectiveness of these qualitative fitness metrics in gauging the aesthetic contents, we used 10 random sequences of the 100-module problem (A100) with MERA algorithm. The resultant layouts were subjectively rated by a couple of expert for the layout quality or aesthetic contents on a scale of 1-10 with a higher rating representing higher perceived aesthetic value of a given layout. We want to reiterate that a rating of 10 represents a highly symmetric layout topology, which cannot usually be achieved for problems consisting of randomly generated unequal modules or when modules dimensions have a high degree of variability. These experts have decades long experience of researching and practicing in facilities layout design applications. It should be noted that these experts had no knowledge of algorithms or fitness metrics used for generating these layouts. They were simply asked to rate these layout based on aesthetic value of layouts. There was no time constraint imposed on experts for providing their rating of these layouts. Results from this exercise are summarized in Table 5-1. In Table 5-1, the first column shows the iteration number. The second column presents the mean of experts' subjective rating for each alternative. The third column shows the *Cohesion* of the layout as measured by Equation 5-7. Similarly, the fourth column shows the *Density* as calculated by Equation 5-19. The fifth column shows the *Count* as calculated by Equation 5-15. The sixth column shows the sum of *Cohesion*, *Density*, and *Count*. The bottom-most row in Table 5-1 shows the coefficient of correlation (r) between expert's rating and the corresponding measure of symmetry. It can be seen that these measures of symmetry have high correlation with expert's rating. For instance, the coefficient of correlation between *Cohesion* and expert's rating is 0.46. Similarly, the coefficient of correlation between expert's rating and *Density* is 0.77. Likewise, the coefficient of correlation between expert's rating and *Count* is 0.83.

Iter#	Experts' Rating	(a) Cohesion	(b) Density	(c) Count	Sum (a)+(b)+(c)
1	6.0	55.31	94.07	86.21	235.59
2	7.0	67.55	96.17	92.27	255.99
3	6.5	64.55	94.89	85.88	245.32
4	6.5	80.03	97.64	87.46	265.13
5	5.0	61.34	80.26	73.09	214.69
6	2.0	61.32	69.84	59.06	190.22
7	5.5	58.41	92.30	81.92	232.63
8	5.0	66.87	91.59	83.99	242.45
9	4.0	47.23	94.31	86.25	227.19
10	4.5	56.80	96.22	83.97	236.99
Coefficient of Correlation (r)		**0.46**	**0.77**	**0.83**	**0.89**

Table 5-1: Ten random iterations of MERA with expert's rating & measures of symmetry (**A100**)

Interestingly, the coefficient of correlation between expert's rating and the sum of the three measures of symmetry is higher than correlation between expert's rating and individual measures of symmetry. It demonstrates that these measures of symmetry together provide a rational regime for gauging aesthetic contents of the layout. Indeed, the provision of assigning different preference weights to various measures of symmetry provides a means for adapting the fitness regime to decision maker's preferences. In the next Section, we describe how preference weights produced by PIA may be used to facilitate multi-criteria decision-making in ILG for generating superior layout alternatives, as described in Chapter 4.

5.4 Fuzzy Multi-Criteria Decision Making

Here we describe some Multi-Criteria Decision Making (MCDM) schemes for use in ILG in generating superior layout alternatives. Indeed, an MCDM model has always been considered as more pragmatic, though seldom practiced, by researchers (Ahmad *et al.,* 2004b, 2004c, 2004d; Armour & Buffa, 1963; Bazaraa, 1975; Blair & Miller, 1985; Dowsland *et al*., 2002; Ligget, 2000; Tompkins *et al*., 2002; Triantaphyllou & Lin, 1996; Youssef *et al*., 2003). In the presence of a large number of decision attributes, some hierarchical arrangement regime may be employed. One popular an effective scheme is Analytic Hierarchy Process or AHP (Ahmad, 2002; Triantaphyllou *et al.*, 1998).

One big obstacle in realizing an MCDM model in layout optimization is the scarcity of efficient, encompassing, and robust fitness metrics that could be combined in the form of some hybrid fitness model to facilitate MCDM. Furthermore, such MCDM approaches are mired not only by conflicting objectives but also by difference in the units of measurement (Triantaphyllou *et al.*, 1998). Consequently, the subjectivity, uncertainty, and incommensurable units render MCDM paradigm inherently difficult to realize in layout analysis and design.

In order to address this shortcoming, we have developed several quantitative and qualitative fitness metrics and culled those that have been demonstrated effective and robust through studies with a variety of large-scale problems. These fitness metrics are discussed in Section 4.5.4 as well as Section 5.3. Here we propose a Fuzzy MCDM (f-MCDM) approach for combining such fitness metrics into a single hybrid fitness function. The ability of FL to realize complex non-linear input-output relationships as a synthesis of multiple simple input-output relations proves invaluable in this regard (Ahmad *et al.*, 2003, 2004b, 2004c; Karray & de Silva, 2004; Negnevitsky, 2002). We want to emphasize that such Fuzzy MCDM approaches are best used as decision-aid tools and not as decision-making tools. Individual tendencies and intuitions of different decision makers are likely to result in different solutions.

5.4.1 Fuzzy Weighted-Sum Model

In a Fuzzy Weighted-Sum Model (FWSM), the total utility or fitness of a layout alternative is calculated by adding up the product of fitness value f_κ of an individual attribute κ and the priority weight or significance S_κ assigned to that attribute. Mathematically, it can be represented by the following Equation:

$$F_{FWSM} = \sum_{\kappa=1}^{p} S_\kappa f_\kappa \qquad \text{Equation 5-20}$$

FWSM is the earliest and probably the most popular and easily amenable approach that has its roots in Utility Theory. The assumption that governs the FWSM is the *additive utility*. FWSM is appropriate for single-dimensional cases where the units of measurements are identical. However, in multi-dimensional cases where are units not commensurable, a conceptual violation occurs because of the usual assumption of additivity of utilities. Nevertheless, this conceptual violation can be overcome by normalizing fitness values for all attributes against some suitable benchmarks.

5.4.2 Fuzzy Normalized Weighted-Sum Loss Function

Here we propose a novel approach to f-MCDM for multi-dimensional multi-attribute decision problems, in general, and layout decision analysis, in particular. Our approach draws from the relative simplicity of FWSM and efficacy of relative fitness values (as in AHP). It is inspired by Taguchi's quality loss function where any deviation from the nominal values results in a loss or reduction in utility depending upon the amount of deviation (Taguchi *et al.*, 1989). Accordingly, our approach involves employing the normalized values of principal layout fitness metrics and calculating the deviation from some preferred nominal values. This deviation, in turn, is used to calculate penalties based on the weight or significance S_κ assigned to each fitness attribute κ. We term this approach as Fuzzy Normalized Weight-Sum Loss Function (f-NWSLF).

Conceivably, the selection of these benchmarks for normalization in such subjective and uncertain work domain as layout design remains a contentious issue and constitutes an open research question. As such, the benchmarks employed for normalizing each fitness dimension mentioned in Section 4.5.4 and Section 5.3 may be contended. However, the selection of these benchmarks was made after extensive preliminary studies with a range of intuitively selected benchmarks, which revealed these as satisficing benchmarks for our purposes.

In essence, the penalty function calculates the weighted sum of penalties, where weights are the significance S_κ assigned to a fitness attribute κ and penalty is the deviation of normalized fitness value $\hat{f}_\kappa$ from its preferred value P_κ. In this manner, we are combining the powers of three effective MCDM techniques. This penalty function may be made more or less precipice using a parameter $\psi > 1$ based on decision-makers' preferences. A value of $\psi > 1$ would result in a more precipice loss function, whereas a value of $\psi < 1$ would result in relatively flat loss function. It should be noted that if ψ is not a multiple of two then it requires the penalty function to be absolute deviation from $\hat{f}_\kappa$. However, currently we are using the penalty as proportional to the square of deviation (i.e. $\psi = 2$) in the following Equation:

$$F_{f-NWSL} = \sum_{\kappa=1}^{p} S_\kappa \left\{ \left| \hat{f}_\kappa - P_\kappa \right| \right\}^{\psi} \quad \text{Equation 5-21}$$

It should be noted that certain parameters could have significant interaction with one another affecting more than one value of crisp weights used subsequently in the layout evaluation phase. Consequently, as a future research direction, we intend to develop some mechanism through which PIA can handle such interactions and interdependencies. In addition, the question of developing more effective and robust layout fitness metrics remains open for further research in MCDM field. We expect our efforts would increase interest in this important research direction.

5.5 Working of Preference Inferencing Agent

In order to elaborate the working of the PIA, we consider a scenario where the small size of the packing space would not permit placement of all the given modules in the layout configuration. Such a scenario is quite common in practice. For instance, only a handful of modules may be displayed on one page of an e-Store. We consider the same 100-module (A100) problem used in Chapter 4. The difference is in the reduced dimensions of packing space that precludes the placement of all 100 modules.

In our example, the amount of 'white space' and the 'size of bin' affect the maximum number of 'bin modules' that could possibly be placed in a single bin or packing space. As indicated, this is an important parameter to be determined for the efficiency and efficacy of the whole process. For instance, it would affect the length of chromosome chosen for a GA used in the ILG. It has dramatic effect on efficiency and quality of results as it determines the search space in a GA. We considered a simple bin-packing problem in which the limited size of the bin might leave some modules outside the resultant layout. In such a situation, employing a chromosome size of 100, as was the case earlier, would result in unnecessarily slow progression of the GA based optimization process.

In our example, we let *x, y,* and *z* (*white_space, bin_size,* and *chromosome_size* respectively) be the linguistic variables; *A1, A2,* and *A3 (small, medium,* and *large*) be the linguistic values determined by fuzzy sets on the universe of discourse *X (white_space)*; *B1, B2, B3* and *B4 (small, medium, large* and *ex-large*) be the linguistic values determined by fuzzy sets on the universe of discourse *Y (bin_size)*; *C1, C2,* and *C3 (small, medium,* and *large*) be the linguistic values determined by fuzzy sets on the universe of discourse *Z (chromosome_size)*. The membership functions for these linguistic variables are shown in Figure 5-5. The complete set of fuzzy rules for determining *choromosome_size* using *white_space* and *bin_size* is provided in Table 5-2.

		Bin Size			
		Small (B1)	Medium (B2)	Large (B3)	Ex-Large (B4)
White Space	Small (A1)	*Small*	*Small*	*Medium*	*Medium*
	Medium (A2)	*Small*	*Medium*	*Medium*	*Large*
	Large (A3)	*Medium*	*Medium*	*Large*	*Large*

Table 5-2: Fuzzy Rules for determining the Chromosome Size.

Our example consists of a simple two-input and one-output scenario involving the following two fuzzy rules specified by an expert:

Rule 1:

If *x* is *A2* (*white_space* is *medium*)

Or *y* is *B3* (*bin_size* is *large*)

Then *z* is *C2* (*chromosome_size* is *medium*)

Rule 2:

If *x* is *A3* (*white_space* is *large*)

Or *y* is *B4* (*bin_size* is *ex-large*)

Then *z* is *C3* (*chromosome_size* is *large*)

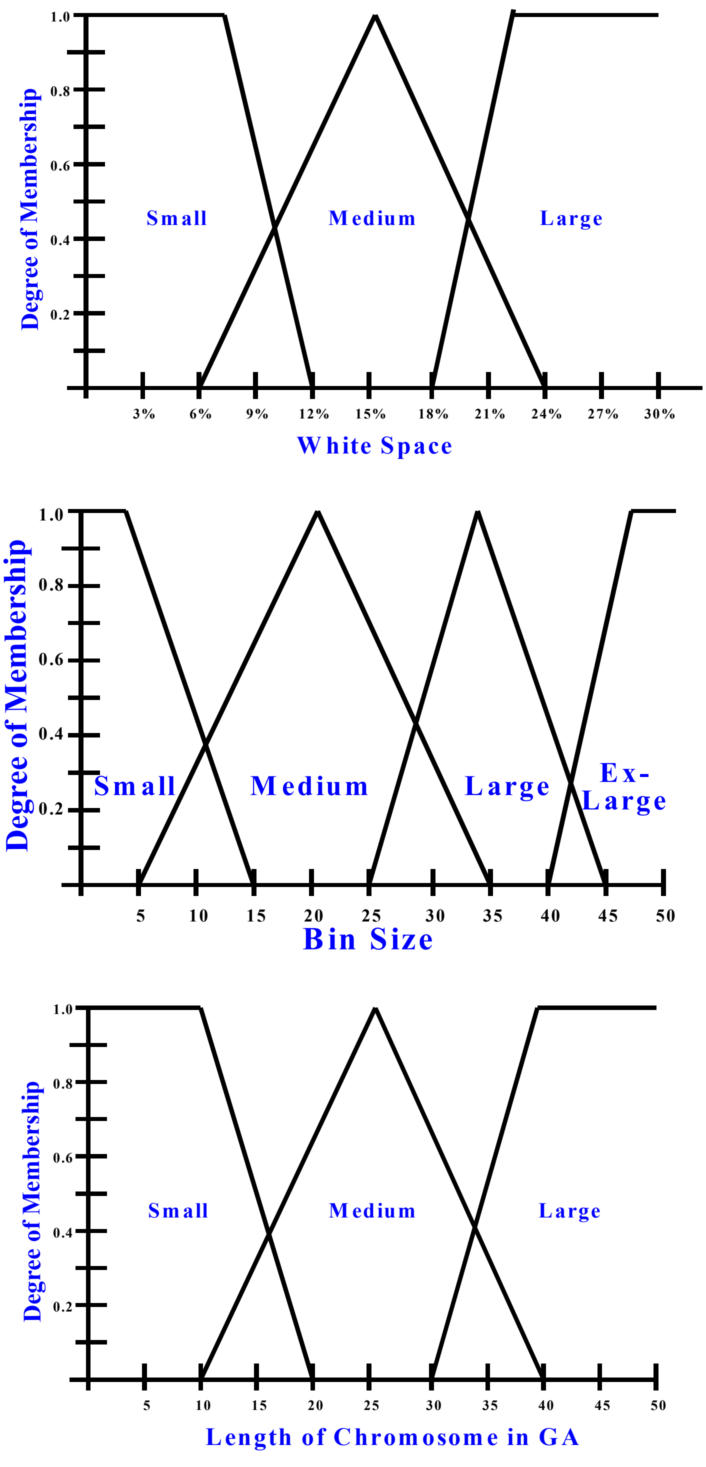


Figure 5-5: Fuzzy Sets For 'White Space', 'Bin Size' and 'Chromosome Size'.

As a first step, we *fuzzified* all the crisp inputs and determined the degree to which these inputs belong to each of the appropriate fuzzy sets. The crisp input *x1* (*white_space* rated by experts as 20%) corresponds to the MF *A2* and *A3* (*medium* and *large*) to the degrees of 0.6 and 0.2, respectively. Likewise, the crisp input *y1* (*bin_size* rated as 44 units) corresponds to the MF *B3* and *B4* (*large* and *ex-large*) to the degrees of 0.15 and 0.5, respectively. The *rule evaluation* involved applying the fuzzified inputs to antecedents in the fuzzy rules. Here we used the *min* operator to evaluate the fuzzy *OR* operation and the *max* operator to evaluate the fuzzy *AND* operation, respectively. This resulted in the following degrees of memberships:

$$\mu_{C2}(z) = \max[\mu_{A2}(x)\ ,\ \mu_{B3}(y)] = 0.6$$
$$\mu_{C3}(z) = \min[\mu_{A3}(x)\ ,\ \mu_{B4}(y)] = 0.2$$

The result of antecedent evaluation is applied to the MF of the consequent by 'clipping' the consequent MF to the level of the truth-value of the rule antecedent. The *Aggregation* involved unification of the outputs of all rules. Here, we used the clipped consequent MF. This way we evaluated the fuzzy rule for selecting the chromosome size in our layout problem. However, the final output of the PIA needs to be a crisp number for use in some GA parameter or fitness function. The most popular *defuzzification* technique is the 'centroid' technique where a vertical line carves the aggregate fuzzy set into two equal masses. Using the Mamdani technique in the given example, the crisp value for the *chromosome_size* comes out to be about 27. This whole inferencing mechanism is summarized in Figure 5-6. In this manner, the ILG could be adapted in terms of *chromosome_size* based on preferences furnished by experts/users through PIA.

In order to evaluate the effect of the *chromosome_size* as determined by the PIA, we ran 1000 iterations of the GA with a chromosome size of 100 as well as 1000 iterations with a chromosome size of 27. In this study, we employed the MERA as a decoding heuristic. The computer system used was Intel Xeon 3.06 GHz processor with 256 MB of RAM running under Windows XP. The average time per GA iteration with a chromosome size of 100 was 15.43 seconds. In contrast, the average time per GA iteration with a chromosome size of 27 was only 0.316 seconds. It elaborates how a simple adaptation of a GA parameter through fuzzy rules and inferencing could affect the efficiency of the overall process. Furthermore, this example illustrates how vague linguistic rules can be used to derive important and useful crisp values. Likewise, the PIA can be used to furnish other parameters for subsequent use. Our preliminary studies show that fuzzy logic constitutes an effective inferencing tool in layout design. It provided greater flexibility, expressive power, and ability to model vague preferences.

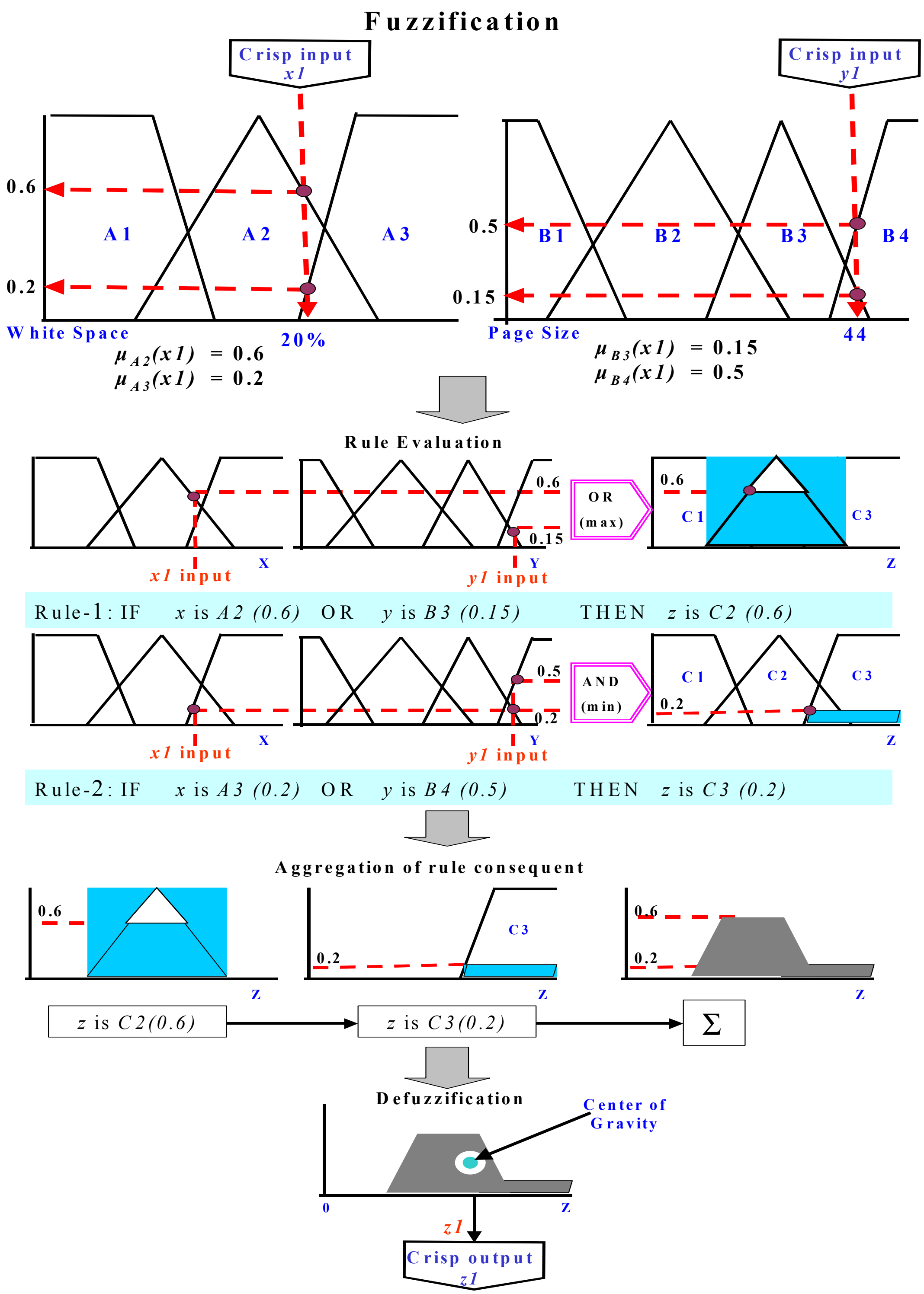


Figure 5-6: Example of Mamdani style Fuzzy Inferencing in Layout Design.

5.6 Discovering User Preferences

The reliability and effectiveness of PIA significantly depends on the reliability of preferences. In this regard, the implicit and dynamic nature of preferences as well as efforts required for building and updating an expert system underscore the need for automated learning. Indeed, learning is an important constituent of any intelligent system (Negnevitsky, 2002).

Although IDEAL permits users to enter preferences explicitly, we envision employing such ML tools as Artificial Neural Networks (ANN) and Reinforcement Learning (RL) for automated and self-updated acquisition of knowledge. An automated LD fitness rating could form a basis for discovering implicit preferences of a user and result in creation of user profiles (Webb, 2001). Such profiles would facilitate personalized generation of superior alternatives for individual decision-makers. For instance, ILG could furnish alternatives that are more likely to be rated highly by the user with higher probability. Indeed, such automated knowledge acquisition and revision would instill another aspect of intelligence into IDEAL, namely, learning and evolution of knowledge. We want to make it explicit that this discussion does not imply that machine learning would eliminate the need for experts' opinion or would enable automated discovery of all or most of the rules and preferences. However, an intelligent PDA in tandem with other knowledge-acquisition approaches could increase the effectiveness and robustness of the system.

5.6.1 Neuro-Based Expert Systems

Our vision of an ANN based PDA is motivated by the success of this powerful technology in improving the robustness of traditional rule-based expert systems (RBES). Indeed, the neuro-based expert systems (NBES) neatly complement the capabilities of RBES. In the presence of high-level expertise, RBES perform better because of a higher degree of utilization of expertise. However, when level of expertise available is low but the number of test examples is large then NBES is more appropriate, enabling capturing of some of the design knowledge hidden in the given data. Such interrelationship is depicted in Figure 5-7. The tedium involved in knowledge acquisition, difficulty of knowledge base modification, and inability to incorporate learning changes means that an innovative and synergistic combination of RBES and NBES could be useful (Chung, 1999; Ruan, 1997; Yasdi, 2000; Zha & Lim, 1999, 2001).

NBES provides capability to analyze and capture highly nonlinear complex dynamics of ill-structured problems through its facility to learn from examples and real or simulation experiments. In

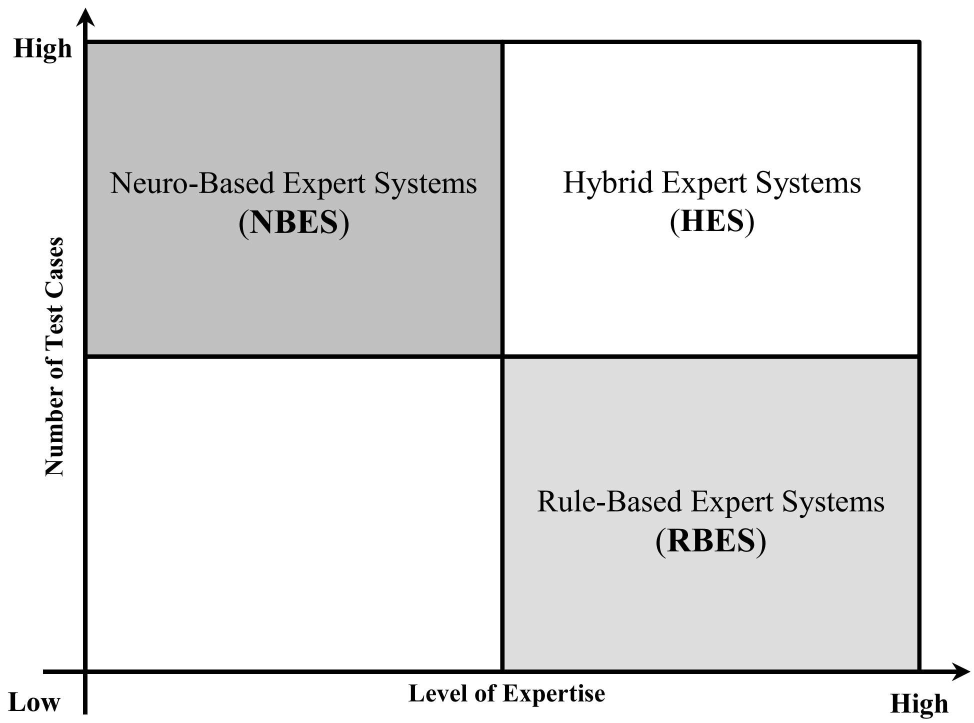


Figure 5-7: RBES Vs NBES – Level of Expertise and Number of Test Instances.

addition, the tolerance to imprecise and uncertain data and ability to generalize an approximate solution are valuable attributes in automated layout design. The popularity of ANN as a machine learning tool stems from its amazing versatility and ability to produce good results in complex domains (Negnevitsky, 2002). Conceivably, there have been attempts to tap on this powerful technology in the layout design applications in various modes. For instance, Tsuchiya *et al.* (1996) used ANN to directly solve a quadratic assignment problem involving *n* modules to be assigned to *n* potential locations. Likewise, Ilumoka (1997) used ANN models for simulation and optimization of VLSI circuits. Zha & Lim (1999, 2001) used neural computing for layout planning, design, and adjustments of a workbench. However, such studies involved very small-scale problems for demonstration purposes.

Nevertheless, NBES are not readily amenable to providing explanation facilities as an ANN has implicit weights, rather than explicit rules, for representing knowledge and preferences (Maleki-Dizaji *et al.*, 2003; Ruan, 1997). The detailed discussions on relative merits of RBES and NBES outlining their complementary capabilities is beyond the scope of this thesis and can be found in the literature (Chung, 1999).

Devising a formal description and implementation of layout aesthetics are deemed very difficult. It forms the basis for our desire to investigate the possibility of using ANN, and other ML tools, for the evaluation of layout aesthetics. However, making a prudent selection of a representative set of layouts suitable for an ANN based PDA, some knowledge on layout aesthetics needs to be acquired (Yasdi, 2000). We acquired such knowledge through interviews with two layout design practitioners. It enabled the choice of a few criteria that seem to neatly gauge layout aesthetics, as discussed in Section 5.3.2. However, our study of PDA is only exploratory in nature and relatively shallow. Our basic goal is the testing of the concept. As such, we elected to employ one qualitative measure of layout aesthetics and one quantitative measure of space utilization, for training of the PDA. The quantitative measure employed was Module Tightness and represents the first input (X_1) to the PDA. Whereas, the qualitative measure employed was Symmetry of Distribution and represents the input (X_2). The output of the PDA is the layout rating (Y), as shown in Figure 5-8.

5.6.2 Multi-Layer Perceptron based PDA

In order to test our concept, we used well-known Multi-Layer Perceptron Network (MLP). We employed a Feed Forward Multi-Perceptron ANN as we were able to generate a modest number of instances for training and testing. However, if there were relatively fewer number instances available then Bi-directional Associative Memory (BAM) or Reinforcement Learning (RL) might have been more appropriate for automated learning (Yasdi, 2000). BAM has incremental learning capabilities that inspired some research level prototypes and have reportedly shown promising results (Chung, 1997). Whereas, RL is based on the notion of a learning system that adapts its behavior in order to maximize some reward (Sutton & Barto, 1998).

In our PDA, we used a fully connected artificial neural network with one hidden layer, as depicted in Figure 5-8. The network consists of two input neurons, three hidden neurons, and a single output neuron forming a directed acyclic graph. The inputs to PDA consist of Module Tightness (X_1) and Symmetry of Distribution (X_2). Furthermore, the output of the PDA is the rating of the layout (Y) for the given inputs. The number of hidden nodes in a network is critical to the network performance. A neural network with too few hidden nodes can lead to underfitting and may not be able to learn a complex task, while a neural network with too many hidden nodes may cause oscillation, overlearning/memorization, and hamper the ability for generalization (Biron, 1999; Nauck *et al.* 1997; Negnevitsjy, 2002). As such, after some initial experimentation, we deemed a hidden layer

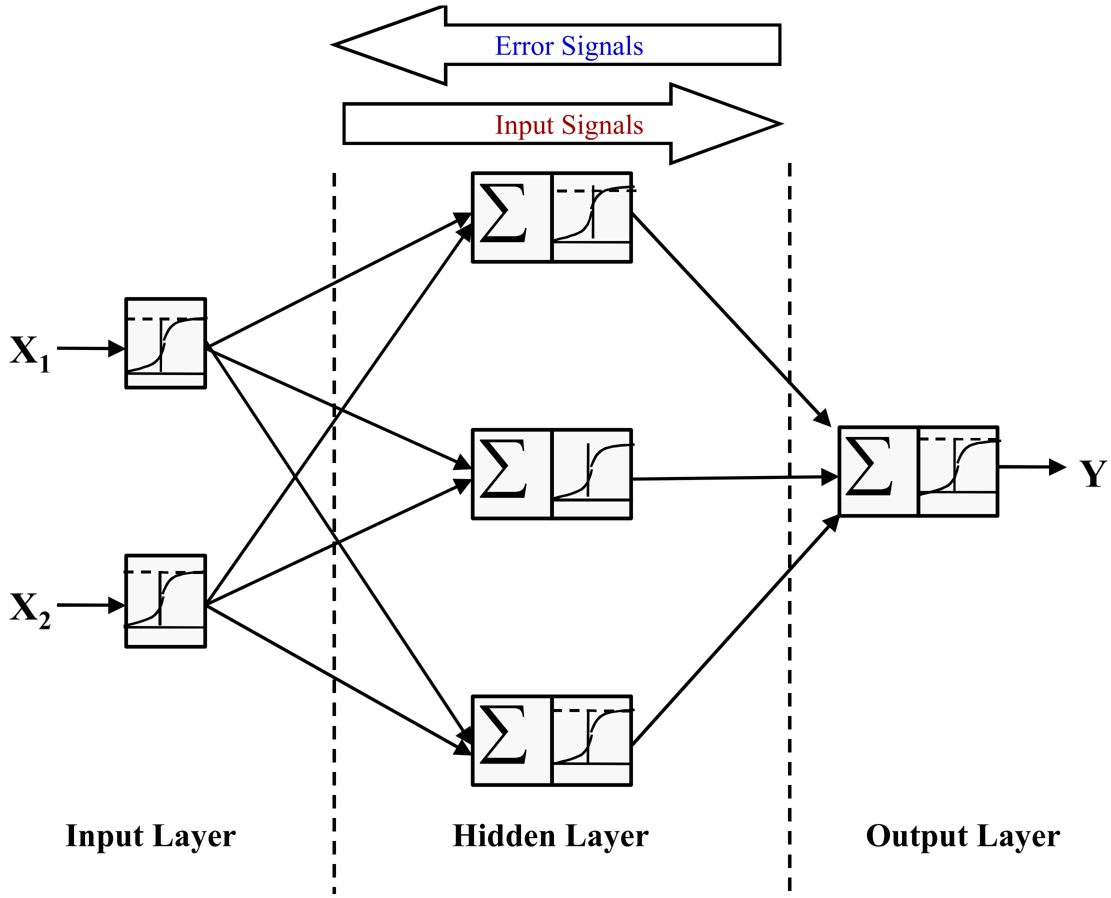


Figure 5-8: Architecture of the Artificial Neural Network based PDA.

comprising of three neurons sufficient for our purposes. The decision on the architecture of an ANN is typically done through a trial-and-error process choosing the architecture leading to the best performance.

We used MATLAB to code our algorithm for training the PDA based on the popular back-propagation supervised learning paradigm. In this paradigm, the network can be trained by measurement data from the training set. It propagates the errors backwards by allocating them to each neuron in accordance to the amount of this error for which the neuron is responsible. The prediction capability of the trained network can be tested for some test data. The caveat in using the back-propagation algorithm and the MLP is that these require a large number of training examples.

Data Collection

In our PDA, we employed Module Tightness as the first input (X_1) and Symmetry of Distribution as the second the input (X_2) to the PDA. The output of the PDA is the layout rating (Y), as shown in Figure 5-8. We gave 80 layout alternatives, for a single test problem shown in Appendix E, to an expert who have more than 20 years experience in researching and practicing in layout design applications. We asked the expert to assign a single rating for each layout alternative based on space utilization and layout symmetry on a scale of 1-100. The expert had neither any knowledge of algorithms used to generate those alternatives nor the values of X_1 and X_2 as calculated by IDEAL. Furthermore, there was no time constraint imposed on the expert for providing those ratings. We used the layout fitness ratings provided by the expert as target outputs (T) for the ANN.

We employed 60 instances for training the PDA and another 20 instances for testing the PDA. We tried to select training and test examples that are representative of the entire spectrum, listed in Appendix E. It should be noted that we have not carried out extensive experiments in fine-tuning ANN parameters or conditioning the training data. The optimization of a given ANN is usually a tedious task. Our aim in this exploratory study is to test the concept and gain more insights in this research direction for future extensions of our work.

5.6.3 Results and Insights

We employed the popular Mean Square Error (MSE) as a measure of performance or convergence. We used a learning rate of 0.01 and programmed to terminate the training of the network after 50,000 epochs or when Absolute MSE goes below 0.001, whichever occurs first. We generated a random permutation of training data set before proceeding to the training of the PDA. Furthermore, we scaled PDA inputs (X_1 and X_2) and target values (T) in the [0,1] range. As such, the PDA outputs (Y) are also obtained as scaled values in the [0,1] range. The convergence of PDA's training is shown in Figure 5-9. The convergence characteristics of an ANN may be described by the ability of ANN to converge to specific error levels (Yasdi, 2000). These figures demonstrate that the PDA has a sound convergence capability. For visual comparison purposes, we have shown the Pattern Error that is calculated as the difference between the target value and the actual output for the training set of PDA in Figure 5-10, with both T and Y scaled to a range of [0,1]. It can be seen that the pattern error remains reasonably low. In addition, we have shown the Pattern Error calculated as the difference between the target value and the actual output for the test set, scaled to a range of [0,1], of PDA in Figure 5-11. Once again, the pattern error remains reasonably low, revealing the capability of PDA to learn and generalize from the given training instances.

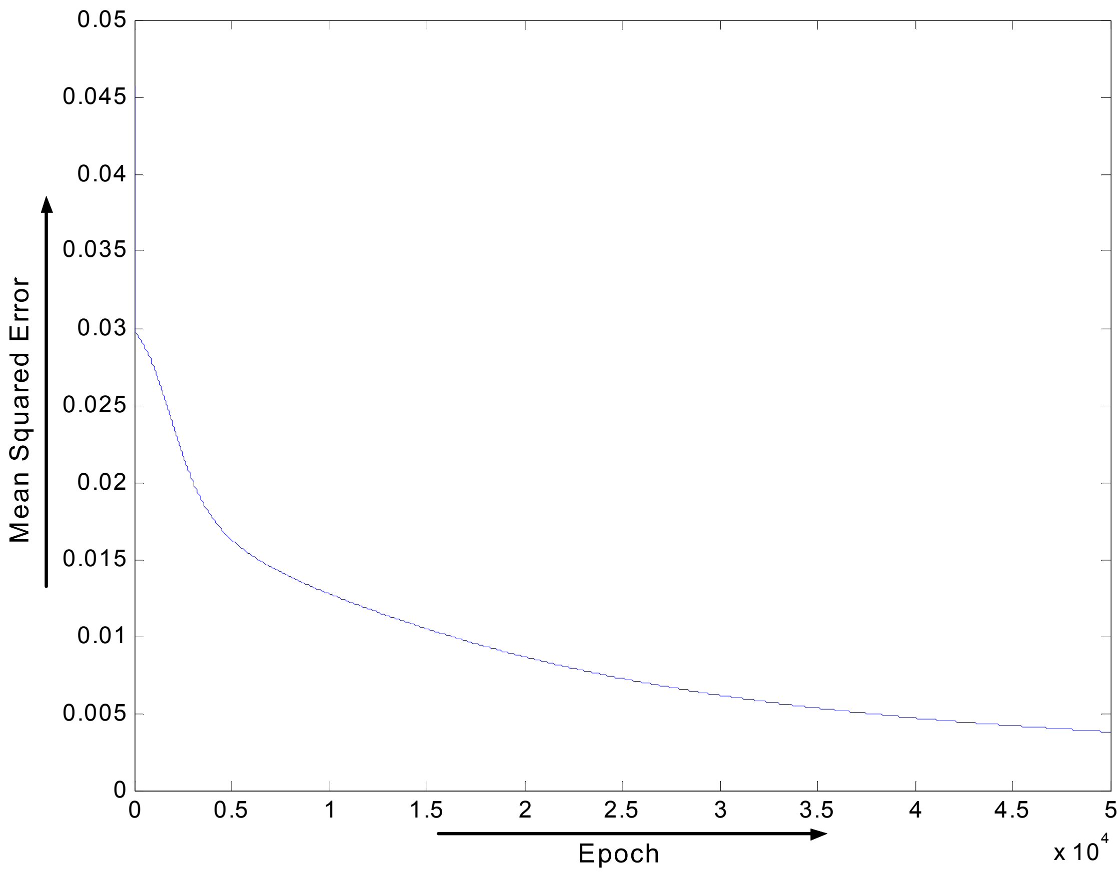


Figure 5-9: Convergence of the Training Phase of the PDA.

The trained network had an MSE of 0.00367 for values scaled in the range [0,1]. Similarly, the MSE obtained with the test data is 0.002486. Considering low MSE for the training and test data sets, we consider the PDA to be reasonably trained with reasonable generalization capability. Furthermore, the ratio between the test MSE and the training MSE, commonly known as generalization factor, is 0.67738, which is desirable as it indicates that test error was less than the training error. Nevertheless, we have shown the Actual and Target network outputs, without scaling, in Table 5-3 and Table 5-4. The mean absolute error (T-Y) for training data is 4.86 and for test data is 4.24, which also implies promise of generalization capability. However, detailed discussions on the tradeoff between training and generalization errors are beyond the scope of this thesis and may be found in the literature (Nauck *et al.*, 1997; Negnevitsky, 2002).

#	Inputs		Target	Output	#	Inputs		Target	Output
	X1	X2	T	Y		X1	X2	T	Y
1	70.8	78.75	**60**	**59.61**	31	74.3	82.21	**70**	**68.11**
2	85.7	73.8	**65**	**60.71**	32	90.2	70.54	**60**	**57.31**
3	70.6	69.38	**35**	**40.46**	33	83	78.37	**75**	**67.27**
4	75	75.1	**50**	**55.61**	34	70.6	69.87	**30**	**41.55**
5	75	69.18	**35**	**43.21**	35	70.6	68.3	**30**	**38.10**
6	80	75.12	**60**	**59.05**	36	74.3	81.26	**70**	**66.40**
7	80	71.74	**50**	**52.26**	37	84.2	77.18	**70**	**65.82**
8	85.7	65.14	**35**	**42.39**	38	70.6	75.75	**55**	**53.53**
9	85.7	63.4	**35**	**38.53**	39	78.9	80.19	**75**	**67.71**
10	85.7	65.14	**40**	**41.90**	40	83	83.3	**80**	**74.90**
11	75	70.95	**45**	**46.54**	41	70.2	87.14	**75**	**73.28**
12	75	71.74	**45**	**48.17**	42	78.4	92.8	**70**	**83.67**
13	85.7	59.17	**30**	**29.73**	43	74.3	73.85	**50**	**52.61**
14	80	66.91	**30**	**41.68**	44	74.1	85.57	**70**	**73.17**
15	80	71.74	**55**	**51.66**	45	83	84.41	**85**	**76.29**
16	80	72.36	**55**	**53.01**	46	83	90.52	**85**	**83.21**
17	85.7	65.14	**40**	**41.9**	47	83	90.52	**90**	**83.22**
18	84.2	72.65	**60**	**56.64**	48	78.4	77.51	**65**	**62.86**
19	90.2	71.74	**75**	**59.01**	49	88.2	80.2	**75**	**73.30**
20	90.2	76.69	**70**	**68.39**	50	78.4	84.41	**80**	**74.10**
21	84.23	67.2	**45**	**45.63**	51	83	89.79	**85**	**82.64**
22	90.2	68.31	**55**	**52.31**	52	83	79.26	**70**	**69.02**
23	84.2	77.18	**65**	**65.63**	53	78.4	85.72	**85**	**75.97**
24	74.3	70.89	**40**	**46.06**	54	83	81.71	**75**	**72.94**
25	90.2	70.53	**65**	**56.84**	55	78.9	80.1	**65**	**68.02**
26	90.2	70.9	**65**	**57.75**	56	85.7	60.47	**25**	**33.28**
27	84.2	81.26	**85**	**72.42**	57	78.9	70.95	**45**	**50.35**
28	90.2	80.1	**85**	**74.07**	58	75	73.65	**50**	**52.90**
29	78.9	68.82	**45**	**45.72**	59	88.2	86.82	**80**	**81.57**
30	78.9	77.11	**50**	**62.47**	60	75	71.1	**40**	**47.52**

Table 5-3: Contrast between the Expert's Rating and the PDA Output (Training Data)

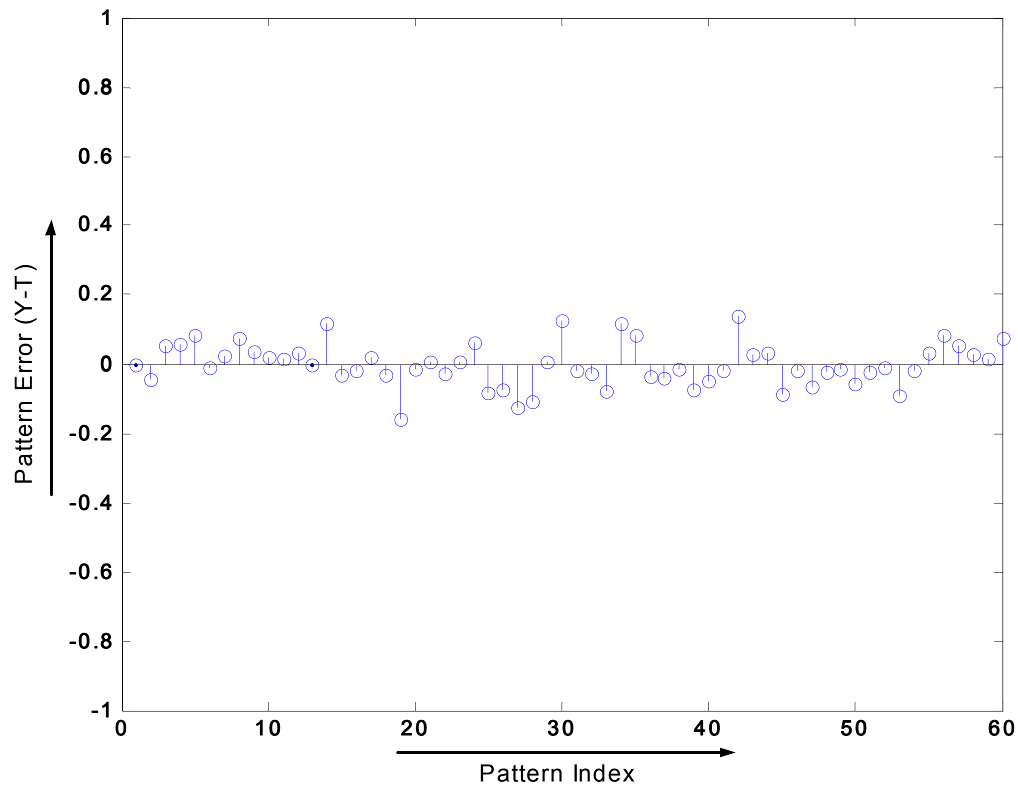


Figure 5-10: The Pattern Error for Training Data - Actual Output minus Scaled Target Value

#	Inputs		Target	Output	#	Inputs		Target	Output
	X1	X2	T	Y		X1	X2	T	Y
1	78.4	72	**40**	**49.27**	11	75	75.75	**50**	**56.91**
2	70.6	70.92	**40**	**43.59**	12	74.1	78.55	**65**	**61.62**
3	70.6	65.09	**25**	**31.68**	13	74.1	88.7	**80**	**77.33**
4	90.2	79.92	**60**	**58.09**	14	83.3	78.37	**75**	**67.43**
5	74.3	77.11	**55**	**59.05**	15	88.2	89.79	**90**	**84.36**
6	84.2	69.51	**45**	**50.94**	16	85.7	60.91	**30**	**33.76**
7	84.2	63.26	**35**	**37.60**	17	90.2	74.47	**65**	**64.89**
8	78.9	76.42	**60**	**61.02**	18	78.9	80.1	**70**	**67.58**
9	78.4	79.3	**75**	**65.90**	19	88.2	71	**60**	**56.88**
10	78.9	78.75	**70**	**65.27**	20	84.2	83.3	**75**	**75.45**

Table 5-4: Contrast between the Expert's Rating and the PDA Output (Test Data)

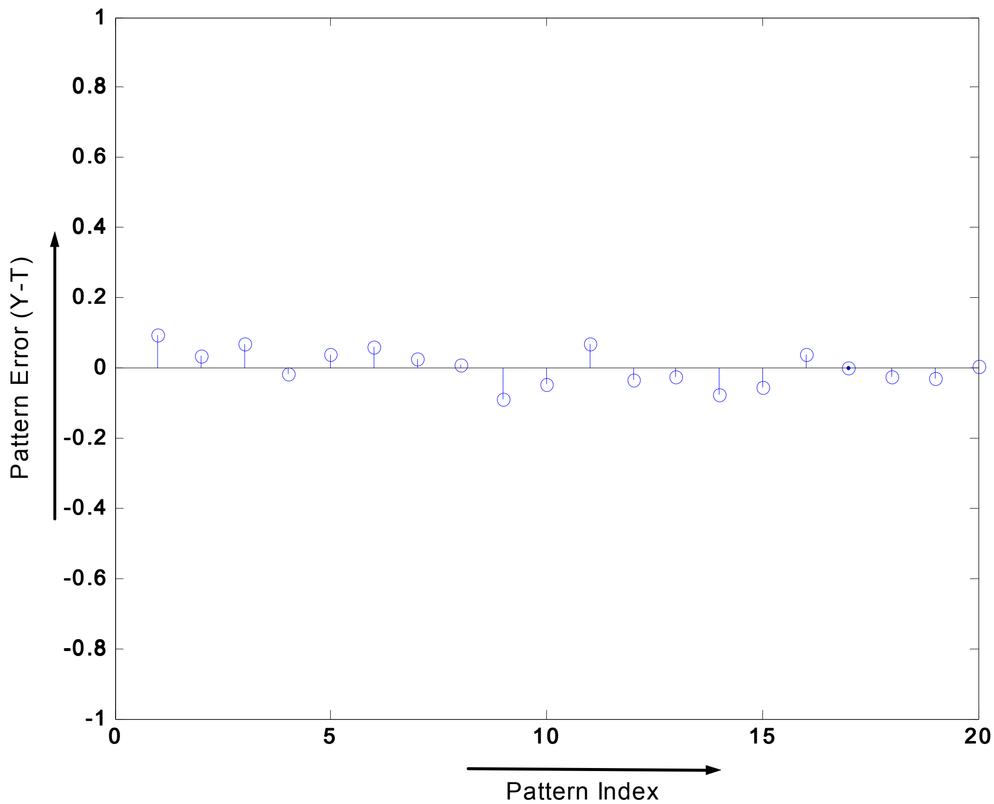


Figure 5-11: The Pattern Error for Test Data - Scaled Target minus Scaled Actual Output

It should be borne in mind that the PDA is geared towards predicting and modeling human expert's rationalization and behavior. Indeed, modeling and predicting human behavior is a very challenging venture and imperfection in predicting human behavior makes such levels of error inevitable. Furthermore, the number of training instances is only modestly sufficient for such complex work domains as layout design. Consequently, we deem the output of PDA maps to the target output reasonably well. However, this exploratory study limited from several perspectives, as discussed below.

Limitations

This exploratory study has several limitations. For instance, training and test data were the layout alternatives generated for the single problem. However, using test data from various other problems would have provided a better assessment of generalization capability of the PDA. In addition, we did not employ any separate validation data set to ensure the MSE calculated during training also indicate good generalization capability and not mere overlearning of the PDA output to the target values.

Furthermore, as an empirical rule, the number of training instances should be about 10 times the number of weights being trained in the ANN. Indeed, in the current setup, we have about 11 training weights against only 60 training instances. The number of training instances is quite inadequate and may have resulted in overfitting to the training data, jeopardizing the much sought for generalization capability. However, these limitations are a direct consequence of limited data availability, as a large number of examples might have resulted in lost interest from the expert.

Another limitation of the study emanates from the *framing* of the question posed to the expert who rated those layouts alternatives. Incidentally, we asked the expert to rate the given layouts as 'alternatives', which tacitly meant that each solution may be treated as a starting point for further improvement and hence rated according to the expected utility after refinement rather than the inherent utility of a given layout alternative. Conceivably, a touch of intuition and foresight was involved in such ratings. A careful study of the data set we generated supports this hypothesis. Some representative cases of such trends have been elaborated in Section 4.9. In general, the expert gave higher ratings to layouts that can easily be modified to superior outcomes and are not merely inherently superior.

However, offering those layout solutions as 'final solutions' and taking away any explicit or implicit notion of some possibility of fine-tuning might have resulted in different layout ratings and PDA training outcome. Nevertheless, it was not easy to carry out such experiments due to the degree of involvement required from experts. Such experiments are time consuming and if carried out for long periods then experts may lose interest in the whole exercise. In general, the enthusiasm shown in the beginning by participants of such studies does not last long, as it quickly becomes a repetitive and boring exercise for them.

In short, the evaluation of results shows that predicting a decision-maker's behavior provides a rough albeit indispensable guideline for determining an appropriate regime for decision alternative selection to be furnished to the decision-maker. It is argued that further improvements can be attained through some extensive experimentation and optimization of ANN as well as through some content-based reinforcement learning process.

5.7 Summary

In this Chapter, we have provided our philosophy and implementation of a Preference Inferencing Agent. We have described a framework for fuzzy inferencing that could furnish crisp weights to a hybrid fitness function as well as various parameters for the ILG. Furthermore, we provided the conceptual basis for an automated Preference Discovery Agent as well as some simulation results to demonstrate the viability of the concept. In the subsequent chapter, we conclude with contributions and insights gained through this thesis. In addition, we describe some limitations of this research. Moreover, we provide several interesting future research directions.

Chapter 6

CONCLUSION

6.1 Introduction

In layout design, the solution elements are modules, and solutions are combinations of modules that form a complex configuration satisfying certain physical constraints and aesthetic considerations. However, such complex problems require the ability to construct and revise plans more flexibly from primitive actions or procedures.

Automated layout design systems can play important role in not only improving efficiency, effectiveness, and productivity of layout designers but also achieving higher superiority and diversity in layout alternatives. Nevertheless, existing automated layout design systems are generally characterized by drawbacks in such important aspects as flexibility, efficiency, scalability, generalizability, and learnability of the system as well as ability to facilitate the creativity of designers and diversity in solution alternatives. The complexity and subjectivity of layout design necessitate the development of an intelligent system for layout design that deals with such challenging issues and provides efficient means of generating, analyzing and manipulating superior alternative layouts.

This work is primarily motivated by the inadequacy of decision, design, and instructional aids in layout design. To address these needs, we have presented a research framework and a functional prototype for an interactive Intelligent Decision Support System for layout design based on an Expert System paradigm (IDEAL). It consists of an Intelligent Layout Generator that provides a diverse set of superior layout alternatives by employing crisp preferences from a fuzzy Preference Inferencing Agent. The Preference Inferencing Agent, in turn, obtains subjective rules and preferences from various knowledge sources. Furthermore, IDEAL supports interactive, efficient, and knowledge-based production and manipulation of superior layout alternatives. The usual time constraints and frequency of updates required in procurement of a layout justify such a system as an indispensable and high priority tool. Preliminary experiments with our prototype (IDEAL) have provided promising results in terms of both efficiency and quality of the outcome.

Here we summarize the dissertation, results, insights, and its efficiencies and deficiencies. We also provide a summary of significant contributions to the existing research body. Furthermore, we delineate some interesting, challenging, and rewarding future research directions for extending the research in layout optimization, preference modeling, uncertainty management, automated learning, etc. both within the context of the layout design problem and in general.

6.2 Summary of Dissertation

Our research methodology involved surveying available models and methodologies in layout design, identifying their scope and limitations, developing a conceptual framework for alleviating some of their limitations, building a functional prototype for testing and validating the viability of our research framework, testing the prototype using case studies, and reporting conclusions and insights gained in the process.

Layout design is such an intricate problem that it requires an interdisciplinary approach as well as a paradigm shift from the usual hard optimization approach to the one of decision-making and soft computing. As such, our conceptual framework employs a decision-making problem-solving paradigm involving synergistic use of several tools and techniques from Soft Computing and Machine Intelligence. The emphasis is on development of a tool that could supplement the knowledge, experience, and design intuition of layout designers. In addition, it provides a vehicle to further the research and instructional efforts in this important direction.

In this thesis, we have limited ourselves to a two-dimensional oriented bin-packing (2D-BPP) formulation of the layout design problem. It is because a 2D-BPP formulation can easily be adapted for several important layout design applications. Consequently, we deem 2D-BPP as a generalized formulation that could form a basis for developing a generic approach towards solving the layout design problem.

6.3 Interpretations of Results and Insights

The exact approach to such subjective and uncertain problem domains as the layout design is neither efficient nor effective. Our results have demonstrated the efficacy of the proposed knowledge-based layout design approach as well as tools and heuristics employed.

The Genetic Algorithms based approach, in conjunction with efficient and effective placement/decoding heuristics, can provide a diverse set of superior layout alternatives to decision-makers. The fuzzy inferencing is useful in furnishing various parameters and weights for hybrid fitness evaluation function in the intelligent layout generator, resulting in improvements in overall efficiency and efficacy. Furthermore, automated preference discovery is a viable knowledge acquisition option in layout design. Technologies like Artificial Neural Networks and Reinforcement Learning might prove valuable towards this end.

This research framework and the prototype are explicitly designed to aid decision-makers and it is expected to continuously increase productivity, lower costs, reduce waste, improve customer satisfaction, and spur innovation. In addition, it is expected to reduce planning efforts and planning time. The shorter cycle time would in turn translate into reduction in the uncertainty involved.

Time efficiency in generating superior layout outcomes is certainly among the more important measures of the success of an automated layout design system. A near-optimal and superior solution procured within a reasonable time-frame is often adequate for practical applications. IDEAL employs very efficient and effective procedures for layout alternative generation. However, the effectiveness and robustness of the system also determine the success of the system. IDEAL consistently provides superior layout alternatives. Furthermore, it provides interactive means for knowledge-based generation and manipulation of the layout alternatives.

However, the caveat is that IDEAL may result in an *anchoring effect* – the tendency to make decisions based on inadequate adjustment of subsequent estimates from an initial estimate that serves as an anchor. As such, a solution rated as superior by an automated system might engender an inadequate judgment by the user. The *Automation bias* means that humans have a propensity to discount or not search for contradictory information in presence of a computer-generated solution that is deemed as an immaculate outcome (Mosier & Skitka, 1996; Parasuraman & Riley, 1997). The inherent complexity means decision support systems that integrate sophisticated levels of automation may inadvertently permit users to "perceive the computer as a legitimate authority, diminish moral agency, and shift accountability to the computer, thus create a moral buffering effect" (Parasuraman & Riley, 1997).

However, such caveats are part of a typical layout design process, manual or otherwise and overly trusting automation in such complex system operations as layout optimization is a well-recognized decision support problem (Mosier & Skitka, 1996). Furthermore, the potential benefits of systems like IDEAL outweigh these shortcomings. Moreover, such shortcomings can easily be evaded with little prudence and creative thinking by users, coupled with a diverse set of superior alternatives, which are essential ingredients of any decision-making process.

6.4 Feedback from Practitioners and Researchers

During the course of this thesis, some experienced researchers and practitioners have been selected to solicit feedback concerning our approach, in general, and IDEAL, in particular. One of those experts had more than 20 years of experience in teaching, researching, and publishing in the field of layout optimization and facilities design. Another has been researching and publishing in this area for more than 15 years; these communications started in February 2002. In addition, an anonymous technical support specialist from VIP-PlanOpt™ had provided many useful insights to the problem and market demands that helped guide this research; these interactions started in January 2003.

These practitioners demonstrated great interest in the research outcome and research ideas were deemed enablers of both productivity improvement of layout designers and procurement of superior layout designs. They were positive about the simplicity and usefulness of paradigms and heuristics presented in this thesis. The technical support specialist at PlanOpt.com demonstrated interest in incorporating such interactive decision support facilities in their future versions of software and was keen to know the progression and outcome of the research. The other two researchers in the layout optimization area employed earlier versions of IDEAL as teaching aids for undergraduate courses in facilities layout design. They were particularly interested in the interactivity and visibility of the design process where students were able to view and compare outcomes against a variety of fitness measures.

Students had the opportunity to contrast the power, performance, productivity, and practicality of IDEAL with such popular and dedicated facility layout optimization software as SPIRAL, VIP-PlanOpt, FACOPT, etc. We deemed comments from novices in the layout design field important from the ease of use, ease of learning, perceived usefulness, and speed of execution perspectives. Such comments were very useful in improving IDEAL. For instance, the idea of augmenting IDEAL with a *comparison* module in future, facilitating interactive comparisons and manipulations of a pair of superior layout alternatives, has come from one of the students who used IDEAL. Indeed, students only had access to the developers' interface, as the planned user interface has still to be implemented completely. Nevertheless, students appreciated the *visibility* of process dynamics, the *usability* of rule-base creation module, and the ability to *control* the various process parameters.

The following are some of the remarks made by those practitioners:

- IDEAL is suitable at the initial layout planning stage, at which time only imprecise information is available. It can help one arrive at a rational and multi-criteria based decisions quickly and help in some operational planning.
- The repeated evaluation and selection process is suitable for activities that involve relatively permanent outcomes of strategic significance.
- The evaluation metrics for considering a layout alternative can affect the outcome and necessitate that special attention be paid to the process of selecting fitness evaluation metrics. Equipping decision-makers with a variety of fitness metrics would alleviate this concern.
- Linguistic modeling using fuzzy logic seems to provide flexibility and ease in knowledge acquisition and approximate reasoning through fuzzy inferencing seems appropriate in uncertain situations. However, the same might not be true if the problem at hand is structured and parameters can be predicted fairly accurately.

6.5 Comparison of IDEAL with Existing ALD Systems

This research is primarily motivated by lack of elaborate methodologies for tackling the layout design problem and inadequacy of existing ALD systems. IDEAL alleviates various limitations of existing ALD systems. In the following, we provide a meta-level comparison between capabilities of IDEAL and capabilities of most existing ALD systems.

Attribute	Existing ALD Systems	IDEAL
Scalability	Cannot efficiently handle large-scale problems	Scalable to handle large problems consisting of thousands of modules
Productivity and Efficiency	Slow and inefficient	fast and efficient
Fitness Metrics & MCDM Capability	Single (or few) rigid and myopic quantitative fitness metric(s) to guide the optimization search	Several encompassing quantitative and qualitative fitness evaluation metrics are available (and easy to augment further) for a hybrid fitness metric affording MCDM
Quality of Alternatives	Lower aesthetic value with no provision to afford layout aesthetics guiding the layout optimization process	Generates superior alternatives against quantitative fitness metrics (more than 25% improvement for medium-scale problems); provision for incorporating layout aesthetics in the design process (more than two-fold experts' qualitative ratings)

Diversity of Alternatives	Little or no diversity; deterministic and localized search	Generates many diverse and superior alternatives; Non-deterministic and global search
Preference Modeling & Uncertainty Handling	Little or no capability to model and incorporate user preferences in design optimization; Lack robust means of handling uncertainty and subjectivity in preferences	Flexible, linguistic, and natural modeling of user preferences; Fuzzy Logic provides capability to handle uncertainty and subjectivity in preferences
Simplicity, Usability, & Visibility of Interface Controls	Simpler interface; fewer functionalities; lack powerful capabilities & flexibility required in such complex domains; Difficult to learn; limited user control	Complex interface with many important and powerful functionalities and controls; usable and easy to use interface offering users much desired flexibility; easy to learn
Creativity; Interactivity; User Control; ease of manipulation	Difficult to learn; limited user control; little or no provision for manipulating alternatives; require auxiliary CAD software; Disregard human creativity/expertise	Allows for interactive, extensive, visible, informed, and easy controls for generation/manipulation of alternatives and benefiting from human creativity; benefits from human creativity and expertise through extensive user control on layout generation/manipulation
Learnability and Transparency	Unable to learn from experience; little or no explanation capability	Can be augmented to incorporate learning abilities; Affords basic explanation capability
Backtracking & Post-Optimization Processing	Most systems require expensive backtracking and/or post-optimization processing for generating feasible and practical solutions	No need for backtracking; little or no need of post-optimization processing
Gneralizability; Portability; Reusability	Employ relatively simpler assignment models that are not applicable in most scenarios; little or no portability; non-portable data storage/retrieval; difficult to adapt in changed scenarios	Employs a generic 2D-BPP problem formulation; easily adaptable to various 2D layout design applications; portable data and knowledge storage/retrieval system; separation of knowledge-base and inferencing mechanisms that affords reusability; flexible in augmenting or updating knowledge-base

6.6 Summary of Contributions

This thesis contributes to the fields of layout optimization, soft computing, and knowledge-based systems in various ways. Here we list some of these contributions:

1. **Surveyed available models and methodologies in layout planning and design:**

   We provided an integrative and comparative survey of existing models, problem formulations, and solution methodologies in layout design and analysis.

2. **Identified the scope and limitations of available models and methodologies for layout planning and design:**
    a. We identified that most research on automation in layout design is limited to the development and improvement of algorithms, heuristics, and mathematical programs and is quite inadequate. It is primarily because the prevailing solution paradigm in layout design field is that of Optimization instead of a more relevant and effective Decision Making paradigm.

    b. We identified various existing ad hoc, user-controlled, deterministic, probabilistic, and machine learning techniques for modeling the layout design problem as well as user preferences and constraints. We compared these techniques based on several important aspects and discussed their pros and cons with some recommendations.

    c. We identified various ad hoc, probabilistic, and approximate reasoning inferencing mechanisms for utilizing subjective and uncertain user preferences. We compared these techniques based on several important aspects and discussed their pros and cons while providing some recommendations.

    d. We identified some promising tools, techniques, and solution paradigms for efficient and effective decision support through automated layout design/planning.

    e. We provided our recommendations regarding the selection of the preference modeling techniques in layout design. Furthermore, results of a small-scale exploratory study are provided, which involved subjective evaluation of merits and demerits of various popular modeling techniques by some experts in layout design.

    f. A mathematical formulation for a generic two-dimensional oriented bin-packing problem is provided and adapted for the use with genetic algorithms based layout optimization.

3. **Developed a conceptual framework for the layout planning and design process:**
    a. We identified various major phases and related issues in the automated layout design process and the logical inter-relationship/flow among those components.

b. We formulated some effective quantitative and qualitative layout fitness evaluation metrics. Furthermore, we proposed an encompassing fitness evaluation regime affording Multi-Criteria Decision Making (MCDM) in layout design.

c. We formulated a research framework for solving a generic layout design problem. We discussed the philosophy and the synergy of tools and techniques deemed promising for implementing the proposed research framework.

4. **Built a working and scalable research prototype of the proposed paradigm for testing and validating the viability and efficacy of the concept through simulation studies:**

    a. We proposed several efficient, effective, and robust layout optimization heuristics, which provide superior solutions in terms of both quantitative and qualitative fitness. An ability to obtain solutions with high aesthetic contents is an important achievement in such subjective problem domains as layout design. Superiority of proposed heuristics is demonstrated through comparative analyses with various efficient and popular existing heuristics using several benchmark problems.

    b. We formulated a metaheuristics based approach for building a fast, effective, and robust system for automating the generation of diverse layout design alternatives.

    c. We developed a Fuzzy inferencing mechanism for modeling of, and reasoning with, subjective and uncertain design preferences.

    d. We developed a functional, scalable, and interactive prototype of the proposed paradigm for testing and validating the effectiveness of the solution paradigm in supplementing the experience, intuition, and erudition of layout designers.

5. **Tested the research prototype using real world case studies:**

    We tested the efficacy of our system using case studies in bin-packing, besides testing on various benchmark problems. Results and insights of our case studies are reported in Chapter 4.

6. **Reported the conclusions and insights gained during this thesis:**

Conclusions and insights gained during course of this thesis are reported in this dissertation with results and their interpretations in a variety of contexts. These conclusions and insights are summarized in this Chapter.

7. **Identified efficiencies, efficacies, and deficiencies of the research done during the course of this thesis:**

   The system we have implemented employs very efficient and effective procedures for layout alternative generation. Our system consistently provides superior layout alternatives. Furthermore, our system provides interactive means for knowledge-based generation and manipulation of the layout alternatives. However, despite having superiority in terms of efficiency, scalability, and efficacy, this system is still a research tool that is continuously being evolved into an even more powerful and effective decision support system. Consequently, IDEAL has its limitations that are pointed out throughout the thesis and summarized in Section 6.7.

8. **Proposed some future directions for furthering this research:**

   Automated layout design is a prolific research area with every research endeavor opening new vistas. Likewise, besides tackling several important issues, this thesis provides many interesting research directions in various contexts. Such research opportunities are pointed out throughout this thesis and summarized in Section 6.8.

6.7 Limitations of Research

Here we describe limitations in the scope of this thesis:

- The scope of this research is largely limited to the Intelligent Layout Generator and the Preference Inferencing Agent.
- It is assumed that all the necessary data is available and does not require any pre-processing. Furthermore, it is assumed that the modules were categorized based on suitable criteria and assigned a utility value based on economic and/or visual appeal. IDEAL does not have such pre-processing capabilities.

- Modules are assumed oriented in nature. However, some provision for decision maker to choose between oriented modules scenario and situation where module rotations are permitted would add to the scope of IDEAL.
- It is assumed that Module shapes are pre-determined and rigid. However, some capability of handling flexible modules, as well, would broaden the scope of IDEAL.
- It is assumed that modules are rectangular. However, some layout design applications may call for packing of non-rectangular modules such as polygons and circles. In such scenarios, algorithms currently employed in ILG would not work and a new knowledge base of procedures and algorithms would be required.
- This research does not explicitly provide means to handle three dimensional layout designs such as those encountered in facilities and VLSI layout designs. Nevertheless, the problem-solving paradigm still is still applicable and it is possible to extend this research to multi-bin scenarios and adapt it for the three-dimensional layout design problems.
- The explanation facility is quite unsophisticated. Users might want to get more in-depth explanation of the behavior of IDEAL.
- The automated preference discovery concept has been demonstrated as a viable option. However, the PDA has not been directly incorporated in IDEAL.
- The end-user interface needs to be implemented and tested.
- Data import and export formats are limited to text and csv (comma separated values) formats only, thereby limiting the portability of IDEAL.

6.8 Future Work

It is hoped that the exclusive and complementary features of various soft computing technologies will result in a synergistic integration that would provide new insights to practitioners and theoreticians and thus open up new frontiers. Here we list some of those interesting future research directions.

6.8.1 Metaheuristics

Currently, the GA based metaheuristic search approach in IDEAL supports layout design scenarios involving only one bin or packing space. However, the system can be modified to support both multi-bin and undersized bin scenarios. Under such scenarios, some peculiarities may transform the dynamics of the problem and open up some interesting research venues.

In a multi-bin scenario, modules may be placed in a given number of bins, possibly with some effect on the total utility of the layout design. For instance, placement of a particular module on the homepage of an e-Store would have different utility than the case where the same module is placed in one of the subsequent pages.

In an undersized bin scenario, the size of a bin might not be adequate to accommodate all modules. As such, only a subset of modules may be accommodated in a specific layout alternative. In such scenarios, the intrinsic utility of modules as well as inter-module interaction would have more significant role in determining the layout fitness.

6.8.2 Layout Design Heuristics

The need for efficient and effective heuristics in layout design is an ongoing research area where the quest for more useful heuristics would not only facilitate improvements in productivity but also provide more insights to the layout design problem. Heuristics capable of producing solutions with higher aesthetic contents are also important in such subjective problem domains as layout design.

In future, we want to investigate means to facilitate fuzzy placement decisions, such as skipping some less promising placement steps for expediting the design process when the hamming distance between two genes is large. For instance, if the hamming distance between two modules in a chromosome, say *A* and *B*, is large then there is little promise in exploring placement of module *B* at the corners of module *A*, which are more likely to be occupied already.

6.8.3 Uncertainty Management

Uncertainty and subjectivity involved in most layout design work domains mean that developing robust methods for uncertainty management would remain an important issue. Although FL seems to be a logical choice in cost-effective and robust uncertainty management as well as explanation capabilities, it would be useful to compare the well-known modeling approaches by evaluating their relative merits and demerits. Thus, empirical and theoretical investigations regarding the suitability of different techniques under different operating conditions are desired. Generally, studies in various techniques for uncertainty management do not provide such comparative treatments. However, such a comparative treatment would potentially provide valuable insights to the strengths and weaknesses of those techniques. Consequently, it would help in identifying suitable uncertainty management technique(s) for various sources of uncertainty in different work domains. It would enable future researchers in tapping on strengths of many techniques and making more informed decisions. An ability to account for interdependencies and interactions among various preferences in the inferencing mechanism provide very prolific but challenging research streams.

6.8.4 Multi-Criteria Decision Making

A review of existing and promising fitness evaluation metrics for various layout design domains would be a worthwhile effort. It would provide guidelines to layout designers regarding prudent selection of fitness metrics that may form the basis for Multi-Criteria Decision Making. We also aspire to explore ways of leveraging on extensive information theory literature in developing more encompassing and meaningful hybrid fitness layout metrics as well as comparison of those in terms of informational value. For instance, two seemingly different metrics for gauging different aspects of layout utility does not always mean any significant improvement in actual informational value.

6.8.5 Automated Learning

We have demonstrated that automated preference discovery is a pragmatic strategy that offers value in face of difficulty in explicitly articulating preferences by the decision maker. The promise of automated preference discovery provides several potential research streams. For instance, such automatically discovered preferences need to be adjusted or refined based on users' interactions with the preliminary or intermediate alternatives. Explicitly articulating such adjustments in learned preferences by the decision maker might not always be a feasible or an efficient approach. As such, we also need some mechanism to automatically update these preferences. ANN may be used in such an incremental learning mode. However, we believe, few instances of user interactions might not provide sufficient or efficient re-training of the ANN. Consequently, we plan to incorporate a Reinforcement Learning (RL) mechanism for automated updating and refining of preferences and test the viability of automated preference discovery concept under dynamic scenarios.

6.8.6 Graphical User Interface

An effective and interactive end-user interface would have a crucial role in the effectiveness as well as acceptability of any computerized layout design approach. Towards this end, a prototype of an effective user interface has been developed, and tested, using the philosophy of ecological interface design as well as various usability and Human-Computer Interaction guidelines. However, we have not implemented the end-user interface as this work is still in the development phase and present interface reflects complexities of the task as well as functionalities required by experts and developers of the system.

6.8.7 Explanation Facilities

An explanation facility is the ability to thoroughly explore the implications of knowledge models and bases of system's adaptations. Explanation Facilities indicating to users the reasoning behind actions is an important part of the proposed research paradigm. It would provide users a sense of control by making the system 'scrutable'. However, such capabilities need to be both visible and comprehensible to the user. In this direction, a visible method for understanding and controlling the system's adaptations or user profiles might augment the acceptability of the system.

6.8.8 Personalized Decision Support

Our system affords both individual and group decision making scenarios. However, as already mentioned in Section 5.6, various technologies can be employed to create profiles of decision makers. It would be interesting to customize the system based on the user profile for providing personalized decision support. For instance, users may have various cognitive biases, as discussed in Section 2.9. A personalized decision support may be used to compensate for such cognitive biases. Research in adaptive user interfaces would play a prominent role in realizing such endeavors.

6.8.9 Empirical Evaluation

The real-world deployment of such interactive decision support systems with a demonstrable effectiveness is a formidable task. Conceivably, a review of existing automated layout design systems reveals little work in this direction. Nevertheless, some carefully designed and conducted empirical studies of actual users would help reinforce, contradict, and refine designs to better accommodate and satisfy users. Some possible metrics for evaluating such systems may include users' subjective evaluation of interaction quality, user-friendliness, effectiveness, generalizability, scalability, accessibility, acceptability, the degree of task simplification, etc. Despite the difficulty in quantifying these aspects, the subjects' rating could be useful for evaluating IDEAL and its interaction quality. Comprehensive research materials on instruments and techniques for guiding such empirical evaluations are available in the literature (IS World, 2005).

6.9 Concluding Remarks

In this thesis, we developed and implemented a new research paradigm for layout design. This efficient, effective, and intelligent neuro-fuzzy-genetic approach for solving the layout design problems provides interesting research directions as well as a vehicle for furthering this research. The research prototype (IDEAL) can solve large-scale continuous space layout design problems consisting of unequal size modules with relatively little computational efforts. The proposed framework and the research prototype system contribute to the field of decision support in the layout design by enabling explicit representation of experts' knowledge and formal modeling of fuzzy user preferences. It is expected to improve the cognitive, ergonomic and economic efficiency and effectiveness of layout designers.

This thesis should furnish researchers and practitioners in layout design area a better understanding of tools and ideas in tackling the layout design problem. This research framework may evolve in a natural progression towards developing some more powerful and robust systems. We believe that such research would prove a worthwhile effort in providing valuable decision support to the layout designers. Furthermore, it may find useful applications in such seemingly disparage areas as intelligent tutoring systems, dynamic memory allocation, multi-server scheduling, and metacomputing. It is expected that the proposed approach will provide answers to some questions raised in practical applications of layout design and facilitate further research in this direction. In addition, this work is expected to result in identification of other areas craving for such interdisciplinary solution approaches. This research also provides a basis to researchers in knowledge-based systems, as well as subjective decision-making problems, on how to expand on existing toolsets for solving various complex operations management problems.

BIBLIOGRAPHY

1. Abdinnour-Helm, S., and Hadley, S.W., "Tabu search based heuristics for multi-floor facility layout", *Int'l Journal of Production Research*, Vol. 38, No. 2, pp. 365-383, 2000.
2. Adaptive Hypermedia Homepage, http://wwwis.win.tue.nl/ah/.
3. Adya, S. N., Markov, I. L., and Villarrubia, P.G., "Improving Min-cut Placement for VLSI Using Analytical Techniques", *Proc. IBM ACAS Conference*, IBM ARL, February 2003, pp. 55-62.
4. Agah, A. and K. Tanie, "Intelligent Graphical User Interface Design Utilizing Multiple Fuzzy Agents", *Interacting with Computers*, Vol. 12, pp. 529-542, 2000.
5. Ahmad, A.R., "A Research Framework for Intelligent Decision Support in the Web Page Layout Design and Presentation Planning", *M.S. Thesis* (completed but not committed), Management Sciences, Faculty of Engineering, University of Waterloo, April 2002.
6. Ahmad, A.R., Basir, O.A., Hassanein, K., "Fuzzy Inferencing in the Web Page Layout Design", In J. Bézivin, J. Hu, Z. Tari (Eds.), *Proceedings of the 1st Workshop on Web Services: Modeling, Architecture and Infrastructure (WSMAI-2003), in conjunction with the 5th Int'l Conference on Enterprise Information Systems (ICEIS-2003),* ICEIS Press, Angers, France, pp. 33-41, April 2003.
7. Ahmad, A.R., "An Effective User Interface for Knowledge-based Decision Support in Layout Design", *Working Paper*, Systems Design Engineering, University of Waterloo, 2004.
8. Ahmad, A.R., Basir, O.A., Hassanein, K., "Intelligent Decision Support System for Layout Design", *Proc. of the 9th Asia-Pacific Decision Sciences Institute Conf. (APDSI'04)*, Seoul, Korea, July 2004.
9. Ahmad, A.R., Basir, O.A., Hassanein, K., "Decision Preferences, Constraints, and Evaluation Objectives in Layout Design: A Review of Modeling Techniques", *Proc. of the 5th Int'l Conf. on Operations and Quantitative Management (ICOQM-V)*, Oct. 2004, Seoul, Korea, 2004.
10. Ahmad, A.R., Basir, O.A., Hassanein, K., "A Placement Algorithm for Efficient Generation of Superior Decision Alternatives in Layout design", *Proc. of the 5th Int'l Conf. on Operations and Quantitative Management (ICOQM-V)*, Oct. 2004, Seoul, Korea, 2004.
11. Ahmad, A.R., Basir, O.A., Hassanein, K., "Adaptive User Interface in Intelligent Multimedia Educational Systems: Issues and Trends", *Proceedings of the Int'l Conference on Electronic Business* (*ICEB'04*), Dec. 2004, Beijing, China, 2004.
12. Ahmad, A.R., Basir, O.A., Hassanein,K., "Improved Placement Algorithm for Layout Optimization", *Proc. of the 2nd Int'l Industrial Engineering Conf. (IIEC'04)*, Dec. 2004, Riyadh, Saudi Arabia, 2004.
13. Aiello, G., and Enea, M. "Fuzzy approach to the robust facility layout in uncertain production environment", *Int'l Journal of Production Research*, Vol. 39, No. 18, pp.4089-4101, 2001.
14. Akin, O., Dave, B., Pithavadian, S., "Heuristic generation of Layouts (HeGeL): based on a paradigm for problem structuring", *Environmental Planning*, Vol. 19, No. 2, pp. 225-245. 1997.
15. Akoumianakis, D., and Stephenidis, C., "Knowledge-Based Support for User Adapted Interaction Design", *Expert Systems with Applications*, Vol. 12, pp. 33-59, 1992.
16. Akoumianakis, D., Savidis, A., Stephanidis, C., "Encapsulating intelligent interactive behaviour in unified user interface artifacts", *Interacting with Computers,* Vol. 12, pp. 383–408, 2000.
17. Albrecht, D., Zukerman, I., Nicholson, A., "Bayesian Models for Keyhole Plan Recognition in an Adventure Game", *User Modeling and User-Adapted Interaction*, Vol. 8, pp. 5-47, 1999.

18. Al-Sultan, K. S., Hussain, M. F., and Nizami, J. S., "A genetic algorithm for the set-covering problem *Journal of the Operations Research Societ*y, Vol. 47, pp. 702-709, 1996.

19. Aouni, B., F. Abdelaziz, J-M. Martel, "Decision-Maker's Preferences Modeling in the Stochastic Goal Programming", *European Journal of Operational Research*, Vol. 162, pp. 610-618, 2005.

20. Armour, G. C., and E. S. Buffa, "A heuristic algorithm and simulation approach to relative allocation of facilities", *Management Science*, Vol. 9, pp. 294-309, 1963.

21. Ayyub, B.M., "*Elicitation of Expert Opinions for Uncertainty and Risks"*, CRC Press, NY, 2001.

22. Azadivar, F., and G. Tompkins, "Simulation Optimization with Quantitative Variables and Structural Model Changes: A Genetic Algorithm Approach", *Euro. J. of Op. Res.*, Vol. 113, pp. 169-182, 1999.

23. Azadivar, F., and J.Wang, "Facility Layout Optimization using Simulation and Genetic Algorithms", *International Journal of Production Research*, Vol. 38, No. 17, pp. 4369-4383, 2000.

24. Badiru, A., and Arif, A., "FLEXPERT: Facility Layout Expert System using Fuzzy Linguistic Relationship Codes", *IIE Transactions*, Vol. 28, No. 4, pp. 295-309, 1996.

25. Baker, B.S., Coffman, E.G., and Rivest, R.L., "Orthogonal Packing in Two Dimensions", *SIAM Journal on Computing*, Vol. 9, pp. 846-855, 1980.

26. Balakrishnan, J., Cheng, C.H., Wong, K.F., "FACOPT: a user-friendly FACility layout OPTimization system", *Computers and Operations Research*, vol. 30, pages 1625-1641, 2003.

27. Baron, L., Achiche, S., and Balazinski, M., "Fuzzy decision support system knowledge base generation using genetic algorithm ", *Int'l J. of Approximate Reasoning*, Vol. 28, pp. 125-148, 2001.

28. Bauer, M., Gmytrasiewicz, P., Pohl, W., "Machine learning for user modeling", *Seventh International Conference on User Modeling*, Banff, Canada, 1999.

29. Baykasoglu, A., and N.N.Z. Gindy, "A Simulated Annealing Algorithm for Dynamic Layout Problem", *Computers & Operations Research*, Vol. 28, pp. 1403-1426, 2001.

30. Bazaraa, M. S., "Computerized Layout Design: A Branch and Bound Approach", *AIIE Transactions*, Vol. 7, No. 4, pp. 432-438, 1975.

31. Benjaafar, S., "Flexible Factory Layouts", in Graves, R., et al., *Progress in Materials Handling Research*, Materials handling Institute, Charlotte, NC, 2000.

32. Berkey, J. O., and P. Y. Wang, "Two-dimensional finite bin-packing algorithms", *Journal of Operational Research Society*, Vol. 38, No. 5, pp. 423-429, 1987.

33. Berkun, S., Design and Usability Training Manager, Microsoft Corporation, *Personal Comm.*, 2002.

34. Berkun, S., "*Usability in Web*", available on: *http://www.uiweb.com/issues/*, accessed on Jan. 2002.

35. Berthold, M.R., and Huber, K.P., "Missing values and learning of fuzzy rules", *Intl Journal of Uncetainty, Fuzziness, and Knowledge-Based Systems*, Vol. 6, No. 2, pp.171-178, 1998.

36. Bertsekas,D.and Tsitsiklis,J, "*Neuro-Dynamic Programming",* Athena Scientific,Belmont,MA, 1996

37. Bianchi-Berthouze, N., L. Berthouze, and T. Kato, "Understanding Subjectivity: An Interactionist View", In Kay, J. (ed.), *User Modeling: Proceedings of the Seventh International Conference, UM99,* Springer Wien, New York, pp. 3-12, 1999.

38. Billsus, D., Brunk, C.A., Evans, C., Gladish, B., Pazzani, M. "Adaptive interfaces for ubiquitous Web access", In P. Brusilovsky and M. T. Maybury (eds.), *Communications of the ACM (*Special Issue on the Adaptive Web), Vol. 45, No. 5, pp. 34-38, 2002.

39. Blair, E. L., and S. Miller, "An interactive approach to facilities design using microcomputers", *Journal of Computers in Industrial Engineering,* Vol. 9, pp. 91-102, 1985.

40. Biron, A., "Artificial Neural Networks", *http://www.ifh.de/~biron/diplom/node29.html,* 1999.

41. Bookbinder, J.H., and J.K. Higginson, "Customer service vs trim waste in corrugated box manufacture", *Journal of Operational Research Society,* Vol. 37, No. 11, pp. 1061-1071, 1986.

42. Borning, A., Lin, R.K.H., and K. Marriott, "Constraint-based document layout for the Web", *Multimedia Systems,* Vol. 8, pp. 177-189, 2000.

43. Bouyssou, D, and Vincke, P. (eds.), *Preference Modeling – Annals of Operations Research*, Vol. 80, pp. 147-180, 1998.

44. Bower, J., and Q. Zi-Lei, "Satisficing when buying information", *Organizational Behavior and Human Decision Processes*, Vol. 51, pp. 471-481, 1992.

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

46. Bozer, Y. A., and R. D. Meller, "A reexamination of the distance-based facility layout problem", *IEEE Transc.*, Vol. 29, pp. 549-560, 1997.

47. Brinck, T., D. Gergle *et al.*, "*Usability for the Web: Designing Web Sites that Work*", San Francisco, CA, Morgan Kaufmann, 2000.

48. Brusilovsky, P., "Methods and Techniques of Adaptive Hypermedia", *User Modeling and User-Adapted Interaction*, Vol. 6, No. 2/3, pp. 87-129, 1996.

49. Brusilovsky, P., Eklund, J., "A Study of User Model Based Link Annotation in Educational Hypermedia", *J. of Universal Computer Science*, Vol. 4, No. 4, pp. 429-448, 1998a.

50. Brusilovsky, P., Pesin, L. "Adaptive navigational support in educational hypermedia: An evaluation of the ISIS-Tutor", *J. of Computing and Information Technology*, Vol. 6, No. 1, pp. 27-38, 1998b

51. Brusilovsky, P., Eklund, J., Schwarz, E., "Web-based education for all: A tool for developing adaptive courseware", *Proc. of 7th Int'l World Wide Web Conference*, April 14- 18, 1998, pp. 291-300, 1998c.

52. Brusilovsky, P., "Adaptive hypermedia", *User Modeling and User Adapted Interaction*, Ten Year Anniversary Issue (Kobsa, A., ed.), Vol. 11, No. 1/2, pp. 87-110, 2001.

53. Brusilovsky,P., Maybury,M.T., "From adaptive hypermedia to adaptive Web", In P.Brusilovsky, M. T.Maybury (eds.), *Comm. of ACM (*Sp. Issue on the Adaptive Web), Vol.45, No.5, pp.31-33, 2002.

54. Bunt, A., Conati, C., "Probabilistic Student Modeling to Improve Exploratory Behaviour", *User Modeling and User-Adapted Interaction*, Vol. 13, pp. 269-309, 2003.

55. Buffa, E.S., Armour, G.C., and Vollman, T.E., "Allocating facilities using CRAFT", *Harvard Business Review,* Vol. 42, 1964.

56. Burke, E.K., Kendall, G., Whitwell, G., "A New Placement Heuristic for the Orthogonal Stock-Cutting Problem", *Operations Research*, Vol. 52, No. 4, pp. 665-671, 2004.

57. Burns, C. M., and Hajdukiewicz, J.R., "*Ecological Interface Design*", CRC Press, 2004.

58. Calistri-Yeh, R. J., "Utilizing User Models to Handle Ambiguity and Misconceptions in Robust Plan Recognition", *User Modeling and User-Adapted Interaction,* Vol. 2, pp. 289-322, 1991.

59. CDS, Cadence Design Systems, Inc., San Jose, CA, USA, *http://www.cadence.com/*, 2004.

60. Chan, F. T. S., Jayaprakash, B., Tang, N. K. H. "Design of automated cellular manufacturing systems with simulation modelling: a case study", *J. of Computer Appl. in Tech.,* Vol. 8*,* No. 1/2, pp. 1-11, 1995.

61. Chazelle, B., “The Bottom-Left Packing Heuristic: An Efficient Implementation”, *IEEE Transactions on Computers,* Vol. 32, pp. 697-707, 1983.

62. Chen, Q., and Norcio, A.F., “Modeling users with neural Architecture”, *Proceedings of 1992 International Joint Conference on Neural Networks,* pp. I547-I552, 1992.

63. Chen, T.Y., and C.C. Shieh, “Fuzzy Multiobjective Topology Optimization”, *Computers and Structures,* Vol. 78*,* pp. 459-466, 2000.

64. Cheng, R., M. Gen, Tozawa, T., “Genetic Search for Facility Layout Design Under Interflows Uncertainty”, *Japanese Journal of Fuzzy Theory and Systems,* Vol. 8*,* pp. 335-346, 1996.

65. Chung, F.K.R., M.R. Garey and D.S. Johnson, “On packing two-dimensional bins”, *SIAM Journal on Algebraic and Discrete Methods*, Vol. 3, No. 1, pp. 66-76, 1982.

66. Chung, Y-K., “A neuro-based expert system for facility layout construction”, *Journal of Intelligent Manufacturing*, Vol. 10, No. 5, pp. 359-3856, 1999.

67. Cintula, P.,Navara,M., “Compactness of Fuzzy Logics”, *Fuzzy Sets & Systems*, Vol. 143, pp. 59-73, 2004.

68. Cohoon, J.P., S.U. Hegde, W.N. Martin, and D.S. Richards, “Distributed genetic algorithms for floor-plan design problem”, *IEEE Transactions On CAD*, Vol. 10, No. 4, pp. 483-492, 1991.

69. Conati, C., Gertner, A., VanLehn, K., “Using Bayesian Networks to Manage Uncertainty in Student Modeling”, *User Modeling and User-Adapted Interaction*, Vol. 12, pp. 371-417, 2002.

70. Conati C. and Zhao X., “Building and Evaluating an Intelligent Pedagogical Agent to Improve the Effectiveness of an Educational Game”, *Proceedings of International Conference on Intelligent User Interfaces (IUI’04)*, Jan. 13-16, Madeira, Portugal, 2004.

71. Corbett, A., McLaughlin, M., Scarpinatto, K.C., “Modeling Student Knowledge: Cognitive Tutors in High School and College”, *User Modeling and User-Adapted Interaction*, Vol. 10, pp. 81-108, 2000.

72. Cordon, O., Gomide, F., Herrera, F., Hoffmann, F., Magdalena, Ten years of genetic fuzzy systems: current framework and new trends”, *Fuzzy Sets and Systems*, Vol. 141, pp. 5-31, 2004.

73. Cox, E., “*The Fuzzy Systems Handbook*”, 2nd ed., Academic Press, Chestnut Hill, MA, 1999.

74. De Bo, J., and Vervenne, L., “Benefits of Explicit Profiling in the IMAGEN Project”, *Technical Report – STAR-2003-3*, Systems Technology and Applications Research (STAR) Lab, Vrije Universitiet, Brussel, Belgium, http://www.imagenlab.org.

75. Deb, S.K., and Bhattacharyya, B., “Fuzzy decision support system for manufacturing facilities planning”, *Decision Support Systems*, to appear, 2004.

76. K. Doll, F.M. Johannes, and G. Sigl, “Domino: deterministic placement improvement with hill-climbing capabilities”, *Proc. of the IFIP International Conference on VLSI*, pp. 91–100, Aug. 1991.

77. Donaghey, C. E., and Pire, V. F., “Solving the facility layout problem with BLOCPLAN”, *Industrial Engineering Department*, University of Houston, Houston, 1990.

78. DOT – the US Department of Transportation, http://www.dot.gov/PerfPlan2004/mobility.html, accessed on November 22, 2004.

79. Dowsland, K.A., S. Vaid, and W.B. Dowsland, “An algorithm for polygon placement using a bottom-left strategy”, *European J. of Operational Res.*, Vol. 141 (Sp. issue on cutting & packing), pp. 371-381, 2002.

80. Dvorak, A., and Novak, V., “Formal Theories and Linguistic Descriptions”, *Fuzzy Sets and Systems*, Vol. 143, pp. 169-188, 2004.

81. Dweiri, F., and F.A. Meier, “Application of fuzzy decision-making in facilities layout planning”, *International Journal of Production Research*, Vol. 34, No. 11, pp. 3207-3225, 1995.

82. Dweiri, F., "Fuzzy development of crisp activity relationship charts for facilities layout", *Computer and Industrial Engineering*, Vol. 36, pp. 1-16, 1999.

83. Dyckhoff, H., "A topology of cutting and packing problems", *European Journal of Operational Research*, Vol. 44, No. 2, pp. 145-159, 1990.

84. Eastman, C., "Automated space planning", *Artificial Intelligence*, Vol. 4, pp. 41-64, 1973.

85. Eguchi, K., Yamashiro, O., Kawamoto, H., Tsuji, N., Yamane, S., and Oshima, K., "Application of fuzzy inference and genetic algorithms to VLSI floorplan design", Industrial Electronics Society, 2000 (IECON 2000), *26th Annual Conference of the IEEE*, Vol. 1, pp. 184-188, 2000.

86. Eisenmann, H, Johannes,F.M, "Generic Global Placement and Floorplanning", *DAC 199*8, pp. 269, 1998.

87. El-Bouri, A., N. Popplewell, S. Balakrishnan, and A. Alfa, "A Search Based Heuristic for Two Dimensional Bin-Packing Problem", *INFOR*, Vol. 32, No. 4, pp. 265-274, 1994.

88. El-Said, M.G., G. Fischer, S.A. Gamael-Din, and M. Zaki, "ADDI: A tool for automating the design of visual interfaces", *Computers and Graphics*, Vol 21, No. 1, pp. 79-87, 1997.

89. Engelbrecht, A.P., "*Computational Intelligence: An Introduction"*, Wiley, 2002.

90. Engineering Optimization Software Inc., "VIP-PLANOPT: A Visual Tool for Automating the Design of Facilities Layout", 1386 Pritchett Industrial Drive, Austell, GA 30168, *www.PlanOpt.com*, 2005.

91. Eom, H.B., and Lee, S.M., "DSS Applications Development Research: Leading Institutions and Most Frequent Contributors (1971 - April 1988)", *Decision Support Systems*, Vol. 6, pp. 269-275, 1990.

92. Epstein, S. L., B. Moulin, W. Chaker, J. I. Glasgow, J. Gancet, "Pragmatism and Spatial Layout Design", *COSIT 2001*, pp.189-205, 2001.

93. Esbensen,H., and P.Mazumdar, "SAGA: a unification of the genetic algorithm with simulated annealing & its application to macrocell placement", *Proc. of IEEE Int'l Conf. on VLSI Design*, 1994.

94. Evans, G.W., M.R. Wilhelm, and W. Karwoski, "A layout design heuristic employing the theory of fuzzy sets", *International Journal of Production Research*, Vol. 25, No. 10, pp. 1431-1450, 1987.

95. Fisher, A.M., and S.Y. Nof, "FADES: Knowledge-based facility design", *Proceedings of the International Industrial Engineering Conference*, Chicago, May 6-10, pp. 74-82, 1984.

96. Flemming, U., Baykan, C., Coyne, R., "Hierarchical generate-and-test versus constraint-directed search", in Gero, J., *Proc. of AI in Design conference (1992)*, pp. 917-838, Kluwer, 1992.

97. Flemming, U., Coyne, R., Fenves, S., Garrett, J., and Woodbury, R., "SEED – software environment to support early phases in building design", *IKM'94*, pp. 5-10, Weimer, Germany, 1994.

98. Fodor, J.L.,M.Roubens, "*Fuzzy Preference Modeling and Multicriteria Decision Support*", Kluwer, 1995.

99. Foley, J., W. Kim, S. Kovacevic, and K. Murray, "UIDE-An intelligent user interface design environment", In J W Sullivan and S.W. Taylor (Eds.), *Intelligent User Interface*, ACM Press, 1991.

100. Foulds, L.R., "*Graph Theory Applications*", Springer Verlag, NY, 1991.

101. Foulds, L.R., "LayoutManager: A Decision Support System for Facilities Planning", *Knowledge Based Hybrid Systems*, Vol. B-11, pp. 293-299, 1993a.

102. Foulds, L.R., "Graph Theoretic-Based Decision Support for Facilities Layout", *Knowledge Based Hybrid Systems*, Vol. B-11, pp. 145-157, 1993b.

103. Foulds, L.R., "*Graph Theory Applications*", Springer Verlag, NY, 1995.

104. Foulds, L.R., "LayoutManager: A Micro-computer based Decision Support System for Facilities Planning", *Decision Support Systems*, Vol. 20, pp. 199-213, 1997.

105. Foulds, L.R., “*Graph Theory Applications*”, Springer Verlag, NY, 1998.

106. Francis, R.L., and J. L. White, “*Facility Layout and Location*”, Prentice Hall, NJ, 1974.

107. Francis,R.L., and J.L.White, “*Facility Layout and Location: An Analytical Approach*”, Prentice,NJ, 1992.

108. Garey, M.R., and D.S. Johnson, “*Computers and Intractability*”, W.H. Freeman Press, NY, 1979.

109. Geigel, J., and A. Loui, “Automatic page layout using genetic algorithms for electronic albuming”, *Working Paper, Research and Development, Eastman Kodak Company*, Rochester, NY, 2001.

110. George, G.M., and G.R. Jones, “*Understanding and Managing the Organizational Behavior*”, 1st ed., MA: Addison-Wesley Inc., 1996.

111. Gloria, A. D., P. Faraboschi, and M. Olovieri, “Block placement with a Boltzman machine”, *IEEE Transc. CAD*, vol. 13, No. 6, pp.694-701, 1994.

112. Goldberg, D.E., “*Genetic Algorithms in Search, Optimization, & Machine Learning*”, Wesley, 1989.

113. Gomez, A., Fernandez, Q.I., Garcia, D.F., Garcia, P.J., “Using genetic algorithms to resolve layout problems in facilities where there are aisles”, *Int'l J. of Prod. Economics*, Vol. 84, pp. 271-282, 2003.

114. Gosavi, A., Bandla, N., Das, T.K., “A reinforcement learning approach to a single leg airline revenue management problem with multiple fare classes and overbooking”, IIE Trans, Vol. 34, pp. 729-742, 2002

115. Graf, W., “The Constraint-Based Layout Framework LayLab and Its Applications”, In: I. F. Cruz, J. Marks, K. Wittenburg, Eds., *Proc. of ACM Workshop on Effective Abstractions in Multimedia: Layout, Presentation, and Interaction*, San Francisco, CA, 1996.

116. Graf, W., “Intelligent multimedia layout: a reference architecture for the constraint-based spatial layout of multimedia presentations”, *Computer Standards and Interfaces*, Vol. 18, pp. 515-524, 1997.

117. Graf, W. H., A. Kroener, S. Neurohr, R. Goebel, “Experience in Integrating AI and Constraint Programming Methods for Automated Yellow Pages Layout”, *Künstliche Intelligenz*, Vol. 12, No. 2, pp. 79-85, 1998.

118. Greenberg, J., R.A. Baron, C.A. Sales and F.A. Owen, “*Behavior in Organizations*”, 2nd Canadian ed., Pearson Education Canada Inc., 2000.

119. Grobelny, J., “The fuzzy approach to facilities layout problems”, *Fuzzy Sets Systems*, vol. 23, pp. 175-190, 1987a.

120. Grobelny, J., “One possible fuzzy approach to facility layout problems”, *Int'l Journal of Production Research*, vol. 25, pp. 1123-1141, 1987b.

121. Grobelny, J., “Flexibility in Layouts: A Simulation Approach”, *Material Flow*, vol. 3, pp. 243-250, 1986.

122. Guiffrida, A. L., and R. Nagi, “Fuzzy Set Theory Applications in Production Management Research: A Literature Survey”, *Journal of Intelligent Manufacturing*, vol. 9, No. 2, pp. 39-56, 1998.

123. Gupta, R.M., “On one ‘possibly’ fuzzy approach to facility layout problems”, *International Journal of Production Research*, vol. 25, pp. 1123-1141, 1987b.

124. Hall, L.O., and A. Kandel, “The Evolution of Expert Systems to Fuzzy Expert Systems”, In A. Kandel (Eds.), *Fuzzy Expert Systems*, pp. 3-21, 1992.

125. Hamachi,G., Mayo,R., Ousterhout,J., Scott,W., Taylor,G. “Magic: an interactive editor for VLSI layouts”, *http://bwrc.eecs.berkeley.edu/Classes/IcBook/magic/index.html*, Uni. of Berkley, 1996.

126. Hassan, M.M.D, “Layout Design in Graph Technology Manufacturing”, *International Journal of Production Research*, vol. 38, pp. 173-188, 1995.

127. Hassan, M.M.D., and G.L. Hogg, "On constructing the block layout by graph theory", *International Journal of Production Research*, vol. 32, No. 11, pp. 2559-2584, 1994.

128. Hassan, M.M.D., "Machine layout problem in modern manufacturing facilities", *International Journal of Production Research*, vol. 29, No. 6, pp. 1263-1278, 1991.

129. Hassan, M.M.D., and G.L. Hogg, "A Review of Graph Theory Applications to the Facilities Layout Problem", *OMEGA*, vol. 15, No. 4, pp. 291-300, 1987.

130. Head, M., and K. Hassanein, "Web Site Usability for eBusiness Success: Dimensions, Guidelines and Evaluation Methods", *Proceedings of the Hawaii International Conference on Business*, 2002.

131. Healy, P., Creavin, M., and Kuusik, A. "An optimal algorithm for rectangle placement", *Operations Research Letters*, Vol. 24, pp. 73-80, 1999.

132. Henze, N., Nejdl, W., "Adaptivity in the KBS Hyperbook System", in Brusilovsky,P., Bra,P.D. (eds.), *The 2nd Workshop on Adaptive Systems and User Modeling on the World Wide Web*, in conj. with *the 8th Int'l World Wide Web Conference*, Toronto, Canada, pp.67-74, May 11-14, 1999.

133. Heragu, S.S., *Facilities Design*, PWS Publishing, Boston, MA, 1997.

134. Heragu, S.S., and A. Kusiak, "Machine layout: an optimization and knowledge-based approach", *International Journal of Production Research*, Vol. 28, No. 4, pp. 615-635, 1990.

135. Heragu, S.S., and A. Kusiak, "Efficient models for the facility layout problem", *European Journal of Operational Research*, Vol. 53, pp. 1-13, 1991.

136. Hirji, K.K., "Exploring Data Mining Implementation", *Comm. of ACM*, vol. 44, No. 7, pp. 87-93, 2001.

137. Holland, J. H., "*Adaptation in natural and artificial systems*", The University of Michigan Press, 1975.

138. Holland, J.H., "*Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence*", MIT Press, 1992.

139. Hopper, E., and B.C.H. Turton, "An Empirical Investigation of meta-heuristic and heuristic algorithms for a 2D packing problem", *Euro. J. of Operational Research*, Vol. 128, pp. 34-57, 2001.

140. Hower, W., W. H. Graf, "A Bibliographical Survey of Constraint-based Approaches to CAD, Graphics, Layout, Visualization, and Related Topics", *Knowledge-Based Systems*, Vol. 9, No. 7, pp. 449-464, 1996.

141. Hower, W., "Placing Computations by Adaptive Procedures", *Artificial Intelligence in Engineering*, Vol. 11, pp. 307-317, 1997.

142. Hurst, "Dynamic Approximation of Complex Graphical Constraints by Linear Constraints", *UIST-2002*.

143. Hurst, "CobWeb: A Constraint based Web Browser", *ACSC-2003*, 2003.

144. Ignizio, J.P., "*Introduction to Expert Systems*", McGraw-Hill Inc., 1991.

145. Ignizio, J.P., T.M. Cavalier "*Linear Programming*", Prentice Hall Inc., 1994.

146. Ilumoka,A.A., "Modular artificial neural network models for simulation and optimization of VLSI circuits", *30th Annual Simulation Symposium (SS'97)*, April 07-09, Atlanta, USA, pp. 190, 1997.

147. Interior Motives, http://www.interiormotivesawards.com/, 2004.

148. Irani, S., Cohen, P.H., and Cavalier, T.M., "Design of Cellular Manufacturing Systems", *Transactions of ASME*, Vol. 114, pp. 352-361,1992.

149. Irani, S., and Huang, H., "Custom Design of Facility Layouts for Multi-Product Facilities using Layout Modules", *IEEE Transactions on Robotics and Automation*, Vol. 16, pp. 259-267, 2000.

150. Irani, S., "Personal Recommendations to Layout Designers", *Yahoo Groups*, 2003.

151. Islier, A. A., "A genetic algorithm approach for multiple criteria facility layout design", *Int'l Journal of Production Research*, Vol. 36, No. 6, pp. 1549-69, 1998.

152. Jackson, P., "*Introduction to Expert Systems*" , Addison Wesley, England, 1999.

153. Jain, V., and R. Krishnapuram, "Applications of Fuzzy Sets in Personalization for E-Commerce" *Proceeding of 20th Intl Conference of North America Fuzzy Information Processing Society (NAFIS)*, pp. 263-268, 2001.

154. Jakobs, S., "On genetic algorithms for packing of polygons", *European Journal of Operational Research*, Vol. 88, pp. 165-181, 1996.

155. Jameson, A., "Numerical uncertainty management in user and student modeling: An overview of systems and issues", *User Modeling and User-Adapted Interaction*, Vol. 5, pp. 193-251, 1996.

156. Jojodia, S., Minis, I., Harhalakis, G., and Proth, J., "CLASSS: Computerized Layout Solutions using Simulated Annealing", *int'l Journal of Production* Research, Vol. 30, No.1, pp. 95-108, 1992.

157. Johns, G., "*Organizational Behavior: Understanding and Managing Life at Work*", 4th ed., HarperCollins Inc., 1996.

158. Johnson, D. S., M. R. Garey e F. R. K. Chung, "On packing two-dimensional bins", *SIAM J. Alg. Disc. Meth*. Vol. 3, pp. 66-76, 1982.

159. Kado, K., "An investigation of genetic algorithms for facility layout problems", *M.S. Thesis,* University of Edinburg, UK, 1995.

160. Kado, K., P. Ross, and D. Corne, "A study of genetic algorithm hybrids for facility layout problems", in *Proc. of the 6th Int. Conf. on Genetic Algorithms (ICGA*), L. J. Eshelman, Ed., San Mateo, CA, 1995, pp. 498–505, Morgan Kaufmann, 1995.

161. Kaelbling, L.P., Littman, M.L., "Reinforcement Learning: A Survey", *Journal of Artificial Intelligence Research*, Vol. 4, pp. 237-285, 1996.

162. Kaku, B.K., Thompson, G.L., and Morton, T.E., "A Hybrid Heuristic for the Facilities Layout Problem", *Computer and Operations Research*, Vol. 18, pp. 241-253, 1991.

163. Kameas, A. and Pintelas, P., "Functional Architecture and Interaction Model of a GENerator of Intelligent Tutoring Applications", *Journal of Systems Software*, Vol. 36, pp. 233-245, 1997.

164. Kang, E.Q., Lin, R.B., and Shragowitz, E., "Fuzzy Logic Approach to VLSI Placement", *IEEE Transactions on Very Large Scale Integration Systems*, Vol. 2, No. 4, 1994.

165. Karray, F., De Silva, C., "*Soft Computing and Intelligent Systems Design: Theory, Tools and Applications*", Readings: Addison-Wesley, 2004.

166. Karray, F., E. Zaneldin, T. Hegazy, A.H.M. Shabeeb, and E. Elbelgati, "Computational Intelligence Tools for Solving Facilities Layout Planning Problem", *Proceedings of the American Control Conference, Chicago, Illinois*, Vol. 8, No. 4, pp. 367-379, June 2000a.

167. Karray, F., E. Zaneldin, T. Hegazy, A.H.M. Shabeeb, and E. Elbelgati, "Tools of Soft Computing as Applied to the Problem of Facilities Layout Planning", *IEEE Transaction on Fuzzy Systems*, Vol. 8, No. 4, pp. 367-379, August 2000b.

168. Kautz, H.A., 'A formal theory of plan recognition', *PhD thesis*, University of Rochester, N.Y., 1987.

169. Kavcic, A., "Dealing with Uncertainty of User Knowledge: Fuzzy User Modelling in Educational Hypermedia", *Advances in Multimedia Communications, Information Processing and Education*, Invited Lecture, Leganés, Spain, October 2002.

170. Keeble, R.J., and R.D. Macredie, "Assistant agents for the world wide web intelligent interface design challenges", *Interacting with Computers*, Vol. 12, pp. 357-381, 2000.

171. Keen, P.G.W., and M.S. Scott-Morton, "*Decision Support Systems: An Organizational Perspective*", Readings: Addison-Wesley Inc., 1978.

172. Kelly,W.E., "Dimensionality in Fuzzy Systems", *Ph.D.Thesis*, Elect. Engg., Texas A&M, Aug. 1997.

173. Khan, J.A., and S.M. Sait, "Fuzzy Aggregating Function for Multiobjective VLSI Placement", *Proc. of the 2002 IEEE Int'l Conference on Fuzzy Systems (FUZZ-IEEE'02)*, May 12-17 2002, pp. 831-836, 2002.

174. Khan, J.A., Sait, S.M., Minhas, M.R., "Fuzzy biasless Simulated Evolution for Multiobjective VLSI Placement", *IEEE ???*, 2002.

175. Kim, J.K., H.L.Kwang, and S.W. Yoo, "Fuzzy bin packing problem", *Fuzzy Sets and Systems*, Vol. 120, pp. 429-434, 2001.

176. Kintsch, W., (1998) *Comprehension: A Paradigm for Cognition*, Cambridge University Press, MA.

177. Kleinhans,J.M., G.Sigl, F.M.Johannes, and K.J.Antreich, "GORDIAN: VLSI placement by quadratic programming and slicing optimization", *IEEE Trans on CAD*, Vol. 10, No. 3, pp. 356–365, 1991.

178. Konar, A., "*Artificial Intelligence and Soft Computing: Behavioral and Cognitive Modeling of the Human Brain*", Readings: CRC Press, Fl., 2000.

179. Kouvelis, P., Kurawarwala A.A., and Gutierrz, G.J., "Algorithms for single and multiple period layout planning for manufacturing systems", *Euro. J. of Op. Res.*, Vol. 63, pp. 287-303, 1992.

180. Kumara, T.S.R., Joshi, T., Kashyap, R.L., Moddie, C.L., and Chang, T.C., "Expert Systems in industrial engineering", *International Journal of Production Research*, 1985.

181. Kumara, T.S.R., Kashyap, R.L., and Moddie, C.L., "Artificial Intelligence techniques in facilities layout planning: the development of an expert system", *TR-ERC 86-1*, Vol. 24, pp. 1107-1125, 1986.

182. Kumara, S.R.T., Kashyap, R.L., and Moodie, C.L., "Application of expert systems and pattern recognition methodologies to facilities layout planning", *International Journal of Production Research,* Vol. 26, No. 5, pp. 905-930, 1988.

183. Kurosu, M., K. Kashimura, "Apparent Usability vs. Inherent Usability", *CHI '95 Proceeding*, Association for Computing Machinery, New York, 1995.

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

185. Lai, K.K., and Chen, W.M., "An evolutionary algorithm for rectangular cutting stock problem", *International Journal of Industrial Engineering,* Vol. 4, pp. 130-139, 1997.

186. Lee, R., J-M.Moore, "CORELAP-computerized relationship layout planning", *Journal of Industrial Engineering,* Vol. 18, No. 3, pp.195-200, 1967.

187. Lee, Y.H., and Lee, M. H., "A shape-based block layout approach to facility layout problem using hybrid genetic algorithm", *Computers and Industrial Engineering*, Vol. 42, pp. 237-248, 2002.

188. Leung, T.W., C.K. Chan, M.D. Troutt, "Application of a mixed simulated annealing-genetic algorithm heuristic for the two-dimensional orthogonal packing problem", *European Journal of Operational Research*, Vol. 145, No. 3, pp. 530-542,2003.

189. Levary, R.R., and S. Kalchik, "Facilities Layout – A survey of solution procedures", *Computers & Industrial Engineering*, Vol. 9, pp. 141-148, 1985.

190. Liao, T.W., "Design of Line typr Cellular Manufacturing Systems for Minimum Operating and Total Material Handling Costs", *Int'l Journal of Production Research*, Vol. 32, No. 2, pp. 387-397, 1993.

191. Ligget, R.S., “A Designer-Automated Algorithm Partnership, An Interactive Graphic Approach to Facility Layout”, In Yehuda E. Kalay (ed.), *Evaluating and Predicting Design Performance*, John Wiley & Sons, 101-124, 1992.

192. Ligget, R.S., “Automated facilities layout: past present and future”, *Automation in Construction*, Vol. 9, 197-215, 2000.

193. Liu, W.Z., White, A.P., Thompson, S.G., and Bramer, M.A., “Techniques for Dealing with Missing Values in Classification”, In Liu, X., Cohen, P., Berthold, M.R. (eds.) *Advances in Intelligent Data Analysis*, LNCS, Vol. 1280, pp. 527-536, 1997.

194. Liu, D., and H. Teng, “An Improved BL-Algorithm for Genetic Algorithm of the Orthogonal Packing of Rectangles”, *European Journal of Operational Research*, Vol. 112, pp. 413-420, 1999.

195. Lodi, A., S. Martello, and D. Vigo, “ Approximation algorithms for the oriented two-dimensional bin packing problem”, *European Journal of Operational Research*, Vol. 112, pp. 158-166, 1999.

196. Lodi, A., S. Martello, and M. Monaci, “Two-dimensional packing problems: A survey”, *European Journal of Operational Research*, Vol. 141, pp. 241-252, 2002.

197. Lok, S. and Feiner, S., “A Survey of Automated Layout Techniques for Information Presentation”, *Communications of ACM,* Vol. 41, No. 7, pp. 81-87, 1998.

198. Love, R.F., and J.Y.Wong, “Solving quadratic assignment problems with rectilinear distance and interger programming”, *Naval Research Log. Quarterly*, Vol. 23, No. 4, pp. 623-627, 1976.

199. Mak, K.L., Y.S. Wong and F.T.S. Chan, “A genetic algorithm for facility layout problems”, *Computer Integrated Manufacturing Systems*, Vol. 11, No. 1-2, pp. 113-127, 1998.

200. Malakooti, B. and Tsurushima, A., “An expert system using priorities for solving multiple-criteria facility layout problems”, *Int’l Journal of Production Research*, Vol. 27, No. 5, pp. 793-808, 1989.

201. Maleki-Dizaji, S., Othman, Z. A., Nyongesa, H.O., Siddiqi, J.I.A., “Evolutionary Reinforcement of User Models in an Adaptive Search Engine”, *Proceedings of IEEE/WIC Int. Conf. On Web Intelligence (WI’03),* pp. 706-709, 2003.

202. Marakas, G.M., “*Decision Support Systems*”, Prentice Hall Inc., 2002.

203. March, J.G., and H.A. Simon, “*Organizations*”, New York: Wiley, 1958.

204. Martello, S. and Vigo, D., “Exact Solution of the Two-Dimensional Finite Bin Packing Problem”, *Management Science*, Vol. 44, No. 3, pp. 388-399, 1998.

205. Martens, J., “Two genetic algorithms to solve a layout problem in fashion industry”, *European Journal of Operational Research*, Vol. 154, pp. 304-322, 2004.

206. Mayo, M., Mitrovic, A., “Optimising ITS Behaviour with Bayesian Networks and Decision Theory”, *Int’l Journal of Artificial Intelligence in Education*, Vol.12, pp. 124-153, 2001.

207. Mazumder, P., and E.M. Rudnick, “*Genetic Algorithms for VLSI Design, Layout and Test Automation*”, Prentice-Hall, NY, 1999.

208. McTear, M.F., ‘Intelligent Interface Technology: From Theory to Reality’, *Interacting with Computers*, Vol.12, pp. 323-336, 2000.

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

210. Mir, M., and M.H. Imam, “Topology optimization of arbitrary-size blocks using bivariate formulation”, *Journal of Computer Aided Design*, Vol. 24, No. 10, pp. 556-564, 1992.

211. Mir, M., and M.H. Imam, "Analytic annealing for macrocell placement problem", *International Journal of Computer and Electrical Engineering*, Vol. 22, No. 2, 1996.

212. Mir, M., and M.H. Imam, "A hybrid optimization approach for layout design of unequal-area facilities", *Computers and Industrial Engineering*, Vol. 39, 2001.

213. Moon, B.R., and Kim, C.K., "Dynamic embedding for genetic VLSI circuit partitioning", *Engineering Applications of Artificial Intelligence*, Vol. 11, pp. 67-76, 1998.

214. Montreuil, B., "A Modeling framework for Integrating Layout Design and Flow Network Design", *Proceedings of the Material Handling Research Colloquium*, Kentucky, pp. 43-58, 1991.

215. Mosier, K. L., and Skitka, L. J. "Human Decision Makers and Automated Decision Aids: Made for Each Other?", In R. Parasuraman and M. Mouloua (Eds.), *Automation and Human Performance: Theory and Applications*, Lawrence Erlbaum Associates, Inc., pp. 201-220, 1996.

216. Murata, H., K. Fujiyoshi, S. Nakatake, and Y. Kajitani, "VLSI Module Placement Based on Rectangle-Packing by the Sequence-Pair", *IEEE Transactions on Computer-Aided Design Of Integrated Circuits And Systems*, Vol. 15, No. 12, December 1996.

217. Muther, R., "*Systematic Plant Layout*", Readings: Ind. Education Institue, Boston, Mas., 1961.

218. Nauck, D., F. Klawonn, and R. Kruse, "*Foundations of Neuro-Fuzzy Systems*", Wiley, 1997.

219. Negnevitsky, M., "*Artificial Intelligence: A Guide to Intelligent Systems*", Pearson, Sydney, 2002.

220. Ngo,D.C.L., "Measuring the aesthetic elements of screen designs" *Displays,* Vol.22, pp.73-78, 2001.

221. Ngo, D.C.L., and B.L. Law, "An expert screen design and evaluation assistant that uses knowledge-based backtracking" *Information and Software Technology,* Vol. 43, 293-304, 2003.

222. Norman, D. A, "*Thing That Make Us Smart: Defending Human Attributes in the Age of Machines*", Persius Books Group, 1993.

223. Norman, B.A., and A.E. Smith, "Considering Uncertainty in Unequal Area Block Layout Design", citeseer.ist.psu.edu/271501.html, 2002.

224. Nyongesa,H.O., Shicheng,T., Maleki-Dizaji,S., Huang,S.-T., Siddiqi,J. "Adaptive web interface design using fuzzy logic", *IEEE/WIC Int. Conf. on Web Intelligence (WI'03),* pp. 671-674, 2003.

225. Osman H.M., A., Georgy, M.E., Ibrahim, M.E., "A hybrid CAD-based construction site layout planning system using genetic algorithms", *Automation in Construction*, Vol. 12, pp. 749-764, 2003.

226. Parasuraman, R., and Riley, V., "Humans and Automation: Use, Misuse, Disuse, Abuse", *Human Factors*, Vol. 39, No. 2, pp. 230-253, 1997.

227. Pfefferkorn,C., "The design problem solver: a system for designing equipment and furniture layouts", In Eastman,C.(ed.), *Spatial Synthesis in Computer-Aided Building Design*, Wiley, pp. 98-146, 1975.

228. Pierce, J.F., and W.B. Crowston, "Tree search algorithms in quadratic assignment problems", *Naval Research Logistics Quarterly*, Vol. 18, pp. 1-36, 1971.

229. Power, D. J., *Supporting Business Decision Making.* DSS Resources, World Wide Web, http://DSSResources.com/, (accessed on Jan. 13, 2002), 2001.

230. Preece, J., Y. Rogers, H. Sharp, D. Benyon, S. Holland, and T. Carey, "*Human-Computer Interaction*", Reading: Addison Wesley Inc., 1996.

231. Raoot, A.D., and A. Rakshit, "A 'fuzzy' approach to facilities layout planning", *International Journal of Production Research*, Vol. 29, pp. 835-857, 1991.

232. Raoot, A.D., and A. Rakshit, “A ‘linguistic pattern’ approach for multiple criteria facilities layout problems”, *International Journal of Production Research*, Vol. 31, pp. 203-222, 1993.

233. Rhodes, J.S., “Usability can save you money”, *http://www.webword.com/moving/savecompany.html*, accessed on Nov. 02 2004, December 2000.

234. Rommelfagner, H, and Slowinski, R., “Fuzzy Linear Programming with Single or Multiple Objective Functions”, In Slowinski, R., *Fuzzy Sets in Decision Analysis, Operations Research and Statistics*, Kluwer, pp. 179-213, 1991.

235. Rosenblatt, M.J. and Kropp, D.H., “Single Period Stochastic Plant Layout Problem”, *AIEE Transactions*, Vol. 24, pp. 169-176, 1992.

236. Ruan, D., “*Intelligent Hybrid Systems: Fuzzy Logic, Neural Networks, and Genetic Algorithms*”, Reading: Kluwer Academic Publishers, 1997.

237. Saaty, R.W., “The Analytic Hierarchy Process-What it is and how it is used”, *Mathematical Modeling*, Vol. 9, No. 3-5, pp. 161-176, 1987.

238. Sainfort, F.C., D.H. Gustafson, K. Bosworth, and R.P. Hawkins, “Decision support systems effectiveness: Conceptual framework and empirical evaluation”, *Organizational Behavior and Human Decision Processes*, Vol. 45, pp. 232-252, 1990.

239. Saletic, D.Z., Valasevic, D.M., Mastorakis, V.M., “The Influence of Shapes of Fuzzy Sets Membership Functions on Fuzzy System Performances”, *Int’l J. of Comp. Res.*, Vol. 11, No. 2, 2002.

240. Sait, M., El-Maleh, A., and Al-Abaji, R., “General Iterative Heuristic for VLSI Multiobjective Partitioning”, *Proc. of the IEEE Int’l Symposium on Circuits & Systems (ISCAS)*, V497-V500, 2003.

241. Schnecke, V., and O. Vonberger, “Genetic Design of VLSI Layouts”, *Proc. of the 1st IEE/IEEE Conference on GAs in Engineering Systems: Innovations and Applications (GALESIA’95), Sept. 1995, IEEE Conference Publication No. 414,* pp. 430-435, 1995.

242. Schnecke, V., and O. Vonberger, “Hybrid Genetic Algorithms for Constrained Placement Problems”, *IEEE Transactions on Evolutionary Computation,* Vol. 1, no. 4, pp. 266-277, 1997.

243. Scott-Morton, M.S., “*Management Decision Systems: Computer based Support for Decision Making*”, Division of Research, Harvard University, Cambridge, Massachusetts, 1971.

244. Seehof, J.M., and Evans, W.O., “Automated layout design programme”, *Journal of Industrial Eng.,* Vol. 18, No. 12, pp. 690-695, 1967.

245. Shafer, G., (1976) *A Mathematical Theory of Evidenc*e. Princeton University Press, Princeton, NJ.

246. Shneiderman, B., “Designing the User Interface: Strategies for Effective Human-Computer Interaction”, *International Journal of Human-Computer Studies*, Vol. 47, pp. 5-29, 1997a.

247. Shneiderman, B. (1997) ‘Direct manipulation for comprehensible, predictable and controllable user Interfaces’, in *Proc. of 1997 Intl. Conf. on Intelligent User Interface*s, ACM, Orlando, FL, USA, January 1997b.

248. Shore, R.H., Tompkins, J.A., “Flexible Facilities Design”, *AIIE Transactions*, Vol. 12, pp. 200-25, 1990.

249. Siarry, P., L. Bergonzi, and D. Dreyfus, “Thermodynamic optimization of block placement”, *IEEE Transaction on CAD*, Vol. 6, No. 2, pp. 211-221, 1987.

250. Simon, H.A., “A Behavioral Model of Rational Choice”, *Quarterly Journal of Economics*, Vol. 69, pp. 99-118, 1955.

251. Simon, H. A., *“Administrative Behavior”*, 2nd ed., Reading: Free Press, NY, 1957a.

252. Simon, H. A., *“Models of Man”*, Reading: John Wiley Inc., NY, 1957b.

253. Singh, N. and M.H. Wang, “Concurrent engineering in a high variety printing environment”, *Int’l Journal of Production Research*, Vol. 32, No. 7, pp. 1675-1691, 1994.

254. Sly, D., “Computerized Facilities Design and Management”, *IIE Solutions*, pp. 43-51, 1995.

255. Soltani, A.R., and Fernando, T.., “A fuzzy based multi-objective path planning of construction sites”, *Automation in Construction,* to appear, 2004.

256. Souliah, A., “Simulated Annealing for manufacturing systems layout design”, *European Journal of Operational Research*, Vol. 82, No. 3, pp. 592-614, 1995.

257. Stam, A., and A.P.D. Silva, “A hierarchical AHP-DEA methodology for the facilities layout design problem”, *European Journal of Operational Research,* Vol. 145, pp. 92-108, 2003.

258. Sutton, R.S. and Barto, A.G., “*Reinforcement Learning: An Introduction”*, MIT Press, 1998.

259. Svestka, J.A., “MOCRAFT: a professional quality microcomputer implementation of craft with multiple objectives”, *Computers & Industrial Engineering*, Vol. 18, No. 1, p. 13-22, 1990.

260. Taguchi,G., Elsayed,E.A., Hsiang, T., “*Quality Engineering in Production Systems*”, McGrawHill, 1989.

261. Tam, K.Y., and S.G. Li, “A hierarchical approach to facility layout problem”, *International Journal of Production Research*, Vol. 29, No. 1, pp. 165-184, 1991.

262. Tam, K.Y., “A simulated annealing algorithm for allocating space to manufacturing cells”, *International Journal of Production Research*, Vol. 30, No. 1, pp. 63-87, 1992.

263. Tam, C.M., Tong, T. K.L., Leung, A.W.T., Chiu, G.W.C., “Site Layout Planning using Nonstructural Fuzzy Decision”, *J. of Construction Engineering and Management*, May/June 2002, pp. 220-231, 2002.

264. Tate, D. M., and A. E. Smith, “Genetic Algorithm Optimization Applied to Variations of the Unequal Area Facilities Layout Problem”, *Proc. of 2nd Industrial Engng Research Conference*, pp. 335-339, 1993.

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

266. Tommelein, I.D., “SightPlan—An Expert System that Models and Augments Human Decision-Making for Designing Construction Site Layouts”, *Ph.D. Dissertation*, Department of Civil Engineering, Stanford University, Stanford, CA, August 1989.

267. Tommelein, I.D., “SightPlan: a Case Study of BB1”, Section 5.5.1 in Sriram, R.D (ed.). *Intelligent Systems for Engineering: A Knowledge-Based Approach*. Springer Verlag, London, U.K, 1997.

268. Tompkins, J.A., J.A. White, Y.A. Bozer, and J.M.A. Tanchoco, “*Facilities Planning*”, 3rd Ed., Reading: John Wiley Inc., NY, 2002.

269. Tresp, V., Ahmad, S., and Neuneier, R., “Training Neural Networks with Deficient Data”, In Cowan, J.D., Tesauro, G., Alspector, J. (eds.), *Advances in Neural Information Processing Systems,* Morgan Kaufman, Vol. 6, pp. 128-135, 1994.

270. Triantaphyllou, E., and C. Lin, "Development and Evaluation of Five Fuzzy Multi-Attribute Decision-Making Methods," *Approximate Reasoning*, Vol. 14, No. 4, pp. 281-310, 1996.

271. Triantaphyllou, E., Shu, B., Nieto Sanchez, S., and Ray, T., “Multi-Criteria Decision Making: An Operations Research Approach”, in J. D. Webster (Ed.) *Encyclopedia of Electrical and Electronics Engineering*, John Wiley, NY, CA, 1998.

272. Tsay, R.S., Koh, E., and Hsu, C.P., ““PROUD: A Sea-Of-Gate Placement Algorithm”, *IEEE Design and Test of Computer*s,, Vol. 5, No. 6, pp. 45-56, 1988.

273. Tsuchiya, K., S. Bharikar, and Y. Takefuji, “A neural network approach to facility layout problems”, *European Journal of Operational Research*, Vol. 89, pp. 556-563, 1996.

274. Turban, E., "*Decision Support and Expert Systems: Management Support Systems*", 4th ed., Reading: Prentice Hall Inc., Englewood Cliffs, NJ, 1995.

275. Turban, E., and J.E. Aronson, "*Decision Support Systems and Intelligent Systems*", 6th ed., Reading: Prentice Hall Inc., Upper Saddle River, NJ, 2001.

276. Tversky, A. and D. Kahneman, "Availability: A heuristic for judging frequency and probability", *Cognitive Psychology*, Vol. 5, pp. 207-232, 1973.

277. Tversky, A., and D. Kahneman, "Judgment under Uncertainty: Heuristics and Biases", *Science*, Vol. 185, pp. 1124-1131, 1974.

278. Unigraphics Solutions Inc., "VisFactory and FactoryOpt", www.ugs.com, 2005.

279. Walenstein, A., *"Cognitive Support in Software Engineering Tools: A Distributed Cognition Framework"*, Ph.D. Thesis, Computing Science, Simon Fraser University, 2002. http://www.cacs.louisiana.edu/~walenste/pubs/2002may-u-sfu-scs-walenstein-phd.pdf

280. Wang, H., Huang, W., Zhang, Q., Xu, D., "An improved algorithm for the packing of unequal circles within a larger containing circle", *European J. of Operational Research,* Vol. 141, pp. 440-453, 2002.

281. Webb, G. I., Pazzani, M. J. and Billsus D., "Machine Learning for User Modeling", *User Modeling and User Adapted Interaction*, Vol. 11, No. 1-2, pp.19-29, 2001.

282. Welgama, P.S., and P.R.Gibson, "A construction algorithm for a machine layout problem with fixed pick-up and drop-off points", *Int'l Journal of Production Research*, Vol. 31, No. 11, pp. 2575-2590, 1993.

283. Welgama, P.S., S. Palitha, and P.R.Gibson, "Computer-Aided Facility Layout – A Status Report", *International Journal of Advanced Manufacturing Technology*, Vol. 10, No. 1, pp. 66-77, 1995.

284. White, L., and A. Taket, "The Death of the Expert", *Journal of the Operational Research Society*, Vol. 45, pp. 733-748, 1994.

285. Whyte, T.C., and M.R. Wilhelm, "The Use of Fuzzy Methodologies And Space-Filling Curves In Facility Layout Design", *Proceedings of IIE Research Conference*, Phoenix, Arizona, May 1999a.

286. Whyte, T.C., "Facility layout design using fuzzy linguistic variables and fractals", *Proceedings of IIE Research Conference*, Phoenix, Arizona, May 1999b.

287. Wong, B.K., and J.A.Monaco, "A Bibliography of Expert System Applications for Business (1984-192)", *European Journal of Operational Research,* Vol. 85, pp. 416-432, 1995.

288. Wu, Y.L., Huang, W., Lau, S.C., Wong, C.K., Young, G.H., "An effective quasi-human based heuristic for solving the rectangle packing problem", *European J. of Operational Res.,* Vol. 141, pp.341-358, 2002.

289. Yang, S., Choi, B.K., and Sarrafzadeh, M., "Timing-Driven Placement using Design Hierarchy Guided Constraint Generation", *ICCAD 2002,* pp. 42, 2002.

290. Yang, T., and C. Kuo, "A Hierarchical AHP/DEA Methodology For The Facilities Layout Design Problem", *European Journal of Operational Research,* Vol. 147, pp. 128-136, 2003.

291. Yasdi, R., "A Literature Survey on Applications of Neural Networks for Human-Computer Interaction", *Neural Computing and Applications*, Vol. 9, pp. 245-258, 2000.

292. Youssef, H., S.M. Sait, and H. Ali, "Fuzzy Simulated Evolution Algorithm for VLSI Placement", *International Journal on Applied Intelligence, Special issue on Applied Metaheuristics*, 2003a.

293. Youssef, H., S.M. Sait, and Adichi, H., "Evolutionary algorithms, Simulated Annealing and Tabu Search: A Comparative Study", *Int'l Journal on Applied Intelligence, Sp. issue on Applied Metaheuristics*, 2003b.

294. Zadeh, L.A., "Fuzzy Sets", *Inform. Contr.*, Vol. 8, pp. 338-353, 1965.

295. Zadeh, L.A., “Fuzzy Sets as a Basis for a Theory of Possibility”, *Fuzzy Sets and Systems*, Vol. 100 (Supplement), pp. 9-34, 1999.

296. Zha, X.F., “Soft Computing Framework For Intelligent Human–Machine System Design, Simulation And Optimization”, *Soft Computing*, Vol. 7, pp. 184-198, 2003.

297. Zha, X.F., and Lim, S.Y.E., “Intelligent design and planning of manual assembly workstations: A neuro-fuzzy approach”, *Computers & Industrial Engineering*, Vol. 44, pp. 611-632, 2003.

298. Zhan, M., and Graf, W., “Backtracking along with constraint processing and their time complexities”, *journal of Experimental and Theoretical Artificial Intelligence*, Vol. 8, No. 1, pp. 63-74, 1996.

299. Zhang, J.P., Liu, L.H., Coble, R.J., “Backtracking along with constraint processing and their time complexities”, *J. of Experimental &Theoretical Artificial Intelligence*, Vol. 8, No. 1, pp. 63-74, 1996.

300. Zukerman, I., Albrecht, D., “Predictive Statistical Models for User Modeling”, *User Modeling and User Adapted Interaction*. Vol. 11, No. 1-2, pp. 5-18, 2001.

Appendix A – Glossary of Terms

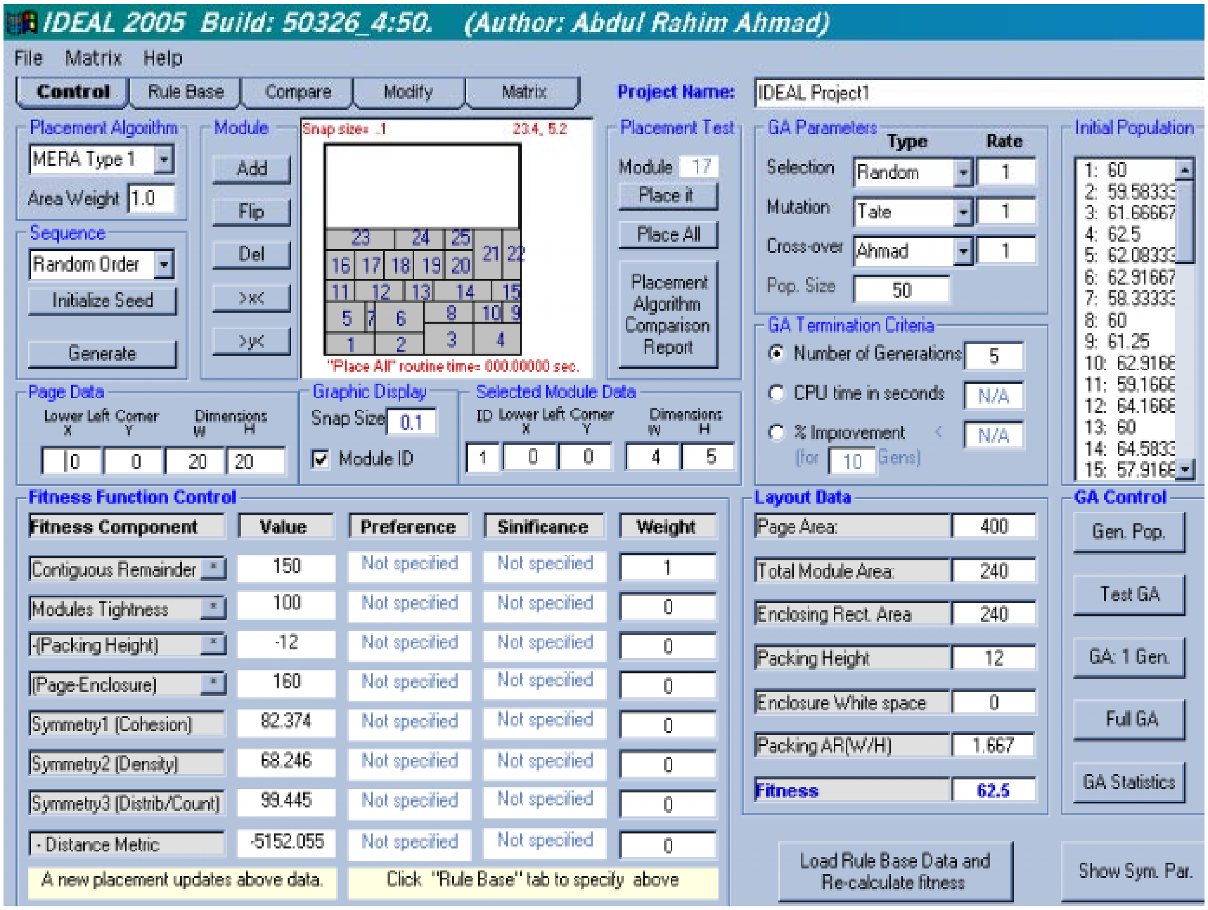
Term	Definition
Automation Bias	Human propensity to discount or not search for contradictory information in presence of a computer-generated solution that is deemed as an immaculate outcome
Availability Heuristic	The human tendency to rely on recent events and information or whatever information readily available to decision makers from their memory. It is a cognitive bias that cannot easily be established and rectified.
Bounded Rationality	Decision makers often resort to *bounded-rationality* reflecting on inadequacy of tangible and intangible resources (Greenberg *et al.*, 2000; Simon, 1957*a*). Framing and cognitive biases demonstrate the operation of bounded rationality.
Cognitive Overhead	"The additional effort and concentration necessary to maintain several tracks or trails at one time" (Conklin, 1987, *pp. 40*). The term *Cognitive Overload* refers to a psychological phenomenon characterized by an overload of information for a decision maker (i.e. the magnitude of information surpasses the person's cognitive capability).
Confirmation Bias	*Confirmation bias* refers to the tendency to dig out only the information that conforms one's own view of the situation. It is a cognitive bias that inhibits people from acquiring additional relevant information
Cognition	The mental process or faculty of knowing, including aspects such as awareness, perception, reasoning, and judgment.
Decision	A decision is a plan of action that is ready for implementation and Decision-Making is the process of developing commitment to some specific course of action (George, 1996). Whereas, a *problem* exists if a gap is perceived between some existing state and the desired state (Johns, 1996).

Term	Definition
Decision Support Systems	Power (1999) defines DSS as "interactive, computer-based tools intended to help decision makers use data, documents, knowledge, and models to identify and solve unstructured problems and make decisions". Keen and Scott-Morton (1978) argue that the idea of decision support evolved from "the theoretical studies of organizational decision-making done at Carnegie Institute of Technology during the late 50's and early 60's and the technical work on interactive computer systems mainly carried out at MIT in 60's". The concepts involved in Decision Support Systems (DSS) were articulated as early as in 1971 by Scott-Morton using the term 'Management Decision Systems' (Scott-Morton, 1971).
Disorientation	*Disorientation* is defined as "the tendency to lose one's sense of location and direction in a nonlinear document" (Conklin, 1987, p. 40).
Dynamic Rationality	Changes in preferences resulting from decision-makers' interaction with existing or intermediate solutions.
Expert Systems	An ES is a computer program capable of performing at the level of a human expert in a narrow domain (Negnevitsky, 2002). ES operate at the decision-gate conducting complex search and interpretive procedures to produce partial to complete solutions that appear as advice, recommendations, or even decisions.
Framing	Decision-makers often resort to making presumptions regarding some aspects of the available information, an affinity referred to as *framing*. It refers to the aspects of presentation of information about a problem that are presumed by decision makers. The way problems and decision alternatives are framed could have a powerful impact on resulting decisions.
Ill-Structured Problem	An *ill-structured* problem is one that tends to be complex, relatively novel, subjective, and uncertain. Such undertakings require high degree of creativity and expertise. Under most favorable conditions, a problem is *well structured* when existing and desired states are clear and the process involved in achieving the desired state is obvious. Such problems are simple, recurring, and familiar. Since decision-making is time consuming and error-prone, a 'program' or standardized procedure for solving well-structured problems is preferred. Ill-structured problems cannot be programmed and decision makers should opt to non-programmed decision-making. These would involve creativity and substantial efforts in collecting more information and be self-consciously extra analytical in approach bringing as much structure to the unstructured problem as possible (George and Jones, 1996).

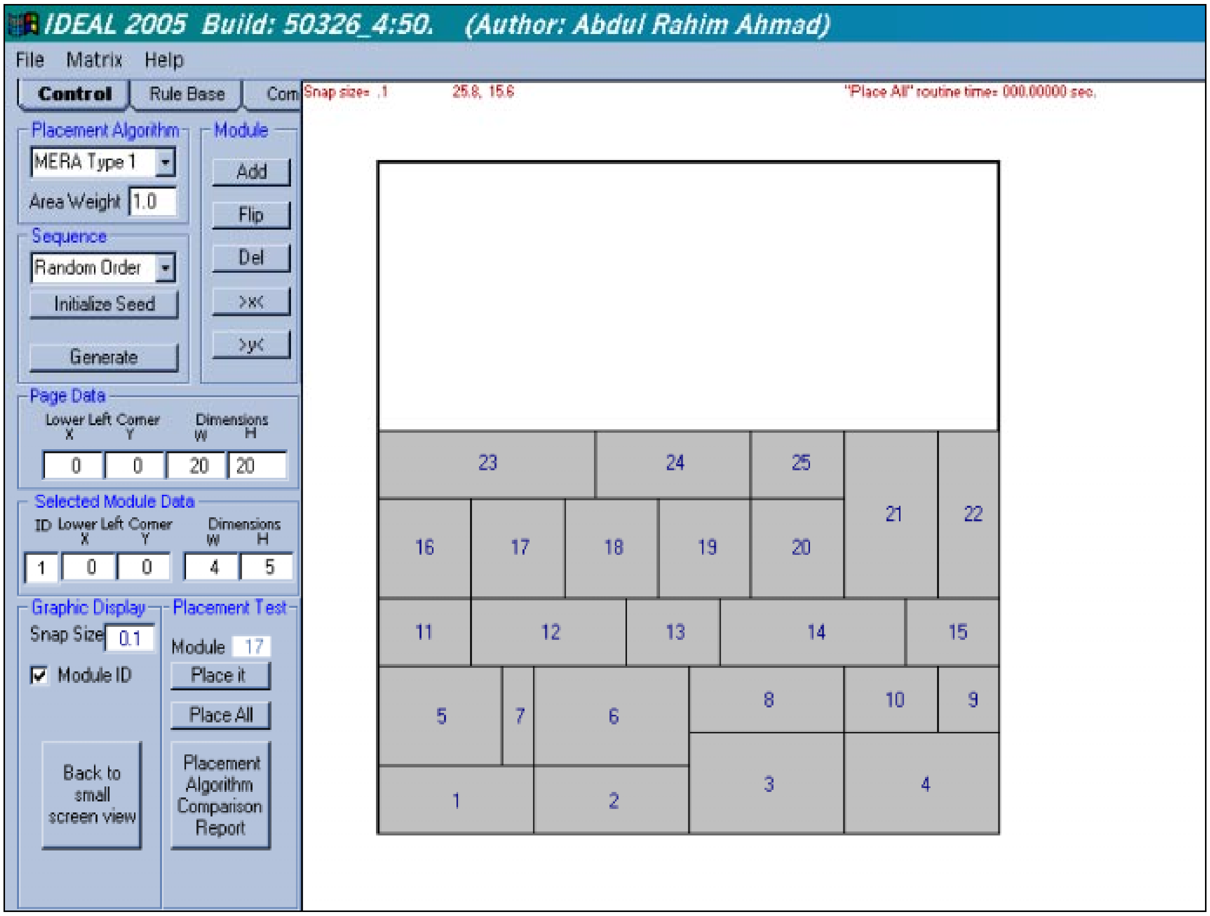
Term	Definition
Information Overload	It refers to a situation where more information is acquired or available than is necessary to make effective decisions. The information overload interferes with performance. Problem is the inability to find key information, to separate the relevant information from noise, and to read all the relevant information – often impossibility. It is obvious that providing information to web designers automatically does not improve decision making and/or efficiency when information flow is so vast, chaotic, and corrupted. Decision makers facing information overload often attempt to use all the information at hand, and then get confused, and permit low-value or irrelevant information to affect their judgments.
Metacomputing	A computing paradigm based on a set of machines networked together for increased computational performance.
Neuron	Neurons are the nerve cells that make up the central nervous system. They consist of a nucleus, a single axon that conveys electrical signals to other neurons and a host of dendrites, which deliver incoming signals.
Perfect Rationality	A decision strategy that is completely informed, perfectly logical, and oriented towards economic reward is referred to as *perfect rationality* (Simon, 1955). Nevertheless, the notion of rational decision-making is unrealistic as the assumption that decision makers have all the relevant information to make an optimal decision bears little resemblance with the real world (Simon, 1955). Another problem with decision-making relates to the evaluation of alternatives. In case of perfect rationality, the evaluation of alternatives is objective. In complex unstructured problems, such as Web page layout design, seeking an objective measure for evaluation and comparison of alternatives might not be an easy task.
Production Paradox	People are not always eager to learn new things, but they want to get their work done, i.e., getting the job done is the primary focus. The *production paradox* attempts to explain why increases in computational support do not necessarily result in increased productivity. One would only ever *want* to learn to use a new tool if one wanted first to get something done. Nevertheless, wanting to get something done can also be a problem, if one lacks the prerequisites: you have to learn to do in order to do. Merely wanting to use a new tool may be necessary but it is not sufficient.

Term	Definition
Satisficing	Decision makers operate under bounded rationality and satisfice rather than optimize. *Satisficing* refers to 'establishing an adequate level of acceptability' for a solution to a problem and screening solutions until one that exceeds this level is found (Simon, 1957*b*; Bower and Zi-Lei, 1992).
Usability	Accommodating users with different skills, knowledge, age, gender, handicaps, literacy, culture, income etc.
User Interface	The aspect of a computer or program that is visible to the user, giving and accepting information from him or her.

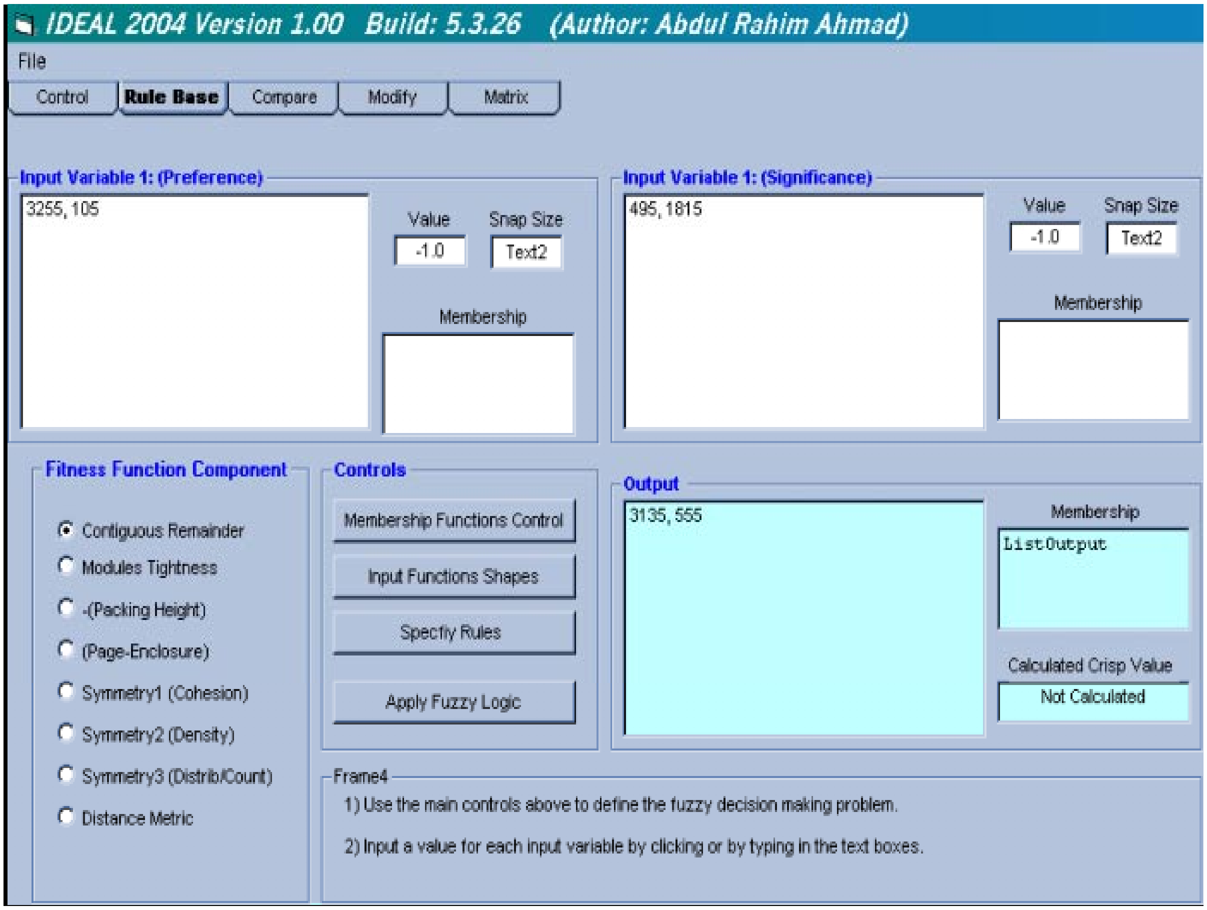
Appendix B – Graphical Interfaces in IDEAL - Screenshots



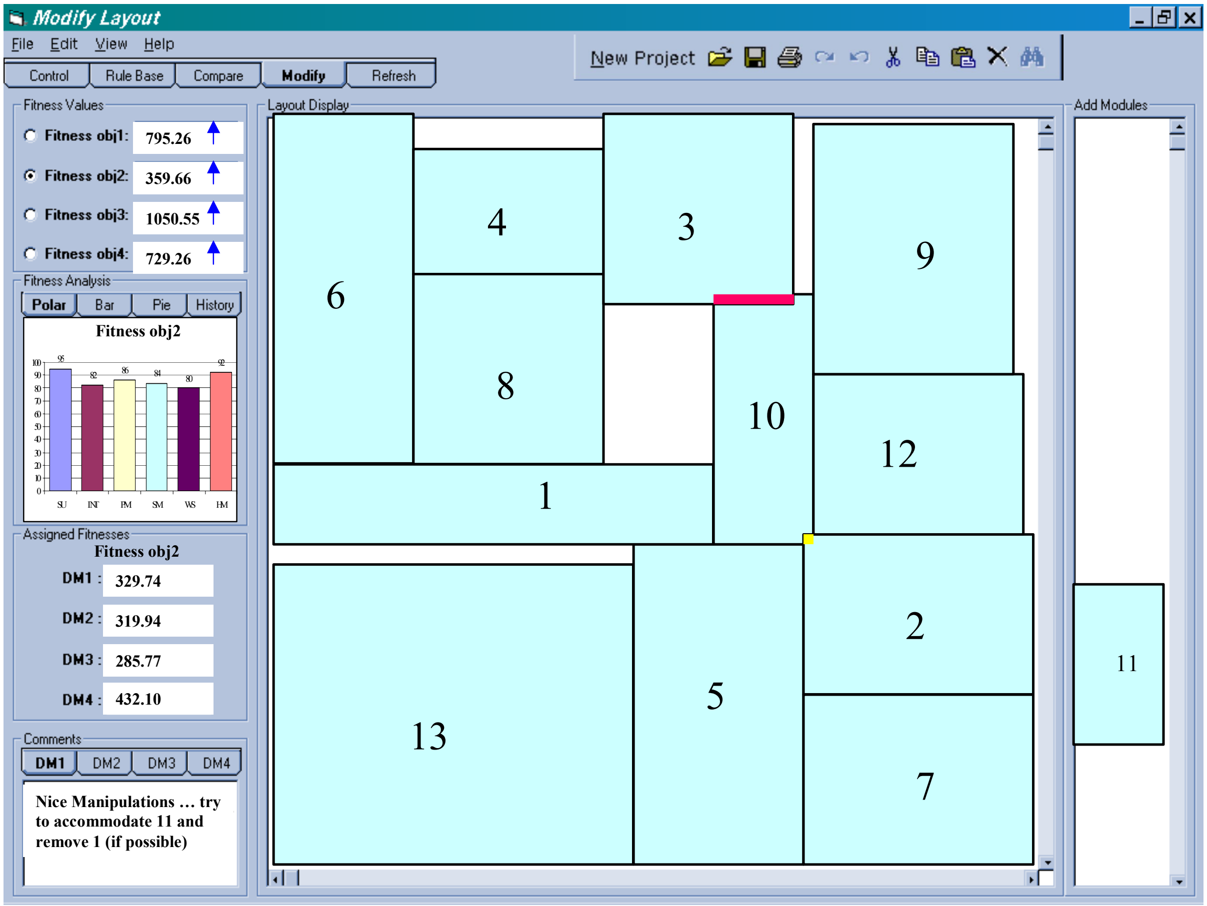
B-I: Graphical Interface for Developers (Expert Controls) – Normal View



B-II: Graphical Interface for Developers (Expert Controls) – Zoomed View



B-III: Graphical Interface for Knowledge Engineers (Knowledge Controls)

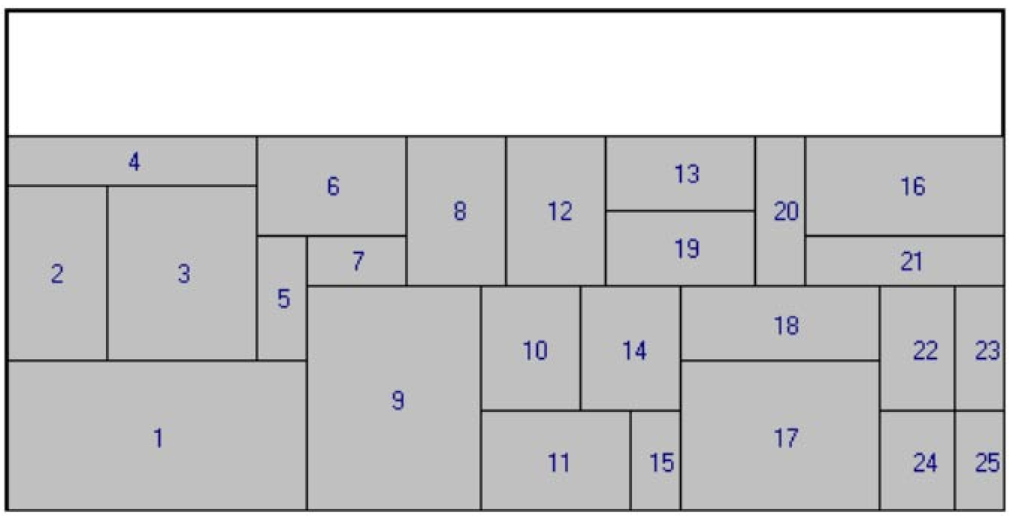


B-IV: Graphical User Interface (User Controls) – Working Prototype

Appendix C – Benchmark Problems[1]

J25

Id	x	y	w	h	Id	x	y	w	h	Id	x	y	w	h
0	0	0	**40**	**30**	11	19	0	6	4	22	35	4	3	5
1	0	0	12	6	12	20	9	4	6	23	38	4	2	5
2	0	6	4	7	13	24	12	6	3	24	35	0	3	4
3	4	6	6	7	14	23	4	4	5	25	38	0	2	4
4	0	13	10	2	15	25	0	2	4					
5	10	6	2	5	16	32	11	8	4					
6	10	11	6	4	17	27	0	8	6					
7	12	9	4	2	18	27	6	8	3					
8	16	9	4	6	19	24	9	6	3					
9	12	0	7	9	20	30	9	2	6					
10	19	4	4	5	21	32	9	8	2					



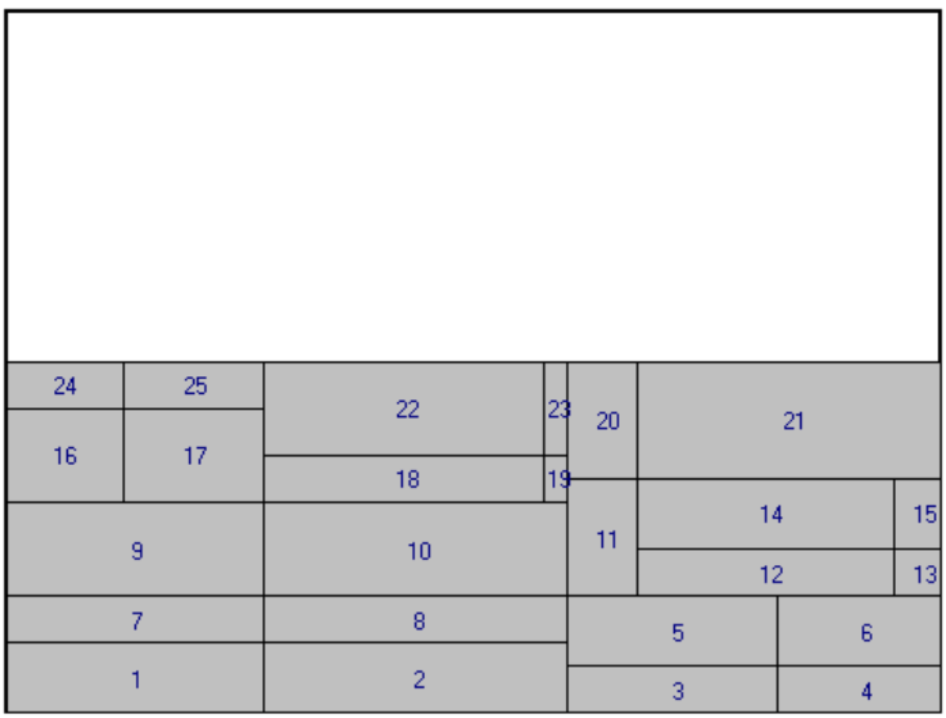
25-Module Problem from Jakobs (1996) and Liu and Teng (1999)

[1] All Problems are shown with trimmed top of bins for ease of visualization.

H25

Id	x	y	w	h	Id	x	y	w	h	Id	x	y	w	h
0	0	0	**40**	**30**	11	0	0	3	5	22	0	0	12	4
1	0	0	11	3	12	0	0	11	2	23	0	0	1	4
2	0	0	13	3	13	0	0	2	2	24	0	0	5	2
3	0	0	9	2	14	0	0	11	3	25	0	0	6	2
4	0	0	7	2	15	0	0	2	3					
5	0	0	9	3	16	0	0	5	4					
6	0	0	7	3	17	0	0	6	4					
7	0	0	11	2	18	0	0	12	2					
8	0	0	13	2	19	0	0	1	2					
9	0	0	11	4	20	0	0	3	5					
10	0	0	13	4	21	0	0	13	5					

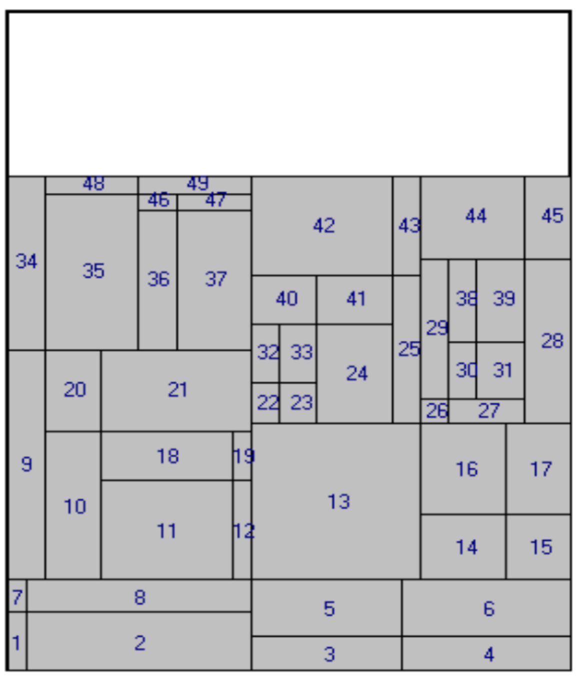
25-Module Problem (C2-P1) from Hopper and Turton (2001)



Note: Hopper and Turton (2001) do not provide any visual depiction of the optimal layout for their benchmark problems. Apparently, their benchmark problems are designed with BLF algorithm in mind. It is reflected from the fact that the ordered sequence of modules (ordered by increasing module index) always resulted in the best layout in terms of the fitness measures used for comparison, namely *HT* (i.e. only one iteration of BLF was needed to obtain optimal). One instance is depicted above for illustration purposes.

H49

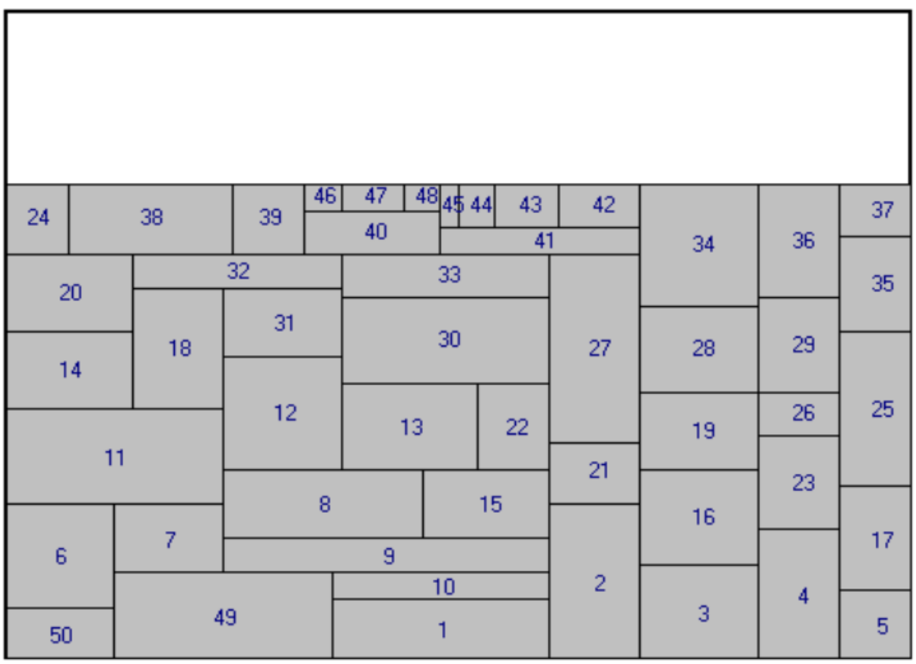
Id	x	y	w	h	Id	x	y	w	h	Id	x	y	w	h
0	0	0	**60**	**210**	18	0	0	14	6	36	0	0	4	17
1	0	0	2	7	19	0	0	2	6	37	0	0	8	17
2	0	0	24	7	20	0	0	6	10	38	0	0	3	10
3	0	0	16	4	21	0	0	16	10	39	0	0	5	10
4	0	0	18	4	22	0	0	3	5	40	0	0	7	6
5	0	0	16	7	23	0	0	4	5	41	0	0	8	6
6	0	0	18	7	24	0	0	8	12	42	0	0	15	12
7	0	0	2	4	25	0	0	3	18	43	0	0	3	12
8	0	0	24	4	26	0	0	3	3	44	0	0	11	10
9	0	0	4	28	27	0	0	8	3	45	0	0	5	10
10	0	0	6	18	28	0	0	5	20	46	0	0	4	2
11	0	0	14	12	29	3.6	19.1	3	17	47	0	0	8	2
12	0	0	2	12	30	0	0	3	7	48	0	0	10	2
13	0	0	18	19	31	0	0	5	7	49	0	0	12	2
14	0	0	9	8	32	0	0	3	7					
15	0	0	7	8	33	0	0	4	7					
16	0	0	9	11	34	0	0	4	21					
17	0	0	7	11	35	0	0	10	19					



49-Module Problem (C4-P1) from Hopper and Turton (2001)

A50

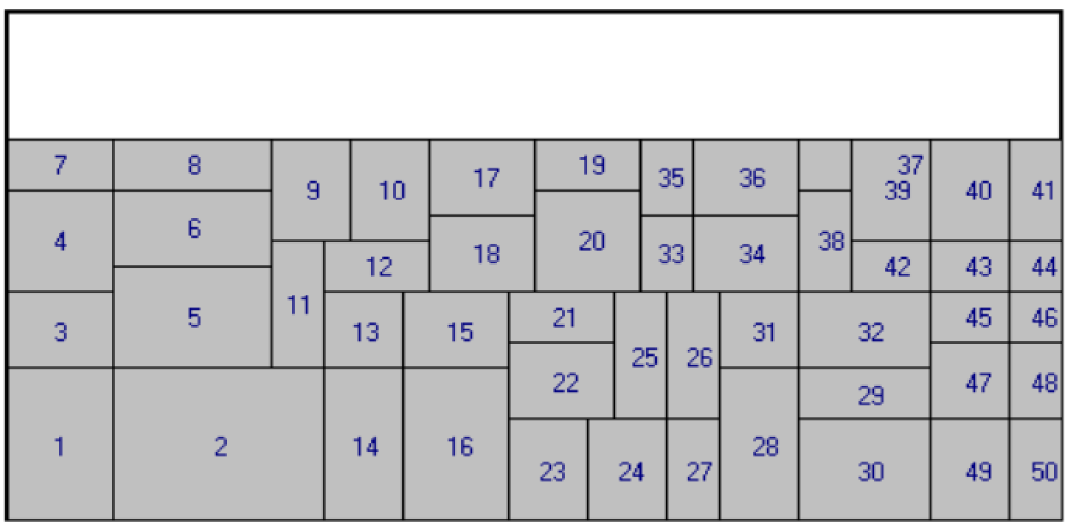
Id	x	y	w	h	Id	x	y	w	h	Id	x	y	w	h
0	0	0	**100**	**200**	18	14	29	10	14	36	83	42	9	13
1	36	0	24	7	19	70	22	13	9	37	92	49	8	6
2	60	0	10	18	20	0	38	14	9	38	7	47	18	8
3	70	0	13	11	21	60	18	10	8	39	25	47	8	8
4	83	0	9	15	22	52	22	8	10	40	33	47	15	5
5	92	0	8	8	23	83	15	9	11	41	48	47	22	3
6	0	6	12	12	24	0	47	7	8	42	61	50	9	5
7	12	10	12	8	25	92	20	8	18	43	54	50	7	5
8	24	14	22	8	26	83	26	9	5	44	50	50	4	5
9	24	10	36	4	27	60	25	10	22	45	48	50	2	5
10	36	7	24	3	28	70	31	13	10	46	33	52	4	3
11	0	18	24	11	29	83	31	9	11	47	37	52	7	3
12	24	22	13	13	30	37	32	23	10	48	44	52	4	3
13	37	22	15	10	31	24	35	13	8	49	12	0	24	10
14	0	29	14	9	32	14	43	23	4	50	0	0	12	6
15	46	14	14	8	33	37	42	23	5					
16	70	11	13	11	34	70	41	13	14					
17	92	8	8	12	35	92	38	8	11					



50-Module Problem from Ahmad *et al.* (2004d)

J50

Id	x	y	w	h	Id	x	y	w	h	Id	x	y	w	h
0	0	0	**40**	**15**	18	16	9	4	3	36	26	12	4	3
1	0	0	4	6	19	20	13	4	2	37	30	13	8	2
2	4	0	8	6	20	20	9	4	4	38	30	9	2	4
3	0	6	4	3	21	19	7	4	2	39	32	11	3	4
4	0	9	4	4	22	19	4	4	3	40	35	11	3	4
5	4	6	6	4	23	19	0	3	4	41	38	11	2	4
6	4	10	6	3	24	22	0	3	4	42	32	9	3	2
7	0	13	4	2	25	23	4	2	5	43	35	9	3	2
8	4	13	6	2	26	25	4	2	5	44	38	9	2	2
9	10	11	3	4	27	25	0	2	4	45	35	7	3	2
10	13	11	3	4	28	27	0	3	6	46	38	7	2	2
11	10	6	2	5	29	30	4	5	2	47	35	4	3	3
12	12	9	4	2	30	30	0	5	4	48	38	4	2	3
13	12	6	3	3	31	27	6	3	3	49	35	0	3	4
14	12	0	3	6	32	30	6	5	3	50	38	0	2	4
15	15	6	4	3	33	24	9	2	3					
16	15	0	4	6	34	26	9	4	3					
17	16	12	4	3	35	24	12	2	3					



50-Module Problem from Jakobs (1996) and Liu and Teng (1999)

H97

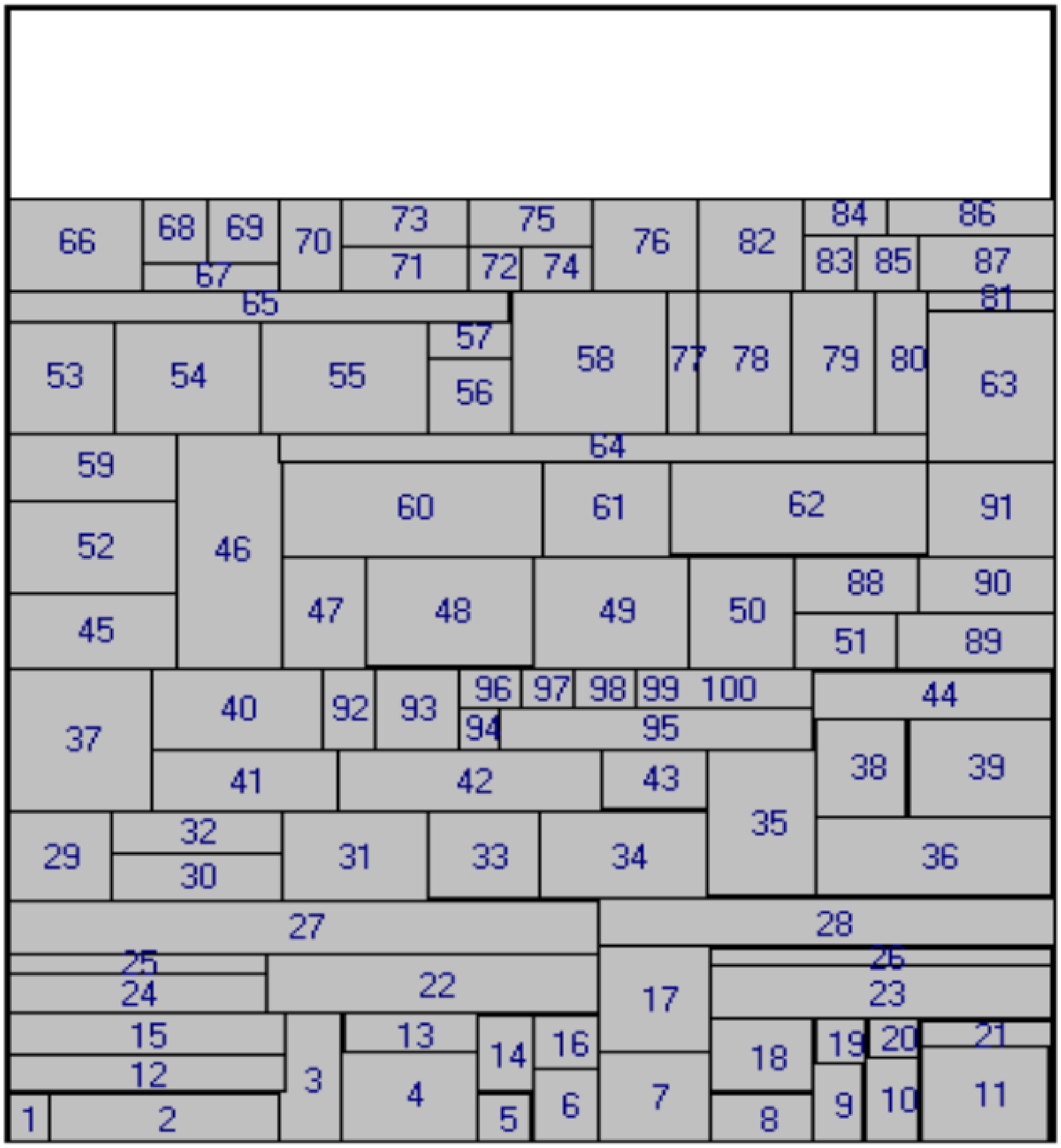
Id	x	y	w	h	Id	x	y	w	h	Id	x	y	w	h
0	0	0	**80**	**150**	34	43	121	6	2	68	27	45	4	26
1	31	0	7	39	35	10	81	15	3	69	50	111	12	6
2	41	42	8	33	36	49	42	30	5	70	67	35	3	1
3	50	117	7	6	37	75	41	8	1	71	73	80	2	3
4	25	115	5	3	38	69	115	10	4	72	71	84	3	2
5	27	71	3	5	39	77	30	2	6	73	76	41	1	1
6	38	0	39	6	40	0	81	6	23	74	33	102	2	3
7	19	102	11	13	41	49	47	29	8	75	45	94	3	2
8	68	91	3	4	42	0	76	26	5	76	25	118	5	2
9	33	105	2	2	43	10	92	9	17	77	6	121	4	4
10	5	117	5	2	44	31	39	7	3	78	57	120	8	2
11	72	6	5	30	45	30	85	19	9	79	70	104	9	11
12	70	35	2	1	46	38	36	36	6	80	10	122	3	2
13	49	61	26	11	47	49	55	28	6	81	14	109	5	11
14	23	45	4	5	48	0	55	6	20	82	62	109	7	9
15	57	118	9	2	49	10	85	20	7	83	38	6	24	30
16	62	6	10	29	50	19	92	11	2	84	77	0	2	11
17	65	121	4	3	51	0	121	6	5	85	30	113	10	8
18	74	36	5	5	52	45	108	5	13	86	0	119	9	2
19	66	119	8	2	53	6	103	4	14	87	35	94	10	2
20	49	80	24	4	54	19	94	16	8	88	40	113	3	11
21	49	84	22	7	55	26	76	23	9	89	6	81	4	22
22	77	21	2	9	56	49	72	26	8	90	19	115	6	9
23	50	108	2	2	57	78	47	1	6	91	69	121	3	3
24	62	35	5	1	58	12	50	15	26	92	71	86	7	18
25	52	96	9	15	59	75	61	4	25	93	0	104	5	15
26	31	42	10	33	60	23	0	8	45	94	61	96	9	13
27	74	41	1	1	61	12	0	11	50	95	77	11	2	10
28	74	119	4	4	62	49	91	19	5	96	25	121	6	5
29	37	121	3	3	63	0	0	12	55	97	35	96	17	5
30	14	120	4	4	64	6	55	5	20					
31	30	102	3	6	65	10	109	4	13					
32	35	101	16	7	66	30	108	15	5					
33	31	121	6	4	67	77	55	2	6					

97-Module Problem (C6-P2) from Hopper and Turton (2001)

A100

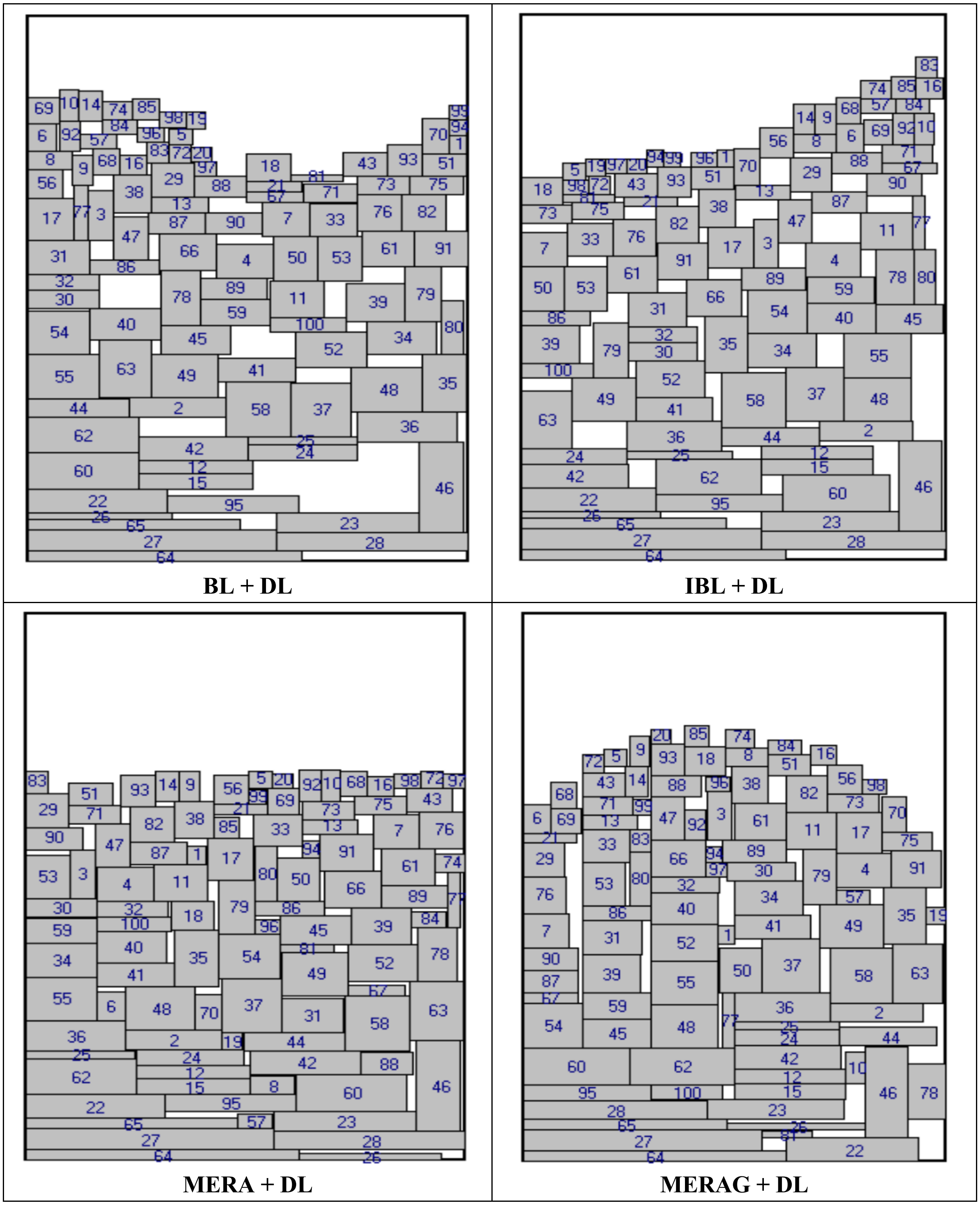
Id	x	y	w	h	Id	x	y	w	h	Id	x	y	w	h
0	**0**	**0**	**100.0**	**150.0**	42	31.7	35.0	25.0	6.5	84	76.0	96.0	8.0	4.0
1	0.0	0.0	4.0	5.0	43	56.9	35.2	10.1	6.5	85	81.0	90.0	6.0	6.0
2	4.0	0.0	22.0	5.0	44	76.9	44.8	22.9	5.1	86	84.0	96.0	16.0	4.0
3	26.0	0.0	6.0	13.5	45	0.2	50.0	16.0	8.0	87	87.0	90.0	13.0	6.0
4	32.0	0.0	13.0	10.0	46	16.2	50.0	10.0	25.0	88	75.3	56.0	12.0	6.0
5	45.0	0.0	5.0	5.0	47	26.2	50.0	8.0	12.0	89	85.0	50.0	15.0	6.0
6	50.2	0.0	6.5	7.8	48	34.2	50.2	16.0	12.0	90	87.0	56.0	13.0	6.0
7	56.5	0.0	10.6	9.5	49	50.2	50.0	15.0	12.0	91	88.0	62.0	12.0	10.0
8	67.2	0.0	10.0	5.0	50	65.2	50.0	10.0	12.0	92	30.0	41.5	5.0	8.5
9	77.3	0.2	4.8	8.2	51	75.2	50.0	10.0	6.0	93	35.0	41.5	8.0	8.5
10	82.0	8.0	5.0	9.0	52	0.2	58.0	16.0	10.0	94	43.0	41.5	4.0	4.5
11	87.2	0.0	12.2	10.1	53	0.2	75.0	10.0	12.0	95	47.0	41.5	30.0	4.5
12	0.2	5.3	26.2	3.8	54	10.2	75.0	14.0	12.0	96	433.0	46.0	6.0	4.0
13	32.2	9.3	12.8	4.1	55	24.2	75.0	16.0	12.0	97	49.0	46.0	5.0	4.0
14	45.0	5.1	5.2	8.2	56	40.2	75.0	8.0	8.0	98	54.0	46.0	6.0	4.0
15	0.2	9.2	26.2	4.2	57	40.2	83.0	8.0	4.0	99	60.0	46.0	4.0	4.0
16	50.3	7.7	6.2	5.5	58	48.2	75.0	14.8	15.0	100	60.0	46.0	17.0	4.0
17	56.6	9.4	10.5	11.3	59	0.2	68.0	16.0	7.0					
18	67.3	5.1	9.8	7.7	60	26.2	62.0	25.0	10.0					
19	77.4	8.2	4.6	4.6	61	51.2	62.0	12.0	10.0					
20	82.2	8.9	4.8	4.0	62	63.2	62.3	25.0	9.8					
21	87.2	10.1	12.5	2.6	63	88.0	72.0	12.0	16.0					
22	24.9	13.5	31.5	6.4	64	26.0	72.0	62.0	3.0					
23	67.2	13.0	32.5	5.5	65	0.0	87.0	48.0	3.0					
24	0.2	13.5	24.7	4.3	66	0.0	90.0	13.0	10.0					
25	0.0	17.8	24.9	2.2	67	13.0	90.0	13.0	3.0					
26	67.1	18.5	32.6	1.9	68	13.0	93.0	6.0	7.0					
27	0.2	19.9	56.4	5.6	69	19.0	93.0	7.0	7.0					
28	56.6	20.7	43.4	5.0	70	26.0	90.0	6.0	10.0					
29	0.2	25.5	9.6	9.4	71	32.0	90.0	12.0	5.0					
30	10.0	25.5	16.2	5.1	72	44.0	90.0	5.0	5.0					
31	26.2	25.4	13.9	9.4	73	32.0	95.0	12.0	5.0					
32	9.8	30.5	16.3	4.4	74	49.0	90.0	7.0	5.0					
33	40.2	25.8	11.0	9.0	75	44.0	95.0	12.0	5.0					
34	51.0	25.7	16.0	9.2	76	56.0	90.0	10.0	10.0					
35	67.0	26.0	10.2	15.6	77	63.0	75.0	3.0	15.0					
36	77.2	26.0	22.5	8.2	78	66.0	75.0	9.0	15.0					
37	0.0	35.0	14.0	15.0	79	75.0	75.0	8.0	15.0					
38	77.2	34.4	8.8	10.4	80	83.0	75.0	5.0	15.0					
39	86.1	34.4	13.6	10.4	81	88.0	88.0	12.0	2.0					
40	13.9	41.5	16.1	8.5	82	66.0	90.0	10.0	10.0					
41	13.9	35.0	17.8	6.5	83	76.0	90.0	5.0	6.0					

100-Module Problem from Ahmad *et al.* (2004d)



100-Module Problem from Ahmad *et al.* (2004d)

Appendix D – Visual Comparison of Placement Algorithms



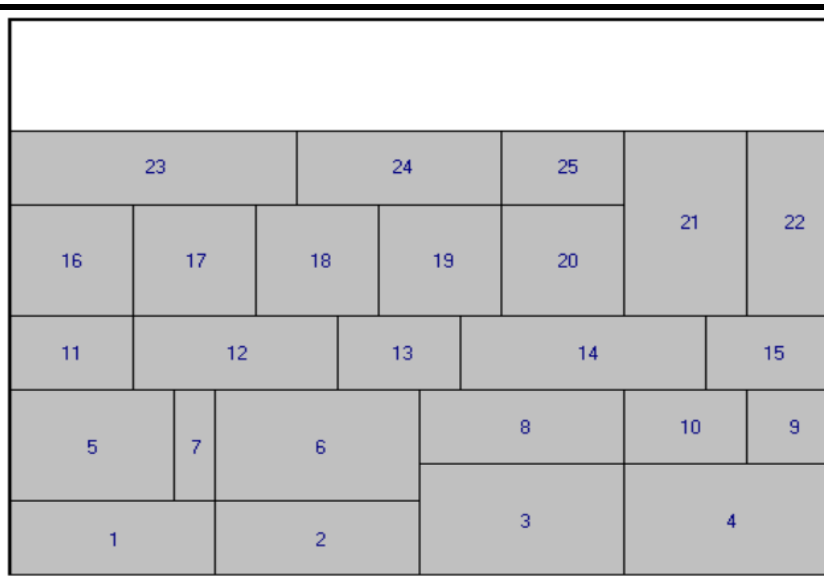
Performance of various algorithms for the 100-Module (A100) Problem, Appendix C, in terms of *CR*

Appendix E – Training & Testing Data used for PDA

D-I (Training Data for PDA)							
Instance #	Module Tightness	Symmetry	Expert's Rating	Instance #	Module Tightness	Symmetry	Expert's Rating
	Inputs		Target		Inputs		Target
	X1	X2	T		X1	X2	T
1	**70.8**	**78.75**	**60**	31	**74.3**	**82.21**	**70**
2	**85.7**	**73.8**	**65**	32	**90.2**	**70.54**	**60**
3	**70.6**	**69.38**	**35**	33	**83**	**78.37**	**75**
4	**75**	**75.1**	**50**	34	**70.6**	**69.87**	**30**
5	**75**	**69.18**	**35**	35	**70.6**	**68.3**	**30**
6	**80**	**75.12**	**60**	36	**74.3**	**81.26**	**70**
7	**80**	**71.74**	**50**	37	**84.2**	**77.18**	**70**
8	**85.7**	**65.14**	**35**	38	**70.6**	**75.75**	**55**
9	**85.7**	**63.4**	**35**	39	**78.9**	**80.19**	**75**
10	**85.7**	**65.14**	**40**	40	**83**	**83.3**	**80**
11	**75**	**70.95**	**45**	41	**70.2**	**87.14**	**75**
12	**75**	**71.74**	**45**	42	**78.4**	**92.8**	**70**
13	**85.7**	**59.17**	**30**	43	**74.3**	**73.85**	**50**
14	**80**	**66.91**	**30**	44	**74.1**	**85.57**	**70**
15	**80**	**71.74**	**55**	45	**83**	**84.41**	**85**
16	**80**	**72.36**	**55**	46	**83**	**90.52**	**85**
17	**85.7**	**65.14**	**40**	47	**83**	**90.52**	**90**
18	**84.2**	**72.65**	**60**	48	**78.4**	**77.51**	**65**
19	**90.2**	**71.74**	**75**	49	**88.2**	**80.2**	**75**
20	**90.2**	**76.69**	**70**	50	**78.4**	**84.41**	**80**
21	**84.23**	**67.2**	**45**	51	**83**	**89.79**	**85**
22	**90.2**	**68.31**	**55**	52	**83**	**79.26**	**70**
23	**84.2**	**77.18**	**65**	53	**78.4**	**85.72**	**85**
24	**74.3**	**70.89**	**40**	54	**83**	**81.71**	**75**
25	**90.2**	**70.53**	**65**	55	**78.9**	**80.1**	**65**
26	**90.2**	**70.9**	**65**	56	**85.7**	**60.47**	**25**
27	**84.2**	**81.26**	**85**	57	**78.9**	**70.95**	**45**
28	**90.2**	**80.1**	**85**	58	**75**	**73.65**	**50**
29	**78.9**	**68.82**	**45**	59	**88.2**	**86.82**	**80**
30	**78.9**	**77.11**	**50**	60	**75**	**71.1**	**40**

D-II (Test Data for PDA)							
Instance #	Module Tightness	Symmetry	Expert's Rating	Instance #	Module Tightness	Symmetry	Expert's Rating
	Inputs		Target		Inputs		Target
	X1	X2	T		X1	X2	T
1	**78.4**	**72**	**40**	11	**75**	**75.75**	**50**
2	**70.6**	**70.92**	**40**	12	**74.1**	**78.55**	**65**
3	**70.6**	**65.09**	**25**	13	**74.1**	**88.7**	**80**
4	**90.2**	**79.92**	**60**	14	**83.3**	**78.37**	**75**
5	**74.3**	**77.11**	**55**	15	**88.2**	**89.79**	**90**
6	**84.2**	**69.51**	**45**	16	**85.7**	**60.91**	**30**
7	**84.2**	**63.26**	**35**	17	**90.2**	**74.47**	**65**
8	**78.9**	**76.42**	**60**	18	**78.9**	**80.1**	**70**
9	**78.4**	**79.3**	**75**	19	**88.2**	**71**	**60**
10	**78.9**	**78.75**	**70**	20	**84.2**	**83.3**	**75**

D-III (Problem Instance Used in Generating Alternatives for PDA Training Data)



VITA

Abdul-Rahim Ahmad received his B.E. degree in Electrical Engineering from the department of Electrical & Computer Engineering, N.E.D. University of Engineering & Technology, Pakistan, in 1993. In 1997, he received his M.S. degree in Systems Engineering (Industrial and Operations Management), King Fahd University, Saudi Arabia. In 2002, he completed requirements for his MASc degree in Management Sciences, University of Waterloo, Canada.

Abdul-Rahim's cross-functional research interests include Decision Modeling, Operations Management, Information Systems, Neural Networks, Fuzzy Logic, Genetic Algorithms, Expert Systems, Decision Support Systems, Heuristics Design, Layout Decision Analysis & Design, and Web Merchandizing. His doctoral studies at the University of Waterloo have provided him the opportunity to explore new theoretical concepts in soft computing and decision modeling/analysis and apply those concepts in practical applications. His research work has appeared in several reputed journals and international conferences.

Abdul-Rahim's interdisciplinary teaching interests are in Expert Systems, Decision Support Systems, Human-Computer Interaction Design & Management, Operations Management, Decision Modeling, E-Commerce, and Telecommunications Management. He taught various courses in Industrial and Operations Management as a Teaching Assistant in the department of Systems Engineering, King Fahd University, Saudi Arabia, from 1994 to 1997. From 1998 to 2000, he taught several courses in Information Systems as a Lecturer in the department of Management Information Systems, King Fahd University, Saudi Arabia. From 2000 to 2002, he had been involved in teaching courses in Human-Computer Interaction Design and Management, Managerial & Engineering Economics, and Organizational Behavior & Structure in the department of Management Sciences, University of Waterloo. From 2002 to 2005, he has been involved in teaching courses in Managerial & Engineering Economics, Numerical Analysis, Decision Analysis & Digital Logic Design, and Electrical Circuits & Instrumentation. In 2005, he offered courses in Business Decision Models & Software Applications, Operations Management, and Management Information Systems in Operations and Decision Sciences, Wilfrid Laurier University, Canada. He received the Distinguished Teaching Award from the University of Waterloo in 2004 as well as other teaching honors from various institutions.

Abdul-Rahim has worked as a Design Engineer in Precision Technology, Pakistan. He has served as a consultant in design and analysis for Engineering Optimization Software Inc., USA and GeoDynamics Engineering Inc., Canada. He has also been a consultant in academic affairs and curriculum design for Sir Syed University of Engineering and Technology, Pakistan.

Abdul-Rahim plans to graduate with his Ph.D. in Systems Design Engineering (Expert & Knowledge-based Systems) from the University of Waterloo in 2005. Upon graduation, Abdul-Rahim will join the department of Management Information Systems, College of Industrial Management, King Fahd University, Saudi Arabia.

His permanent e-mail is ar.ahmad@gmail.com and permanent postal address is E#9, Karim Plaza, Block#14, Ghulshan-e-Iqbal, Karachi#75300, Pakistan.