12

An Intelligent Expert Systems' Approach to Layout Decision Analysis and Design under Uncertainty

Abdul-Rahim Ahmad[1], Otman Basir[1], Khaled Hassanein[2], and Shahid Azam[3]

[1] Systems Design Engineering, University of Waterloo, Canada {arahim, obasir}@engmail.uwaterloo.ca
[2] School of Business, McMaster University, Hamilton, Canada hassank@mcmaster.ca
[3] Environmental Systems Engineering, University of Regina, Canada Shahid.azam@uregina.ca

Summary. This chapter describes an intelligent soft computing based approach to layout decision analysis and design. The solution methodology involves the use of heuristics, metaheuristics, human intuition as well as soft computing tools like artificial neural networks, fuzzy logic, and expert systems. The research framework and prototype contribute to the field of intelligent decision making in layout analysis and design by enabling explicit representation of experts' knowledge, formal modeling of fuzzy user preferences, and swift generation/manipulation of superior layout alternatives to facilitate the cognitive, ergonomic, and economic efficiency of layout designers.

12.1 Introduction

The Layout Design (LD) process is geared towards seeking some superior outcome in the spatial arrangement of modules in a given space while satisfying a set of given preferences and constraints. A generic approach to the LD problem is to treat it as an oriented and orthogonal two-dimensional rectangular packing problem (2D-BPP). In this problem, n rectangular modules of length L_i and width W_i $(i = 1, 2, \ldots, n)$ are to be packed on a large rectangular packing space of length L_o and width W_o without overlaps and within the boundary constraints (Dyckhoff 1990, Garey and Johnson 1979). Each module i is of fixed orientation and must be packed with its edges parallel to the edges of rectangular packing space. Each module i is associated with a utility u_i and the objective is to maximize the total utility of the packing pattern. This problem is relevant to various facilities planning, cutting, packing, storing, transporting, scheduling, and resource allocation functions of businesses

A.-R. Ahmad et al.: *An Intelligent Expert Systems' Approach to Layout Decision Analysis and Design under Uncertainty*, Studies in Computational Intelligence (SCI) **97**, 321–364 (2008)
www.springerlink.com

(Islier 1998, Lodi et al. 2002, Martens 2004). Only in facilities planning area, US businesses spend about a trillion dollars in new facilities annually and their layouts directly affect more than 20% of the operating costs (Ahmad 2005). Thus, the research efforts in improving the efficiency and efficacy of tools for layout decision analysts and decision-makers are imperative and ongoing.

The LD is a tedious process that calls for sophisticated decision analysis and design support. The existing solution approaches largely employ very rigid and overly simplistic design algorithms and guidelines, largely without an elaborate methodology for their utilization. Nevertheless, the complex, subjective, uncertain, and evolving nature of layout design preferences and fitness objectives means 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 LD problem lack the requisite flexibility, efficacy, and robustness, as discussed in detail in the subsequent sections (Abdinnour-Helm and Hadley 2000, Ahmad 2005, Badiru and Arif 1996, Osman et al. 2003). The situation is further complicated by the high cognitive overhead encountered by layout designers 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 have rendered powerful soft computing tools for tackling with such complex and uncertain problems as layout design. Such alternatives include an array of emerging computing disciplines such as Decision Support Systems, Expert Systems, Fuzzy Logic, Neural Networks, Genetic Algorithms, and hybrids like Neuro-Fuzzy-Genetic systems (Ahmad 2005, Karray and De Silva 2004). These technologies share the common denominator in their digression from classical reasoning and modeling approaches through a set of more flexible computing tools (Negnevitsky 2002). Such approaches are gaining favor in modeling cognition and intelligent systems, as the underlying procedures are most analogous to human reasoning (Ahmad 2002, Akoumianakis et al. 2000, Zadeh 1999). Such technologies have demonstrated the power and philosophy to solve complex and ill-defined problems, offering significant potential in dealing with the LD problem.

In this chapter, a promising research framework for an Intelligent System for Decision Support and Expert Analysis in Layout 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 preferences as a key to enhancing productivity of layout designers. Instead of pursuing some perfect methods, our emphasis is 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. Our approach involves tackling various important aspects of the problem through a synergistic utilization of some promising soft computing techniques, advanced heuristics, and metaheuristics.

The rest of the chapter is organized as follows. Section 12.1 provides motivation for our research. Section 12.2 presents a brief literature review of some relevant faculties and their significance in this research. Section 12.3 provides an overview of traditional approaches to the LD problem. Section 12.4 provides a survey of intelligent and knowledge-based approaches to the LD problem. Section 12.5 delineates the proposed solution paradigm and its various major constituents. Section 12.6 outlines results of some case studies undertaken to test the effectiveness of the proposed paradigm. Section 12.7 lists some promising research directions. Section 12.8 concludes the chapter.

12.2 Literature Survey

The diverse scope of the LD problem means that a substantial literature is available in a variety of work domains (Abdinnour-Helm and Hadley 2000, Ahmad 2005, Akoumianakis et al. 2000, Burke et al. 2004, Karray et al. 2000, Tompkins et al. 2002, Youssef et al. 2003). This problem has been variously referred to as *topology optimization* (Mir and Imam 1992), *block placement* (Ahmad 2005), *macro cell placement* or *VLSI layout design* (Schnecke and Vonberger 1997), *layout optimization* (Cohoon et al. 1991), *facilities layout* (Tam et al. 2002), *plant layout* or *machine layout* (Hassan and Hogg 1994), *bin-packing* (Jakobs 1996), *partitioning* (Moon and Kim 1998), etc. However, we may classify LD problems into four major application categories including 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.

In facilities LD, various activities and components are allocated spaces in the given periphery (Abdinnour-Helm and Hadley 2000). The resulting layout of facility establishes the physical relationship among activities and their objectives (Badiru and Arif 1996, 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 and Gau 1996). Indeed, an adequately designed facility layout improves the efficiency, efficacy, productivity, and profitability of an organization (Norman and Smith 2002). The relative permanency of outcome and the scale of strategic investment stipulations mean more research efforts have been dedicated to facility LD than any other LD area.

The bin-packing problem is directed at packing a greater number of items in the smallest number of fixed size bins (Dyckhoff 1990). As such, the typical goal is to maximize the space utilization (Kim et al. 2001). Among the several variants of general bin-packing problem, we limit ourselves to the oriented and orthogonal two-dimensional rectangular packing problem (2D-BPP). This problem provides a basis for devising a generic approach to 2-D layout design and used for elaborating our research paradigm.

The design of VLSI microchips involves several phases including functional design, circuit design, physical design, and fabrication (Mazumder and Rudnick 1999). An important step in physical design is the macrocell placement based on a range of subjective and conflicting preferences and constraints (Moon and Kim 1998). Macrocells are the circuit components lumped together in functional entities with connection terminals along their borders. These terminals are connected by signal nets, along which signals or power is transmitted among the various components. As such, the macrocell layout also characterizes routes selected for the signal nets. During the macrocell placement phase, an estimated amount of routing space or white space is added between the cells.

12.2.1 Popular Approaches to Mathematical Formulations

A range of formulations for the LD problem has been proposed in the literature and a good account of these can be found in (Ahmad 2005, Bozer and 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 (Ahmad 2005).

QAP formulations deal with decisions regarding location of equal area modules. This approach works by assigning one module to every location and at most one module to a given location. Due to NP-Complete nature, it is very hard to procure a verifiably optimal solution for more than 16 modules (Meller and Gau 1996).

QSC formulation requires data on the size of each module, candidate locations of each module, and utilities of each module. QSC allows layout designers to introduce candidate locations of each modules, which 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 (Bazaraa 1975, Ligget 2000).

The LD problem may also be formulated as an oriented and orthogonal 2D-BPP. It has the advantage of maintaining the integrity and the shape of modules. Such a formulation requires minimal post-optimization processing in comparison with other prevailing LD problem formulations. Furthermore, it constitutes a generic approach to many LD problems (Ahmad 2005, Burke et al. 2004, Dyckhoff 1990, Garey and Johnson 1979, Lodi et al. 2002).

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 and 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 and 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 and Huang 2000, Mir and 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).

12.3 Traditional Solution Approaches

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 a QAP (Bazaraa 1975, Wu et al. 2002). The 2-dimensional plane is discretized into a grid structure, which results in high computational costs (Gloria et al. 1994). Other solution approaches include tree search algorithms (Pierce and Crowston 1971), binary mixed integer-programming (Love and Wong 1976), and network decomposition (Mak et al. 1998) etc. The NP-Hard and subjective nature of the LD problem means that traditional hard optimization approaches do not hold much promise. Nevertheless, 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.

12.3.1 Algorithmic Approaches

Here we discuss some popular algorithmic approaches to solving layout design problem.

The development of a layout through a *Graph* based 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 non-identical 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 and Hogg 1994).

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. A tree search method may employ a either breadth-first search by enumerating all possible ways of adding a new module or depth-first search by creating a full layout by placing all the modules sequentially (Akin et al. 1997). However, such an approach is inherently inefficient and requires frequent backtracking when feasibility constraints are violated, which adds to the computational complexity (Ligget 2000).

There are various *analytical techniques* dealing with continuous design space with relatively minimal computational requirements (Adya et al. 2003, Mir and Imam (1992, 1996, 2001), Tam et al. 2002, Welgama et al. 1995). However, analytical approaches have yet to be developed to furnish results comparable to advanced heuristic/metaheuristic techniques. Nevertheless, these provide more insights to the structure of the problem leading to advanced and effective heuristics.

12.3.2 Metaheuristic Approaches

Decision-makers often resort to heuristics for dealing with difficult and uncertain problems. Similarly, 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 2005, Dowsland et al. 2002, El-Bouri et al. 1994, Hopper and Turton 2001, Jakobs 1996, Kim et al. 2001, Leung et al. 2003, Liu and Teng 1999, Lodi et al. (1999, 2002), Martens 2004). In this regard, the importance of effectively limiting an intractable search space to some reasonable subset of possible solution topologies cannot be overemphasized (Dowsland et al. 2002, 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 2005, Leung et al. 2003). Recent metaheuristics that have shown good results for LD include simulated annealing (Adya et al. 2003), genetic algorithms (Ahmad 2005, Gloria et al. 1994, Martens 2004), tabu search (Hopper and Turton 2001), random search (Ahmad 2005, Jakobs 1996, Liu and Teng 1999), naive evolution Hopper and Turton 2001, and hybrids (Ahmad 2005, Lee and 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 and 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 and Turton 2001, Leung et al. 2003, Youssef et al. 2003). 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 method. Consequently, here we discuss some merits and demerits to provide some insights to these popular metaheuristics.

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 (Ahmad et al. 2006, Youssef et al. 2003). GA allow incorporation of domain-specific knowledge into the fitness of individual solutions as well as in genetic selections and operations (Youssef et al. 2003). Moreover, GA creates a population of optimized solutions.

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 and Turton 2001). As such, GA provide a very promising approach for LD through generation of a diverse set of superior alternatives (Ahmad 2005, Lee and Lee 2002, Martens 2004, Moon and Kim 1998). Further advantages of GA within the context of LD are discussed in Sect. 12.5.1.

Simulated Annealing (SA) is a well-known, high-performance, and effective stochastic optimization technique for combinatorial problems (Mir and Imam 2001, Tam et al. 2002). Any domain specific knowledge is incorporated mainly in the SA cost function (Youssef et al. 2003). 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. 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. 2003). However, SA is generally very slow to converge to good solutions when compared to GA. SA may provide solutions to an LD problem that is comparable to or marginally better than GA (Hopper and Turton 2001, Youssef et al. 2003). 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 2005).

Tabu Search (TS) is another successful, effective, and robust metaheuristic approach for solving complex combinatorial and continuous optimization problems. In a generic sense, TS is an iterative procedure that starts from some initial feasible solution and attempts to determine a better solution 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.

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. 2003). The tabu list is analogous to a window on accepted moves that permit the search beyond the points of local optimality while making the best possible move.

Naive Evolution (NE) search is somewhat similar to GA in its basic form. However, it employs only a mutation operator in order to generate successive populations of solutions. Understandably, it is very easy to implement but lacks the structured search engendered by crossover operators in GA. 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 and Turton 2001).

Random Search (RS) is another naive search strategy where the ordering of modules is generated randomly (Ahmad 2005, Ahmad et al. 2006, Hopper and 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. 2003).

12.3.3 Heuristic Approaches

The combinatorial complexity of the LD problem formulations has lead to development of various efficient heuristics, which may be used alone or in conjunction with metaheuristics. Indeed, metaheuristics 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 module at a time from a sequence of modules and determines its position in the packing space based on pre-specified steps, usually designed to realize some local improvements in the search process (Healy et al. 1999, Youssef et al. 2003). An efficient placement strategy that generates superior layouts is critical for the efficacy of such an endeavor (Dowsland et al. 2002). Here we discuss some of the 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 and Turton 2001). In Sect. 12.5.1, we provide some a new decoding heuristic and demonstrate its efficiency and efficacy.

The *Bottom-Left* (BL) placement algorithm calls for placing a module at the bottom-most and left-most feasible position through successive vertical and horizontal movements of the module (Ahmad et al. 2006, Chazelle 1983, Dowsland et al. 2002, Healy et al. 1999, Hopper and Turton 2001, Jakobs 1996, Liu and Teng 1999). 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 (Jakobs 1996). The apparent advantages of such approaches include speed and simplicity (Dowsland et al. 2002). However, BL tends to leave holes in the packing rendering poor space utilization.

Various improvement schemes have been proposed for the BL such as the *Improved-BL heuristic* (IBL) (Liu and Teng 1999). Such improved strategies give precedence to a shift towards the bottom and allow module rotations. However, even these improvised strategies encounter such problems as dead-area and inferior aesthetic contents.

The *Bottom-Left Fill* (BLF) placement algorithm is a more sophisticated version of BL, attempting to fill empty spaces by placing a module into the lowest available position and maintaining a list of candidate placement locations. Consequently, BLF overcomes the problem of poor space utilization. Nevertheless, the major disadvantage lies in its $O(n\hat{}3)$ time complexity (Burke et al. 2004, Chazelle 1983, Hopper and Turton 2001).

The BL and the IBL are overly simplistic heuristics with inherent deficiencies such as poor space utilization. The optimal packing configuration may be obtained by the BL even after exhaustive enumeration (Jakobs 1996). In addition, the BL, the IBL, and the BLF are not very effective in incorporating qualitative considerations such as the layout symmetry and aesthetics. Further, these algorithms are more appropriate for the minimization of the packing height. Consequently, the quest for efficient and effective module placement strategies is an interesting and popular research direction (Burke et al. 2004).

12.4 Intelligent and Knowledge-Based Approaches

Intelligent and knowledge-based approaches are very promising in the LD area. Here we provide a discussion on the promise of these approaches.

12.4.1 Decision Support Systems

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 (Turban and Aronson 2001). Consequently, the generation of superior layout alternatives in a flexible and automated manner is critical to any LD process (Turban and Aronson 2001). Conceivably, some DSS mechanism could be beneficial in solving the LD problem.

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 and Aronson 2001). 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 and Aronson 2001). 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.

Foulds (1995) describes a system called LayoutManager that is reportedly deemed a decision support system in facilities planning. LayoutManager permits users to select the layout design algorithm and other necessary starting conditions. 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 rigid fitness metric. Further alternatives may be generated through trial and error. The deterministic layout design heuristics, based on graph search, do not allow diversified and extensive search of the solution space. The LayoutManager does not provide any means for giving users any real control over the proceedings. Furthermore, 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. In short, the system lacks the flexibility, efficiency, efficacy, scalability, and robustness that would be logical requisites for a DSS in LD.

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 reported in Tommelein 1997 provides many useful insights and future research directions in this field.

12.4.2 Expert Systems

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 (Tommelein 1997). Consequently, the generation of superior layout alternatives in a flexible and automated manner is critical to any layout planning process. Conceivably, some DSS mechanism could be beneficial in solving the LD problem.

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 (Tompkins et al. 2002) ES are usually characterized by the existence of a large repository of knowledge for solving problems in a very constricted work domain (Malakooti and Tsurushima 1989). Such a knowledge repository may comprise of human

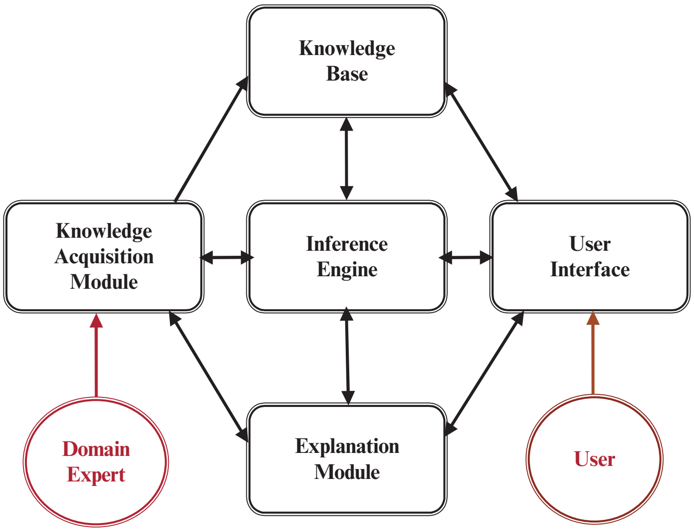

Fig. 12.1. A typical expert system

knowledge and expertise formulated as specific rules and heuristics (Jackson 1999). The distinguishing feature between ES and DSS is the separation of knowledge and the reasoning method involved in an ES, resulting in greater modularity in the system (Negnevitsky 2002). As such, ES afford 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 behind a decision.

A traditional ES is shown in Fig. 12.1. 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). The details about individual components and their synergy follow in Sect. 12.5 within the context of the proposed intelligent system for decision support and expert analysis in layout design. An ES designed specifically to aid decision makers continuously increases productivity, lowers costs, and spurs innovation (Ahmad 2005). However, existing literature on the application of the ES paradigm in LD is quite meager. In addition, such systems have considerable shortcomings, summarized as follows.

Fisher and Nof (1984) present a FAcilities Design Expert System (FADES) for machine LD applications. The reported prototype contains various heuristics and an inferencing mechanism to select a heuristic appropriate for the given scenario. Knowledge is represented using first-order predicate logic. 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.

Kumara et al. (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 steps, which means that it neither affords any actual optimization nor furnishes any diversity in alternatives. IFLAPS requires a significantly high degree of user inputs and interventions and it does not provide functionalities to modify or refine the alternative generated by the system.

Malakooti and 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 solutions do not exhibit diversity. Further, 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.

Heragu and Kusiak (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 is an ES that generates layouts for temporary facilities on construction sites (Tommelein 1997). However, it neither provides ways to incorporate soft constraints and preferences nor it cannot handle conflicting preferences and requires user to manually rectify conflicts. In addition, the layout solutions do not have any diversity, a key requirement in providing design support to LD experts.

12.4.3 Limitations of Existing Knowledge-Based Approaches

Most existing Knowledge-based Layout Design (KBLD) systems are not very robust and flexible, as users might want or as the state of affairs might require.

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, these 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, the time required to solve the problem through these systems could be prohibitive. 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. Nevertheless, despite some claims to the contrary, 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: The quality of solution alternatives is another core issue in layout decision analysis and design. The deterministic nature of LD algorithms and the lack of diversity in decision alternatives mean that the existing KBLD systems require many reruns before a satisficing layout alternative is obtained. The primary reason is the difficulty in modeling sub-cognitive and implicit preferences as well as difficulty in quantifying the qualitative determinants of layout fitness.

Transparency: The existing KBLD systems offer little or no explanation facilities. Towards this end, simply providing the sequence of the rules employed in reaching a decision may still be considered sufficient. However, relating the accumulated heuristic knowledge to deeper understanding of the domain is still elusive.

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 (Walenstein 2002). 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 do have an ability to learn from experience and 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 affords effective and interactive analysis and design. Apparently, the LD practitioners themselves designed most interfaces. Thus, these lag considerably in interactivity, usability, and suitability to the work domain.

12.5 Proposed Intelligent Approach to Layout Design

It has been noted that the computer-based layout design 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). Nevertheless, it is often effortless for experts to visually inspect a layout alternative and endorse its acceptability or otherwise. Conceivably, there are strong prospects for devising some incomplete models and intelligent methods to supplement human erudition and intuition. For instance, computerized generations of alternate layouts could provide efficacious support to the layout analyst by 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 (Bazaraa 1975). The 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 in various LD optimization tools is depicted in Fig. 12.2 [17].

In Fig. 12.2, on one end are enumerative search techniques, which are superior in terms of exhaustiveness of solution space search. However, such general techniques incorporate very meager amount of problem-specific information and their application is marred by the process speed and computational

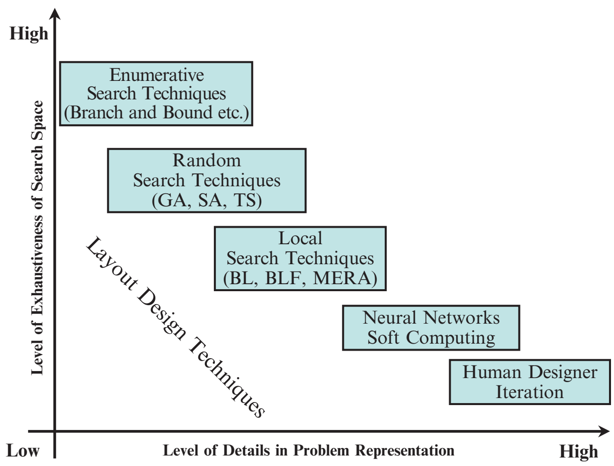

Fig. 12.2. Flexibility and robustness of various layout design approaches (Ahmad 2005, Chung 1999)

complexity. On the other end, human designers command high level of flexibility and the capability of incorporating detailed problem-specific information into the design process. However, the cognitive and information processing limitations of human designers translate into inadequate 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 and efficacy.

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 (Ahmad et al. 2004). In this regard, computerized decision support tools may be viewed as a mechanism for *redistribution of cognition* (Welgama et al. 1995). Such tools provide support through various means such as *process distribution, data distribution, plan distribution*, etc. (Walenstein 2002).

Our research framework is based on the Expert System (ES) paradigm for facilitating intelligent decision support in layout design. The emphasis of this research 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 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. A traditional ES paradigm is shown in Fig. 12.2. However, an efficient and effective means of tackling the subjectivity and uncertainty in the layout design problem requires complementing the traditional ES paradigm through various intelligent components. Such intelligent components 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 Fig. 12.3, 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 research as more insights are gained about the structure of the problem at hand and the underlying dynamics.

As mentioned, an array of efficient algorithms for generating superior and diverse layout alternatives is an important step in automating the layout design process. We use a hybrid fuzzy-genetic Intelligent Layout Generator (ILG) towards this end. The intelligence aspect emerges from the

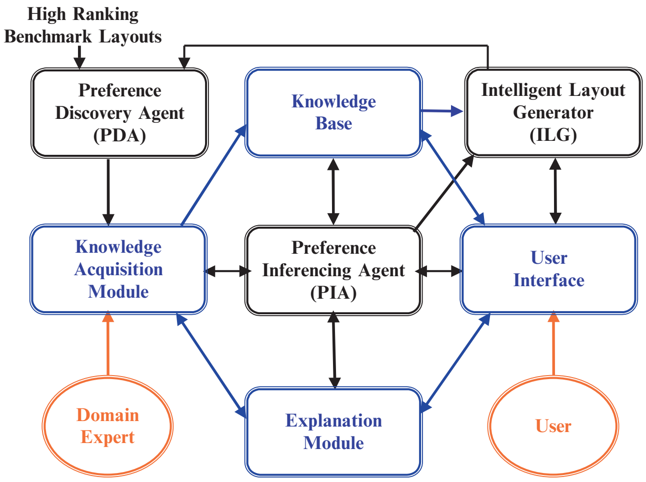

Fig. 12.3. Intelligent System for Decision Support/Expert Analysis in Layout Design

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 a decision-aid tool.

As noted, the 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.

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. Here we provide further details of various components of IdeaL, including their philosophy and operation.

12.5.1 Intelligent Layout Generator

We present a Genetic Algorithms (GA) based approach for building an Intelligent Layout Generator (ILG) by employing various layout design heuristics, including some new, fast, and efficacious ones. 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.

The primary task involved in automating the LD process is to produce superior layout alternatives for further consideration and treatment by decision makers (Akoumianakis et al. 2000, Tompkins et al. 2002). In this regard, past studies have demonstrated that Genetic Algorithms provide a promising search and optimization approach (Abdinnour-Helm and Hadley 2000, Ahmad et al. 2006, Youssef et al. 2003). Our system incorporates experts' knowledge and user preferences in the LD process through composite fitness functions of the ILG. This fitness function utilizes crisp preference weights furnished by the Preference Inferencing Agent.

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 research 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 employed in this research. It 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.

Most of the existing research applies GA in solving layout problem involving 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 involving oriented modules with fixed dimensions, which are 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 and 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 here.

12.5.2 Fitness Evaluation Metrics

As already noted, the LD problem involves such a plethora of subjective and uncertain considerations that no single objective could solely be used to generate layout alternatives. However, automated LD systems require some fitness quantification and evaluation mechanism in order to guide the search to superior solutions. We, therefore, propose the use of some hybrid fuzzy-genetic fitness function that would combine multiple objectives 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. As a preliminary research model, we envisaged the following major categories of design preferences as determinants of layout fitness: Intrinsic Utility of a module, Inter-Module Interaction, Space Utilization, and Qualitative Fitness or Aesthetic Appeal. Intrinsic utility of a module is the utility a module brings when it is included in a layout design. For simplicity sake and without any loss of generalization, we ignore inherent utility of a module in our discussions.

We consider inter-module interaction as an important determinant of layout fitness. IdeaL has been equipped with functionalities for modifying these inter-module interactions in an *interaction matrix*, containing 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. We calculate it as the sum of mutual distances between geometric centers of all pairs of modules or the total Inter-Module Distances (*IMD*).

The space utilization is among the more popular layout design fitness metrics and the literature proposes the Contiguous Remainder (*CR*) or the 'reusable trim loss' as a more appropriate measure of space utilization (Jakobs 1996, Liu and Teng 1999). The *CR* refers to the largest contiguous vacant portion of the packing space available for further placements (Ahmad 2005, Jakobs 1996, Liu and Teng 1999). In other words, *CR* is the empty area on a bin outside the edges of the boundaries created by the packed modules in a layout, as shown in Fig. 12.4. Conceivably, a larger value of *CR* implies that more space is available for further placements.

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

$$CR = Page\ Area - Total\ Module\ Area - Trapped\ Dead\ Space \quad (12.1)$$

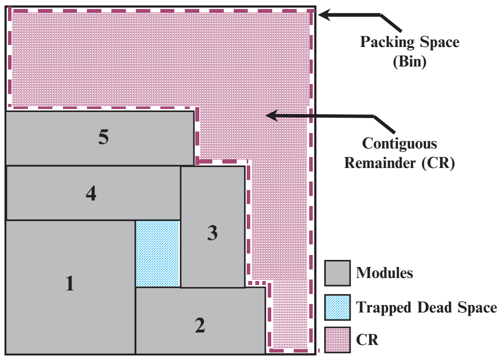

Fig. 12.4. Elaboration of the concept of Contiguous Remainder

If H and W are the height and width of the packing space and h_i and w_i are the height and width of an module M_i, then:

$$CR = H \times W - \sum_{i=1}^{n} w_i h_i - \mathit{Trapped\ Dead\ Space} \tag{12.2}$$

A dual of *CR* is the White Space Level (*WSL*), which is a normalized function and suits the GA and MCDM paradigm more than the *CR* and calculated as follows:

$$C\widehat{R} = WSL = \frac{CR}{\sum_{i=1}^{n} w_i h_i} \times 100 \tag{12.3}$$

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. 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.

Aesthetic values are subjective measures of layout quality. Such values cannot easily be defined in specific terms and usually depend on users' personal judgments. Different people may rate the perceived aesthetic appeal of a given layout differently. 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 [10]. 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.

12.5.3 Genetic Algorithms based Optimization

The Genetic Algorithms (GA) based approach for solving the layout design problem requires determining several critical features including an adequate encoding scheme, an adequate population size, an adequate set of genetic operators, an adequate fitness function, an efficient module placement strategy, and adequate stopping criteria. It should be noted that final set of evolution operators (selection, crossover, mutation, and replacement) and parameters (population size, crossover rates, mutation rates, and termination criteria) would be determined after extensive experimentations with the GA. Nevertheless, it has been argued that the effectiveness of GA methodology is largely insensitive to the exact values of these parameters (Tate and Smith 1995).

The GA encoding scheme for the layout design is a sequence of modules similar to the one adopted by Tate and Smith (Tate and Smith 1995). The sequence S of the module indices (or names). For example: *Sequence of Modules* $= S = \{12,\ 4,\ 9,\ 25,\ 11,\ 47,\ 2,\ 8,\ 16,\ 13,\ 31,\ 45,\ 29,\ 19,\ 33,\ 5,\ 19,\ 7,\ 34,\ 50\}$. This example shows how a sequence of 20 modules, out of a set of 50, to be placed in a given bin. The total length l of the sequence S could be specified either by the expert or possibly be determined by using the maximum number of modules that could be placed on a single bin, amount of white space desired, etc.

We used a pre-specified and static population size P in each generation in evolution process. The initialization step in the GA randomly generates P sequences of modules $(S_1,\ S_2, \ldots S_P)$. Previous studies have shown that a population size of 10–20 provides superior results (Tate and Smith 1995, Jakobs 1996).

In GA, genetic evolution of population creates new layout solutions through genetic operators (crossover and mutation of individual layouts from previous generation). The means of performing these operations must be defined for the layout design problem. A variety of genetic operators could be suggested for the GA. However, we limit ourselves to genetic operators used by Tate and Smith (1995) and Jakobs (1996) for solving layout design problems. These constitute the most popular although only a small extract of possible operators.

The selection operator selects individual layout solutions for genetic operations. We used the rank based selection strategy commonly known as Roulette Wheel selection, one of the most commonly used selection strategies, which is biased towards selecting the fitter solutions for further evolution (Negnevitsky 2002).

In mutation, mutating a single solution generates new individuals. In the context of layout design problem, mutation results in small changes in an existing layout. The mutation rate is selected to be high (around 50%). The reason is that any given chromosome contains only a small subset of the given modules and high mutation rate would ensure that higher chances of incorporating all or most of the modules in test solutions. Furthermore, higher cost of placement algorithms pronounces the need of using 'incremental' GA. Consequently, a higher mutation rate ensures diversity in the population of layouts (Ahmad 2005, Jakobs 1996). The following mutation operators are used in the ILG:

1. Tate and Smith (1995) proposed following set of mutation operator: Reverse the subsequence of the sequence in the mutating layout solution (random selection of the mutating solution and mutating subsequence).
2. Jakobs 1996 used the following set of mutation operator: Exchange elements of two randomly selected layout subsequences.
3. Replace a randomly selected module with a randomly selected module.

During crossover, one or more offspring layouts are derived from two or more parent layouts. In the context of layout design problem, crossover results in combining parts of two existing layouts in order to generate a new layout. The following crossover operators will be used on two parents (say S_j and S_k) selected randomly based on their ranks in the population. Previous studies have demonstrated the success of these operators (Tate and Smith 1995, Jakobs 1996).

1. Tate and Smith (1995) Crossover consists of following steps:
   a. 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).
   b. Insert leftover genes in order (or in random order) to fill in the blanks (unresolved conflicts).
2. Jakobs (1996) Crossover consists of following steps:
   a. Copy q elements of the sequence S_j at a random position p in the new sequence S_{new}. It should be noted that $1 \leq p,\ q \leq n$.
   b. Fill up the remaining elements of S_{new} with other elements of S_k.
3. Append a Randomly selected subsequence from one parent to another.

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, regardless of the fitness of the offspring, on an ongoing basis. As a result, 'superior' offspring could influence the layout solution quality. However, such strategy pronounces the need for high mutation rate to ensure population diversity. An approach similar to his one has proved to be effective and superior for the layout design problem in (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.

The most taxing and application specific task in any particular problem domain exploiting GA is definition of the fitness function. The fitness function is used to differentiate between a 'good' and a 'poor' layout solution. A fitness function should be a well-thought function, as the GA will converge on layout solutions deemed 'fit' by this fitness function. As discussed, a layout design problem involves such a plethora of considerations that no single objective could solely be used to generate alternate layouts. We, therefore, propose a genetic fitness function that combines multiple objectives in terms of rewards/penalties arising from various layout design considerations. The various determinants of layout utility are combined through some crisp weights or preference parameters.

We terminated the GA when the improvement in the fitness of new population over the preceding population is less than a certain value (say *0.1%* or so) or after a certain number of Generations. However, the user would finalize this criterion after performing some focused experimentations with GA.

12.5.4 Proposed Decoding Heuristic

As discussed in Sect. 12.3, existing decoding algorithms lack the requisite efficiency and efficacy. Such shortcomings are more pronounced when layout evaluation criteria include such aesthetic values. In this section, we outline a new, efficient, efficacious, and robust placement algorithms developed for constructing the actual layouts with higher aesthetic contents [3]. The placement algorithm works with an ordering of modules obtained through some non-deterministic and evolutionary metaheuristic-based approach, which is GA in case of IdeaL. The new module placement algorithm is 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. 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 the proposed algorithm is more efficient in the metaheuristic-based layout optimization than other existing heuristics.

We call the proposed placement algorithm as Minimization of Enclosing Rectangle Area (MERA). The name is inspired by the underlying notion where a reduction of the rectangular area of the packing pattern, called Area of Enclosing Rectangle or *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. It should be noted that IdeaL also contains several intelligent adaptations of MERA to provide greater flexibility and power to the user.

The algorithm (Ahmad et al. 2006) proceeds by investigating the placement prospects for all four corners of an in-coming module at all four corners of all in-place modules seeking to find the minimum value of the composite objective function that includes a bias in favor of placement at the bottom-left position in the layout, which is a general packing preference in various placement heuristics or LD contexts such as bin-packing.

In 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 $O(n^\wedge 2)$ cost as for BL and IBL (Jakobs 1996, Liu and Teng 1999). Moreover, some increase in the computational complexity is considered quite rational if significant improvements in terms of both quantitative and qualitative fitness metrics are realized, as demonstrated by the comparative analyses.

12.5.5 Comparative Evaluation of Decoding Heuristics

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 literature for our studies.

A computer program was written in Visual BASIC to implement the BL, IBL, BLF, MERA, and the GA based optimization component including various fitness evaluation functions. The computer program is used for comparative analyses on Intel Xeon 3.06 GHz processor with 256 MB RAM under Windows XP.

Apart from quantitative analyses based on contiguous remainder and inter-module distances, 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 for the given problem.

We used several benchmark problems from the literature for our comparative studies. 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, and MERA placement strategies for 100 random sequences of each benchmark problem instance is discussed in (Ahmad et al. 2006). Results have shown that MERA outperforms the existing algorithms by wide margins. The proposed algorithm generate superior outcomes in terms of 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.

We also employed GA based metaheuristic search in our comparison. The average of ten GA runs for the 100-module problem with a population size of 50, a mutation rate of 0.8, and a crossover rate of 0.2 is shown in Table 12.1. It can be seen that MERA outperforms the existing algorithms by wide margins.

Table 12.1. Comparison of Decoding Heuristics with GA search

Objective	Tech.	Best Fitness (% difference from optimal)
CR	**BL** + GA	3432 (−31.4%)
(Optimal = 5,000)	**IBL** + GA	3905 (−21.9%)
The Higher the Better	**BLF** + GA	4235 (−11.3%)
	MERA + GA	**4709** (−**5.8**%)
IMD	**BL** + GA	553459.5 (+1.7%)
(Reference = 536,000)	**IBL** + GA	521419.6 (+7.4%)
The Lower the Better	**BLF** + GA	483010.3 (+14.2%)
	MERA + GA	**450759.9** (+**19.9**%)
QR	**BL** + GA	1.5
(Ideal = 10)	**IBL** + GA	1.75
The Higher the Better	**BLF** + GA	3.5
	MERA + GA	**5.25**

12.5.6 Fuzzy Preference Inferencing Agent

Here we provide details about modeling of, and inferencing from, subjective and uncertain preferences as well as the design, implementation, and working of the Preference Inferencing Agent.

The brain of any ES is an Inference Engine that contains general algorithms capable of manipulating, and reasoning about, the knowledge stored in the knowledge base for solving problems by devising conclusions (Turban and Aronson 2001). 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 research, 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 LD 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. However, little effort has been done in formalizing such an application of fuzzy logic in LD systems. Nevertheless, an FL based Preference Inferencing Agent seems to be an important component in any LD decision aid tool (Ahmad 2002, 2005, Karray et al. 2000, Raoot and Rakshit 1993).

As such, the underlying concept in IdeaL's inferencing uses 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). Consequently, an FL based PIA is expected to deliver much of the flexibility in the automated LD process that the LD practitioners have always longed for. As such, we deem PIA as one of the core components, along with ILG, in tackling and automating the LD 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 Sect. 12.5.1.

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.

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 reduction in utility. 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 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 $\psi > 1$. 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. $\psi = 2$), as follows:

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

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. In addition, the question of developing more effective and robust layout fitness metrics remains open for further research in MCDM field.

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, a common scenario in practice. We consider the same 100-module problem used in Sect. 12.5.1, but the reduced size of the packing space 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 be placed in a single bin or packing space. This important parameter determines 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, determining the search space, dramatically affecting the efficiency and quality of results. IT is because employing a chromosome size of 100 would result in unnecessary search and 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 Fig. 12.7. The complete set of fuzzy rules for determining *choromosome_size* using *white_space* and *bin_size* is provided in Table 12.2. Our example consists of a simple two-input and one-output scenario with the following two fuzzy rules specified by an expert:

We used the Mamdani-style inference method, as it is the most popular technique for capturing experts' knowledge, (Negnevitsky 2002) Using this technique, the crisp value for the chromosome size came out to be 27 (Ahmad 2005).

In order to evaluate the effect of the *chromosome_size* as determined by the PIA, we ran 1,000 iterations of the GA with chromosome sizes of 27 and 100 employing MERA as the decoding heuristic. The average time per GA iteration with a chromosome size of 100 was 15.43 s. In contrast, the average

Table 12.2. Fuzzy rules for determining the chromosome size

Rule 1:	**Rule 2:**
If x is *A2* (*white_space* is *medium*)	If x is *A3* (*white_space* is *large*)
Or y is *B3* (*bin_size* is *large*)	Or y is *B4* (*bin_size* is *ex-large*)
Then z is *C2* (*chromosome_size* is *medium*)	Then z is *C3* (*chromosome_size* is *large*)

		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*

time per GA iteration with a chromosome size of 27 was only 0.316 s. 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 LD, providing greater flexibility, expressive power, and ability to model vague preferences.

12.5.7 Preference Discovery and Validation Agent

The reliability and effectiveness of PIA significantly depends on the reliability of preferences. The task of extracting knowledge from experts is extremely tedious, expensive, and time consuming. 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). However, a traditional ES cannot automatically learn preferences or improve through experience. Here we describe a small-scale Preference Discovery Agent (PDA) for testing the idea of automated preference discovery and revision in LD.

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. 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. 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 fully functional 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 IdealL. Nevertheless, here we provide 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.

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 reported in (Ahmad 2005). In our PDA, we used a fully connected artificial neural network with one hidden layer. 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), the later one is a subjective measure of fitness and details of which can be found in (Ahmad 2005, Mak et al. 1998). 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 (Ahmad 2005, Negnevitsky 2002). The decision on the architecture of an ANN is typically done through a trial-and-error. We found a hidden layer with three neurons sufficient for our purposes.

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.

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.

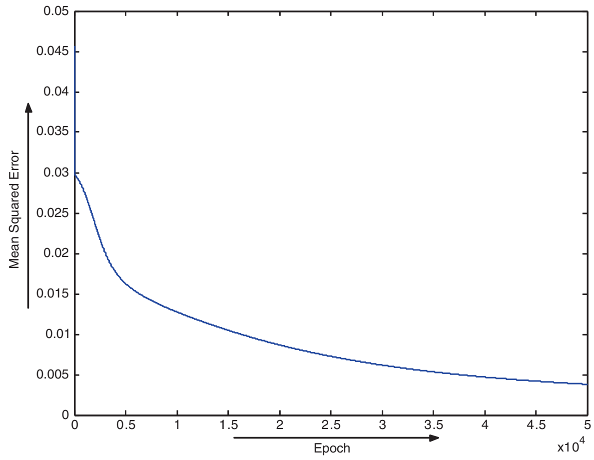

Fig. 12.5. Convergence of the training phase of the PDA

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 Fig. 12.5, demonstrating a sound convergence capability of the PDA. For comparison purposes, the Pattern Error, or the difference between the target value and the actual output for the training set of PDA, was less than 4%, indicating the capability of PDA to learn and generalize from the given training instances.

12.5.8 Knowledge Base

Knowledge is the primary raw material in an ES (Walenstein 2002). The conceptual model of the elicited knowledge is converted to a format suitable for computer manipulation through a process called the Knowledge Representation (Negnevitsky 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 as knowledge may 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. Representing knowledge in the form of such traditional production rules enhances the modularity of the system and prompted us to adopt this approach. However, 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 LD systems. This furnished another reason for our choice of fuzzy logic modeling preferences and building the inference engine for IDEAL.

12.5.9 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 and 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 (Negnevitsky 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 derived from domain experts, the existing knowledge, as well as through some automated 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 research.

12.5.10 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 and 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 piece of information is needed or how conclusions are obtained.

Explanation Facilities are vital from both system development and marketing perspectives. These facilitate both debugging of the knowledge base as well as user acceptance and adoption. Such 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 research. 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.

12.5.11 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" (Healy et al. 1999). Furthermore, research has shown that interface aesthetics as well as interactivity perform a larger role in users' attitudes than users would admit (Ngo et al. 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.and Law 2003, Schnecke and Vonberger 1997). 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 Figs. 12.6 and 12.7 for reference purposes. Details regarding the UI can be found in (Ahmad 2005).

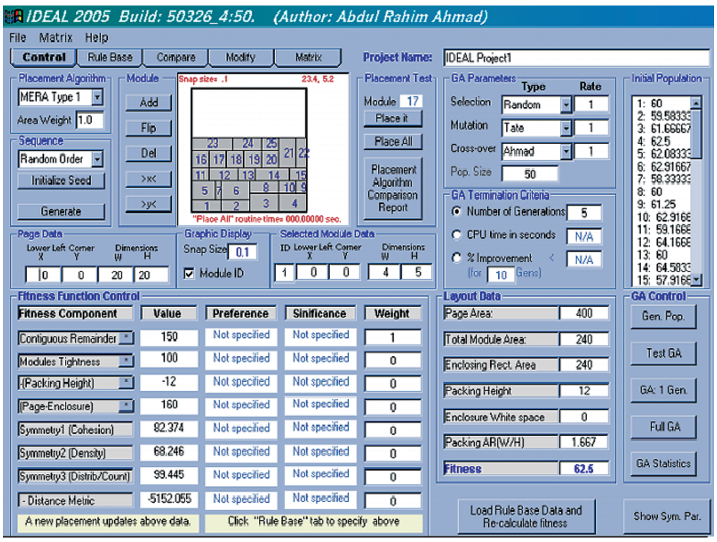

Fig. 12.6. User interface for developers (Normal view)

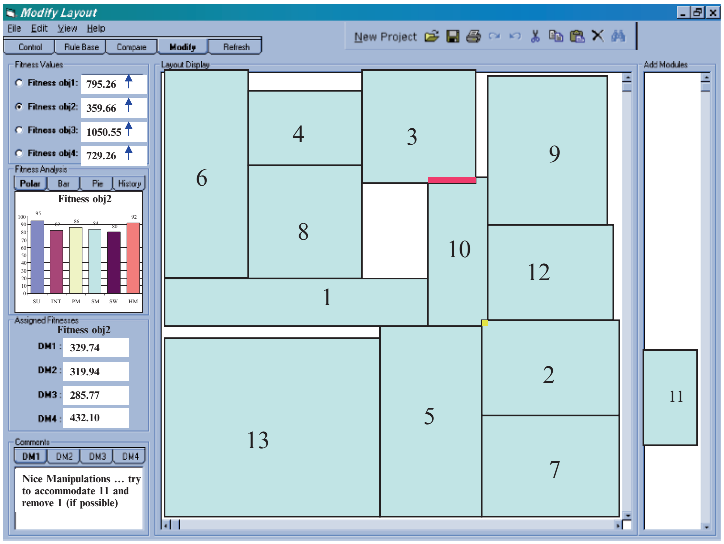

Fig. 12.7. User interface for layout designers

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 and Hajdukiewicz 2004). However, a prototype of an end-user interface has been developed, and tested, using the philosophy of Ecological Interface Design and various usability and Human-Computer Interaction guidelines (Ahmad 2004). We employed a combination of digital and analog displays for increasing the interface efficacy. Further, 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 would furnish creative ways in which they can support decision-makers' work.

12.5.12 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 2005, 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).

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 Agent* (PDA) complements the ILG and the PIA by automatically discovering and refining some preferences. The proposed synergy is shown in Fig. 12.8.

In this synergy, 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 through its inferencing mechanism, furnishing crisp weights for use in the ILG. The ILG, in turn, generates superior layout alternatives for ranking and manipulation by decision-makers. The layout alternatives generated by the ILG could be

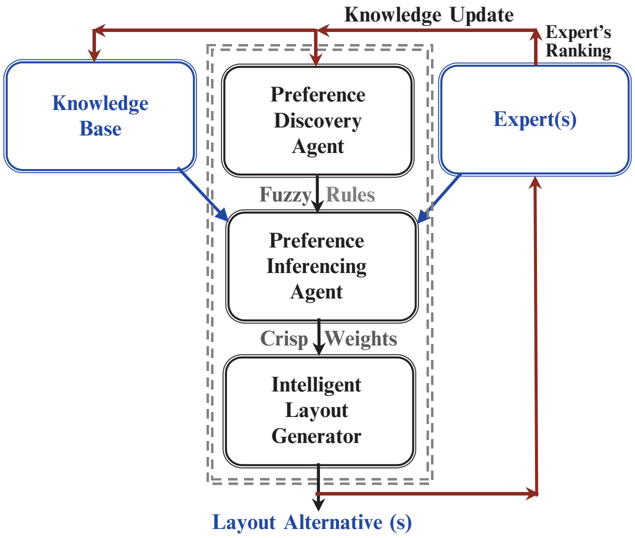

Fig. 12.8. The Synergy of the Intelligent Components in IDEAL

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). 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 2005, Cordon et al. 2004, Negnevitsky 2002).

We want to emphasize that these components have deliberately been designed to 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 various layout design problems.

12.6 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 (Ahmad 2004).

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 Sect. 12.5.7. These alternatives were given 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 Sect. 12.5.7. Few interesting instances of this exercise are presented here to demonstrate the efficacy of IdeaL.

Case I. The layout alternative presented in Fig. 12.9 was generated by IdeaL and received a rating a rating of 70 out of 100 from the expert. Apparently, the layout shown in Fig. 12.9 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 Fig. 12.10 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 module-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 mins. to complete and naturally followed each other.

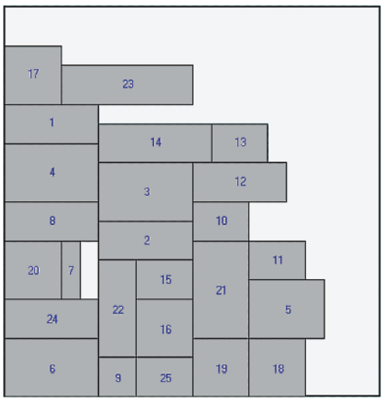

Fig. 12.9. Case I – layout alternative

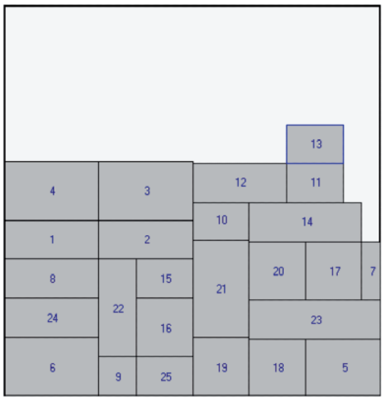

Fig. 12.10. Case I – refined layout

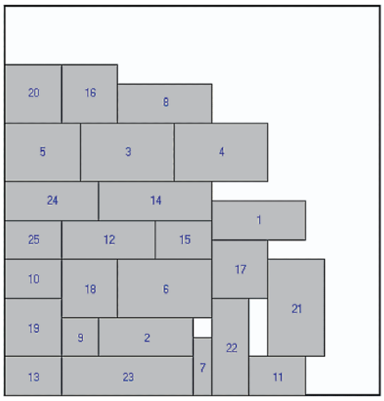

Fig. 12.11. Case II – layout alternative

The resultant layout subsequently received a subjective a rating of 90 out of 100 by the DM.

Case II. The layout alternative presented in Fig. 12.11 was generated by IDEAL and 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 Fig. 12.12 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. When this resultant pattern was given to experts, it received a subjective rating of 85 out of 100.

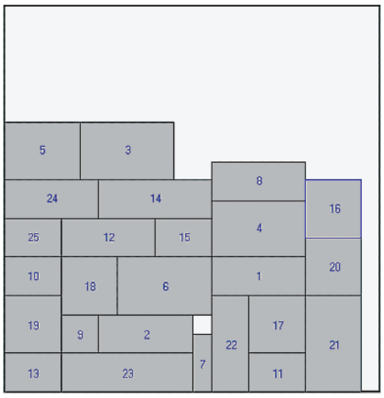

Fig. 12.12. Case II – refined layout

12.7 Future Research

It is hoped that the exclusive and complementary features of various soft computing technologies will result in a synergistic integration, providing new insights to practitioners and theoreticians. Here we list some interesting research directions.

12.7.1 Knowledge Base

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.

12.7.2 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.

12.7.3 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.

12.8 Conclusion

In this chapter, we have described the layout design problem, its significance and relevance, and the role intelligent systems and soft computing tools can play in improving the efficacy and efficiency of layout design process. In particular, we have explained the development and working of a novel intelligent approach to solving this important and intricate problem. Our approach involves the use of human intuition, heuristics, metaheuristics, and soft computing tools like artificial neural networks, fuzzy logic, and expert systems. We have explained the philosophy and synergy of the various intelligent components of the system. This research framework and prototype contribute to the field of intelligent decision making in layout design and analysis by enabling explicit representation of experts' knowledge, formal modeling of fuzzy user preferences, as well as swift generation and effective manipulation of superior layout alternatives. Such efforts are expected to afford efficient procurement of superior outcomes and to facilitate the cognitive, ergonomic, and economic efficiency of layout designers as well as future research in related areas.

References

Abdinnour-Helm, S., Hadley, S.W., "Tabu search based heuristics for multi-floor facility layout", *International Journal of Production Research*, Vol. 38, No. 2, pp. 365–383, 2000.

Adya, S.N., Markov, I.L., Villarrubia, P.G., "Improving Min–cut Placement for VLSI Using Analytical Techniques", *Proc. IBM ACAS Conf.*, IBM ARL, Feb. 2003, pp. 55–62.

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, Uni. of Waterloo, April 2002.

Ahmad, A.R., "An Effective User Interface for Knowledge-based Decision Support in Layout Design", *Working Paper*, Systems Design Engineering, Uni. of Waterloo, 2004b.

Ahmad, A.R., An Intelligent Expert System for Decision Analysis and Support in Multi-Attribute Layout Optimization. Ph.D. Thesis, University of Waterloo, Canada, 2005. http://etd.uwaterloo.ca/etd/arahim2005.pdf

Ahmad, A.R., Basir, O.A., Hassanein, K., "Decision Preferences, Constraints, and Evaluation Objectives in Layout Design: A Review of Modeling Techniques", *5th Int'l Conf. on Operations and Quantitative Management (ICOQM-V)*, Oct. 2004, Seoul, 2004a.

Ahmad, A.R., Basir, O.A., Imam, M.H., Hassanein, K., "An effective module placement strategy for genetic algorithms based layout design", *International Journal of Production Research*, Vol. 44, No. 8, pp. 1545–1567, 2006

Akin, O., Dave, B., Pithavadian, S., "Heuristic Generation of Layouts (HeGeL): based on a paradigm for problem structuring", *Environmental Planning*, Vol. 19, No. 2, 225–245, 1997.

Akoumianakis, D., Savidis, A., Stephanidis, C., "Encapsulating intelligent interactive behaviour in unified user interface artifacts", *Interacting with Computers*, Vol. 12, pp. 383–408, 2000.

Azadivar, F., Tompkins, G., "Simulation Optimization with Quantitative Variables and Structural Model Changes: A Genetic Algorithm Approach", *European Journal of Operational Research*, Vol. 113, pp. 169–182, 1999.

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

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

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

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.

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

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

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

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

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.

Deb, S.K., Bhattacharyya, B., "Fuzzy decision support system for manufacturing facilities planning", *Decision Support Systems*, 2004.

Dowsland, K.A., S. Vaid, W.B. Dowsland, "An algorithm for polygon placement using a bottom-left strategy", *European Journal of Operational Research*, Vol. 141, pp. 371–381, 2002.

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

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

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

Foulds, L.R., "*Graph Theory Applications*", Springer Berlin Heidelberg New York, 1995.

Garey, M., Johnson, D.,"*Computers and Intractability*", Freeman, NY, 1979.

Gloria, A.D., Faraboschi, P., Olovieri, M., "Block placement with a Boltzman machine", *IEEE Transctions CAD*, Vol. 13, No. 6, pp. 694–701, 1994.

Hassan, M.M.D., Hogg, G.L., "On constructing the block layout by graph theory", *International Journal of Production Research*, vol. 32, No. 11, pp. 2559–2584, 1994.

Head, M., Hassanein, K. "Web Site Usability for eBusiness Success: Dimensions, Guidelines and Evaluation Methods", *Proc. of the Hawaii Int'l Conf. on Business*, 2002.

Healy, P., Creavin, M., Kuusik, A., "An optimal algorithm for rectangle placement", *Operations Research Letters*, Vol. 24, pp. 73–80, 1999.

Heragu, S.S., A. Kusiak, "Machine layout: an optimization and knowledge-based approach", *International Journal of Production Research*, Vol. 28, No. 4, pp. 615–635, 1990.

Hopper, E., Turton, B., "An Empirical Investigation of meta-heuristic and heuristic algorithms for a 2D packing problem", *European Journal of Operational Research*. Vol. 128, pp. 34–57, 2001.

Hower, W., "Placing Computations by Adaptive Procedures", *Artificial Intelligence in Engineering*, Vol. 11, pp. 307–317, 1997.

Irani, S., Huang, H., "Custom Design of Facility Layouts for Multi-Product Facilities Layout Modules", *IEEE Transactions on Robotics and Automation*, Vol. 16, pp. 259–267, 2000.

Islier, A.A., "A genetic algorithm approach for multiple criteria facility layout design", *International Journal of Production Research*, Vol. 36, No. 6, pp. 1549–69, 1998.

Jackson, P., "*Introduction to Expert Systems*", Addison Wesley, England, 1999.

Jakobs, S., "On genetic algorithms for packing of polygons", *European Journal of Operational Research*, Vol. 88, pp. 165–181, 1996.

Karray, F., De Silva, C., "*Soft Computing and Intelligent Systems Design: Theory, Tools and Applications*", Readings, MA: Addison-Wesley, 2004.

Karray, F., E. Zaneldin, T. Hegazy, A.H.M. Shabeeb, E. Elbelgati, "Computational Intelligence Tools for Solving Facilities Layout Planning Problem", *Proc. of the American Control Conference, Chicago*, Vol. 8(4), pp. 367–379, June 2000a.

Kim, J.K., Kwang, H.L.,Yoo, S.W., "Fuzzy bin packing problem", *Fuzzy Sets and Systems*, Vol. 120, pp. 429–434, 2001.

Kumara, S.R.T., Kashyap, R.L., 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.

Lee, Y.H., Lee, M.H., "A shape-based block layout approach to facility layout problem using hybrid genetic algorithm", *Computers and Ind. Engineering*, Vol. 42, pp. 237–248, 2002.

Leung, T.W., Chan, C.K., Troutt, M.D., "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.

Ligget, R.S., "Automated facilities layout: past present and future", *Automation in Construction*, Vol. 9, 197–215, 2000.

Liu, D., Teng, H., "An Improved BL-Algorithm for Genetic Algorithm of the Orthogonal Packing of Rectangles", *European Journal of Operational. Research*, Vol. 112, pp. 413–420, 1999.

Lodi, A., Martello, S., Vigo, D., "Approximation algorithms for the oriented two-dimensional bin packing problem", *European Journal of Operational. Research*, Vol. 112, pp. 158–166, 1999.

Lodi, A., Martello, S., Monaci, M., "Two-dimensional packing problems: A survey", *European Journal of Operational Research*, Vol. 141, pp. 241–252, 2002.

Love, R.F., Wong, J.Y., "Solving quadratic assignment problems with rectilinear distance and integer programming", *Naval Res. Log. Quarterly*, 23(4), 623–627, 1976.

Mak, K.L., Wong Y.S., Chan, F.T.S., "A genetic algorithm for facility layout problems", *Computer Integrated Manufacturing Systems*, Vol. 11, No. 1–2, pp. 113–127, 1998.

Malakooti, B., Tsurushima, A., "An expert system using priorities for solving multiple-criteria facility layout problems", *International Journal of Production Research*, Vol. 27, No. 5, pp. 793–808, 1989.

Martens, J., "Two genetic algorithms to solve a layout problem in fashion industry", *European Journal of Operational Research*, Vol. 154, pp. 304–322, 2004.

Mazumder, P., Rudnick, E.M., "*Genetic Algorithms for VLSI Design, Layout and Test Automation*", New York, NY: Prentice-Hall 1999.

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

Mir, M., Imam, M.H., "Topology optimization of arbitrary-size blocks using bivariate formulation", *Journal of Computer Aided Design*, Vol. 24, No. 10, pp. 556–564, 1992.

Mir, M., Imam, M.H., "Analytic annealing for macrocell placement problem", *International Journal of Computer and Electrical Engineering*, Vol. 22, No. 2, 1996.

Mir, M., M.H. Imam, "A hybrid optimization approach for layout design of unequal-area facilities", *Computers and Industrial Engineering*, Vol. 39, No. 1/2, pp. 49–63, 2001.

Moon, B.R., Kim, C.K., "Dynamic embedding for genetic VLSI circuit partitioning", *Engineering Applications of Artificial Intelligence*, Vol. 11, pp. 67–76, 1998.

Negnevitsky, M., *Artificial Intelligence: A Guide to Intelligent Systems*", Pearson, 2002.

Ngo, D., "Measuring aesthetic elements of screen designs" *Displays*, 22, pp. 73–78, 2001.

Ngo, D.C.L., Law, B.L., "An expert screen design and evaluation assistant that uses knowledge-based backtracking" *Information & Software Technology*, Vol. 43, 293, 2003.

Norman, B.A., Smith, A.E., "Considering Uncertainty in Unequal Area Block Layout Design", citeseer.ist.psu.edu/271501.html, 2002.

Osman H., Georgy, M., Ibrahim, M., "A hybrid CAD-based construction site layout planning using genetic algorithms", *Automation in Construction*, Vol. 12, pp. 749–764, 2003.

Pierce, J.F., Crowston, W.B., "Tree search algorithms in quadratic assignment problems", *Naval Research Logistics Quarterly*, Vol. 18, pp. 1–36, 1971.

Raoot, A.D., A. Rakshit, "A 'linguistic pattern' approach for multiple criteria facilities layout problems", *International Journal of Production Research*, Vol. 31, pp. 203–222, 1993.

Schnecke, V., Vonberger, O., "Hybrid Genetic Algorithms for Constrained Placement Problems", *IEEE Trans. on Evolutionary Computation*, Vol. 1, No. 4, 266–277, 1997.

Tam, C., Tong, T., Leung, A., Chiu, G., "Site Layout Planning using Nonstructural Fuzzy Decision", *J. of Construction Engg and Management*, May 2002, pp. 220–231, 2002.

Tate, D.M., Smith, A.E., Unequal-area facility layout by genetic search. IIE Transactions 1995, 27: 465–472.

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*. Verlag, London, 1997.

Tompkins, J.A., White, J.A., Bozer, Y.A., Tanchoco, J.M.A., "*Facilities Planning*", 3rd Ed., Reading: Wiley, NY, 2002.

Turban, E., Aronson, J.E., "*Decision Support Systems and Intelligent Systems*", 6th ed., Reading: Prentice Hall, Upper Saddle River, NJ, 2001.

Walenstein, A., *"Cognitive Support in Software Engineering Tools: A Distributed Cognition Framework"*, Ph.D. Thesis, Computing Science, Simon Fraser University, 2002.

Welgama, P.S., Palitha, S., Gibson, P.R., "Computer-Aided Facility Layout – A Status Report", *International Journal of Advanced Manufacturing Technology*, Vol. 10, No. 1, pp. 66–77, 1995.

Wu, Y., Huang, W., Lau, S., Wong, C., Young, G., "An effective quasi-human based heuristic for solving the rectangle packing problem", *European Journal of Operational Research*, Vol. 141, pp. 341–358, 2002.

Youssef, H., Sait, S.M., Ali, H., "Fuzzy Simulated Evolution Algorithm for VLSI Placement", *International J. on Applied Intelligence, Sp. issue on Applied Metaheuristics*, 2003a.

Zadeh, L.A., "Fuzzy Sets as a Basis for a Theory of Possibility", *Fuzzy Sets and Systems*, Vol. 100 (Supplement), pp. 9–34, 1999.