Songklanakarin J. Sci. Technol.
30 (5), 673-686, Sep. - Oct. 2008

Songklanakarin Journal of Science and Technology
http://www.sjst.psu.ac.th

Original Article

Mixed integer programming model with non-circular and guided constraints for architectural layout design optimization

Kamol Keatruangkamala* and Krung Sinapiromsaran

Department of Mathematics, Faculty of Science,
Chulalongkorn University, Phayathai Road, Pathum Wan, Bangkok, 10330 Thailand.

Received 14 December 2007; Accepted 12 September 2008

Abstract

Various techniques have been used to solve a challenging architectural layout design problem for more than a decade, such as an expert system, an evolutionary algorithm, a simulated annealing and a mathematical programming method. This paper concentrates on the mathematical programming technique that formulates an architectural layout design optimization as the mixed integer programming model using the state-of-the art optimization solver to determine the optimal solution. All non-linear relationships among design components are captured using the corresponding linear equalities and linear inequalities. Due to the combinatorial nature of the MIP solutions, the MIP can be solved for small problem sizes, 2-6 rooms, within a reasonable time limit. To remedy this situation, the valid inequality of non-circular connections has been adopted that reduces the computational time significantly. Moreover, the guided constraints based on the architect's preferences of a specific room have been embraced. This helps abandon some alternative solutions and reduces the search space considerably. The computational time and iterations gain of more than 80% is now achievable for the architectural layout design for 7-10 rooms.

Keywords: architectural layout design, optimization, mixed integer programming, valid inequalities, automated design tool

1. Introduction

Architectural layout design is an initial phase of a design process during which architect takes the specification of spatial objects and generates numerous feasible drafts. It is the most critical phase which influences the final designed decision. This architectural layout design can be interpreted as solving a combinatorial problem. Solution methodologies for architectural layout design present the most comprehensive challenges in the area of architectural design computation. This problem is known to be NP-hard (Michalek *et al.*, 2002; Russell *et al.*, 1999), see Figure 1.

Structural representation (Bloch *et al.*, 1978; Gero, 1990; Honda *et al.*,1995; Schwarz *et al.*, 1994) of a spatial requirement is needed to form the basic component of a physical design problem to be automatically solved by a computer. One representation used a grid system (Homayouni, 2006), see Figure 2(a). This representation is an in-

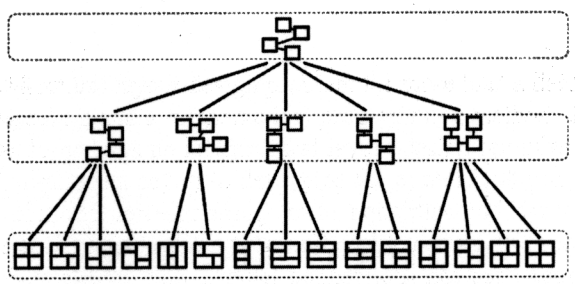


Figure 1. Conceptually, there is a large number of possible placements adjusted in a variety of ways.

*Corresponding author.
Email address: kamolkeat@hotmail.com

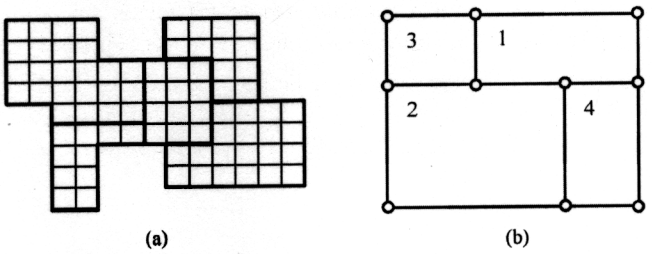


Figure 2. (a) Grid system and (b) dissections of rectangular component based on wall-representation.

herently discrete and multi-modal.

Due to the combinatorial configurations, the problem cannot be solved exhaustively for reasonable-sized layout. Liggett and Mitchell (1981) used a constructive placement strategy on the grid system where a room space is allocated one at a time. Then the iterative improvement based on the objective function has been applied to improve the current solution.

Another structural representation is the wall representation. Flemming (1978) identified the location of walls in the space to partition a layout into rectangular components, see Figure 2(b). This structural representation has an advantage over the grid-based layout by limiting nonrectangular shapes of space patterns which helps with decreasing the computational time.

The primary structural representation used in this paper is based on a mathematical programming (Bloch, 1978; O'Sullivan, 1999; Willoughby, 1970) using a coordinated system. Michalek, Choudhary and Papalambros (2002) constructed an optimization model of the quantifiable aspects of an architectural layout design that determines the best location and size of a group of interrelated rectangular spaces using a middle coordinate (x, y) of each room. This allows an optimization algorithm to alter the position of a room independently to achieve the optimal cost satisfying all architectural design requirements.

This research proposes a mathematical programming technique to solve the layout design problem. We develop the mixed integer programming model (MIP) (Grorge, 1988; Linderoth *et al.*, 1999; Russell *et al.*, 1999) to determine the optimal objective architectural layout design which simultaneously solves the topological and geometrical requirements (Scott *et al.*, 1999). The approach presented here called base AL-MIP guarantee the global optimal solution if the base AL-MIP solver stops normally (Keatruangkamala and Sinapiromsaran, 2005). The advantage of base AL-MIP model is an easy adaptability for other architectural requirements such as the fixed room location, the unused grid cells, the fixed border location and the favorable choice of the nearest room to the top left corner. These formulations allow architects to design a layout beyond the rectangular boundary (Scott *et al.*, 1999). However, the architectural layout design problem is considered to be an ill-defined and over-constrained problem (Simon, 1973) that deals with a large set of possible solutions which cannot be solved exhaustively for reasonably-sized problems. This situation can be resolved using special constraints via the mathematical programming technique.

Our base AL-MIP model is formulated as the multi-objective MIP model based on objective and subjective preferences which can practically be solved for 2-6 rooms. To handle the medium-sized instances, two classes of mathematical inequalities have been developed to decrease the computational time. The first class of mathematical inequalities is based on a valid inequality of non-circular constraints, called non-circular AL-MIP. This utilizes two decision variables p and q from the base AL-MIP, which causes the reduction of the feasible region while maintaining all integral solution points. The second class of mathematical inequalities is based on the North-South-East-West guided constraint called guided AL-MIP. Both non-circular and guided AL-MIP abandon alternative solutions but maintain the final objective value. By applying these two classes of mathematical inequalities, we could immensely reduce computational time from the base AL-MIP.

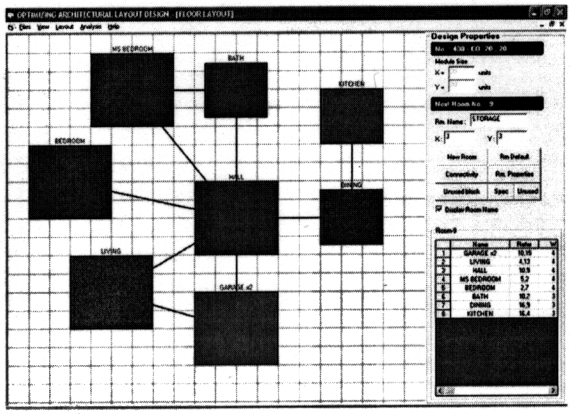
The empirical study in Keatruangkamal and Sinapiromsaran 2006, showed that no binary variables were assigned all ones or all zeros. Therefore, they eliminated those extreme cases using the heuristic cuts such as the sum of all binary variables must be less than a specified threshold. There is a trade-off between the processing speed and the merit of the objective values. It was shown that the solution time could be reduced up to 70% by accepting non-optimal solution based on a specified threshold.

To practically apply AL-MIP model, this research has been developed using the software named ALDO (Architectural Layout Design Optimization) to help an architect identifying the layout requirements graphically (Keatruangkamala and Sinapiromsaran, 2005), (Figure 3). This software utilizes the graphic user interface (GUI) running on the Windows operating system and automatically solves the given architectural layout instance. Furthermore, an architect can request a drawing presentation of the global optimal solution or save it as a DXF format file to use with other CAD softwares.

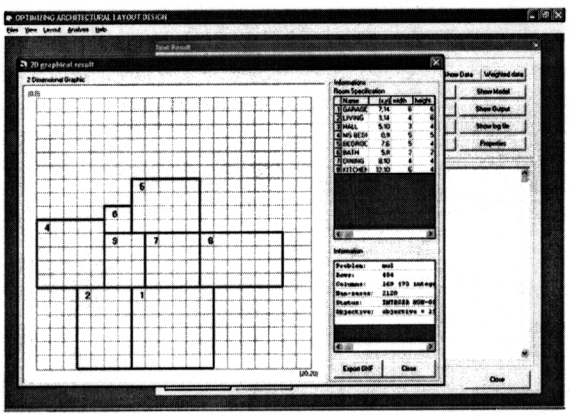
In the next section, we briefly present the base AL-MIP. In section 3, we describe the non-circular AL-MIP and the guided AL-MIP. The valid inequalities and the architect's preference will be presented and explained in details. In section 4, we present experiments and show the computational time improvement. Also, comparison has been resulted with a case study from another paper. Finally, in section 5, we draw the conclusion and suggest future work.

2. Optimizing architectural layout design via mixed integer programming

In this research, we proposed the mixed integer programming to solve an architectural layout design problem. Our AL-MIP formulation proposed here guarantees the



(a)



(b)

Figure 3. The GUI software ALDO, (a) the graphical input of room connectivity and (b) the graphical output.

optimal design based on our objective preferences and architect's requirements, which are formulated as linear relationship.

2.1 Mixed integer programming (MIP)

The integer linear programming problem (George, 1988) is simply a linear program (LP) in which all variables are restricted to integral values. The mixed integer program (MIP) is a linear program which some variables must be integer.

(IP) maximize $c^T x$
subject to: $Ax \leq b$
$x \geq 0$,
x is integer

(MIP) maximize $c^T x + d^T y$
subject to: $Ax + Dy \leq b$
$x \geq 0$,
x is integer
$y \geq 0$

where x and y are vectors of design variables, A and D are matrices of inequality constraints.

2.2 Formulations

2.2.1 Architectural layout design optimization

The architectural layout design problem is posed as a process of finding the best location and size of a group of interrelated rectangular rooms. In this research, we define the room as a rectangular space to represent a specific architectural function such as living spaces, storage spaces, and facilities space. Given a set of rooms {1, 2, ..., n}, Figure 4(a) shows the i^{th} room represented by (x_i, y_i) from the top left corner with its height h_i and width w_i.

Our AL-MIP model is formulated based on the coordinate system using the top left corner of the boundary area as the reference origin (0, 0). The positive value of x corresponds to x units to the right of the origin while the positive value of y corresponds to y units below the origin. According to this AL-MIP model, x, y, w_i and h_i are defined as continuous variables. We avoid the excessive use of integral variables due to the explosive combinatorial number of possible values. Coordinates and dimensions are used as design variables, see Figure 4(a).

x_i = X coordinate of the top left corner of the room i.
y_i = Y coordinate of the top left corner of the room i.
w_i = the horizontal width of the room i.
h_i = the vertical height of the room i.

Two boundary parameters are the layout width and the layout height which are represented by W and H, respectively. Moreover, there are specific parameters for each room,

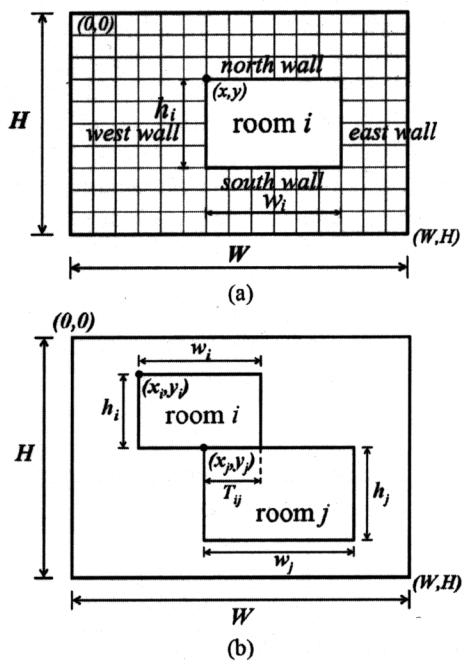


Figure 4. (a) Model variables and parameters based on the coordinate system and (b) model relationships between two connected rooms.

the lower and upper limits of the room width and the room height, $w_{min,i}$, $w_{max,i}$, $h_{min,i}$, $h_{max,i}$, respectively. In addition, T_{ij} is a minimal contact length parameter between room i and j for a passage way or a door, see Figure 4(b).

2.2.2 Multiobjective

In the past, research usually concentrated on a single objective function. Fleming (1978) presented a singular objective layout via the representation and generation of rectangular dissections that minimized room space. In 2000, the work of Li, *et al.* (2000) dealt with maximizing the area in a given floor layout. In contrast, new researches are more interested in multiobjective preferences (Medjdoub *et al.*, 1998; Michalek *et al.*, 2002; Mitchell *et al.*, 1976). In this paper, we are interested in maximizing room sizes and minimizing absolute distance between rooms. To cope with these multiobjective preferences, we combine two objective functions into a summation of weighted components. These weights can be adjusted according to architect's favor. In our experiment, we use equal weights to measure performance of our MIP model.

At the optimum, there always exist alternative solutions with the same objective values due to the layout rotation. In order to eliminate alternative solutions, we select the first room to be placed near the top left corner. For selected i^{th} room,

Minimize
$u_1(x_i + y_i) + u_2\Sigma$ absolute distance $- u_3\Sigma$ maximizing room sizes

where u_1 is the weight of the i^{th} room positioning to the nearest top left corner, u_2 is the weight of the total absolute distance and u_3 is the weight of the total room sizes. If an architect prefers a larger total room size then the weighted sum of u_3 is set to be greater than u_2. If an architect prefers a short total distance among rooms then u_2 is set to be greater than u_3.

1) Placing a room position near the origin.

The combinatorial nature of the alternative optimal solutions having the same objective values could affect the total solution time. To allow the MIP algorithm to prune other alternative solutions, architects will pick the i_o^{th} room position to the nearest origin of the boundary area.

$$\min \quad x_{io} + y_{io} \tag{1}$$

where x_{io} is the X coordinate of the i_o^{th} room,
y_{io} is the Y coordinate of the i_o^{th} room.

2) Minimizing the absolute room distance

One interesting criterion of an architect preference deals with a short distance among rooms. We apply the absolute distance function called Manhattan distance, instead of the normal Euclidean distance, see Figure 5(a). This distance function is preferred over the Euclidean distance function since it maintains the unit scale and a distance measure from the room layout should not be diagonal. The linear formulation of Manhattan distance is computed as the summation of an absolute difference on the X coordinate and Y coordinate, (Figure 5(b)).

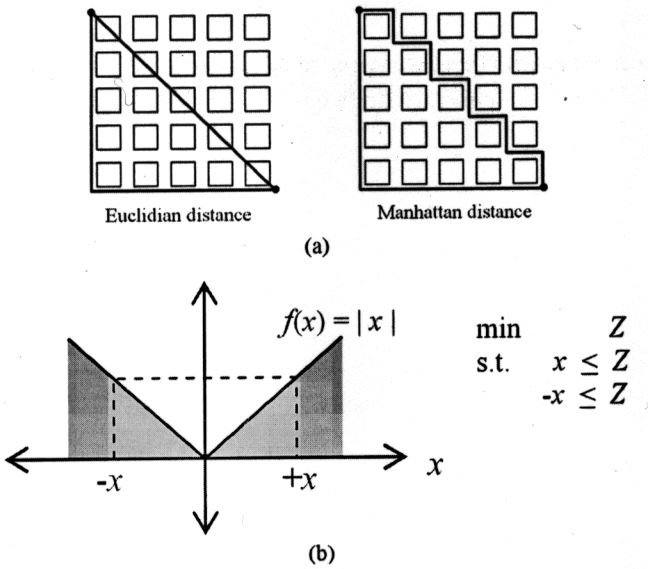


Figure 5. (a) Comparison between Euclidian distance function and Manhattan distance function and (b) an absolute distance function formulated using a linear model.

$$\min \quad z^x_{i,j} + z^y_{i,j} \tag{2}$$

$$\text{subject to:} \quad x_i - x_j \le z^x_{i,j} \tag{3}$$

$$x_j - x_i \le z^x_{i,j} \tag{4}$$

$$y_i - y_j \le z^y_{i,j} \tag{5}$$

$$y_j - y_i \le z^y_{i,j} \tag{6}$$

where $z^x_{i,j}$ and $z^y_{i,j}$ are absolute distances on X coordinate and Y coordinate, respectively, x_i and x_j are the X coordinates of room i and j, y_i and y_j are the Y coordinates of room i and j. Figure 5(b) illustrates the absolute distance and its equivalent linear model.

3) Maximizing room size

Another important architect's preference is the spacious room space. A rectangular area can be computed by multiplying two sides as a non-linear function. However, the MIP model only deals with linear functions and constraints. Therefore, we decide to maximize the room size which has the direct effect to the room area. The larger the room size

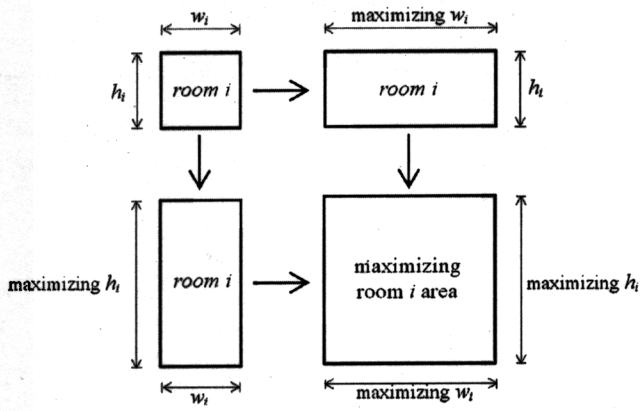


Figure 6. Maximizing room area is constructed by maximizing each room side.

is, the greater the area will be, (Figure 6).

$$\max \quad z_i^a \tag{7}$$

$$\text{subject to:} \quad z_i^a \le w_i \tag{8}$$

$$z_i^a \le h_i \tag{9}$$

2.2.3 Architectural constraints

Architect's requirements have been captured using two principle constraints (Honda *et al.*, 1995; Li *et al.*, 2000), functional constraints and dimensional constraints. Functional constraints determine the placement location of rooms according to the architectural requirements while dimensional constraints determine the room geometries according to the room proportional requirements. The connectivity constraints, the unoccupied unit constraints, the fixed room location constraints, boundary constraints and the fixed border constraints are classified as functional constraints while the non-intersecting constraints, the overlapping constraints and the length constraints are classified as dimensional constraints, see Figure 7.

1) Functional constraints

- Location constraints explain the relationship between distinct rooms that ensure the non-overlapping of rooms using two binary variables (p_{ij}, q_{ij}), see Figure 8.

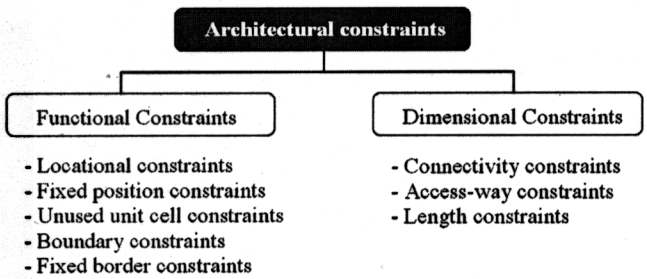


Figure 7. The functional and dimensional constraints.

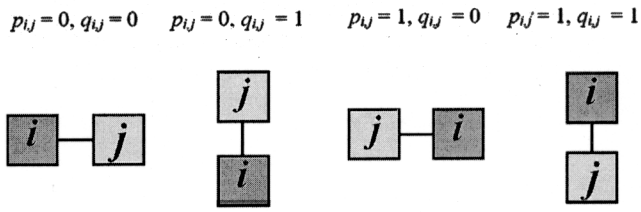


Figure 8. p_{ij} and q_{ij} represent the connection of room i and room j.

$$x_i + w_i \le x_j + W^*(p_{ij} + q_{ij}) \quad i \text{ to the left of } j,\ p_{ij}=0,\ q_{ij}=0 \tag{10}$$

$$y_j + h_j \le y_i + H^*(1 + p_{ij} - q_{ij}) \quad i \text{ below } j,\ p_{ij}=0,\ q_{ij}=1 \tag{11}$$

$$x_j + w_j \le x_i + W^*(1 - p_{ij} + q_{ij}) \quad i \text{ to the right of } j,\ p_{ij}=1,\ q_{ij}=0 \tag{12}$$

$$y_i + h_i \le y_j + H^*(2 - p_{ij} - q_{ij}) \quad i \text{ above } j,\ p_{ij}=1,\ q_{ij}=1 \tag{13}$$

The decision variables p_{ij} and q_{ij} force the room i to the left, the bottom, the right and the above of room j corresponding to constraint 10, 11, 12 and 13. Four possible cases of (p_{ij}, q_{ij}) are (0, 0), (0, 1), (1, 0) and (1, 1) which can easily be checked. The following figures present the i^{th} room placement on the left, the bottom, the right and the top for different values of p_{ij} and q_{ij}.

- Fixed position constraints determine the room positioning in a boundary area. In a practical design, this constraint helps an architect to secure the room location. For example, a high-rise building has the lift core position fixed in every level.

$$x_i = \text{fixed X coordinate} \tag{14}$$

$$y_i = \text{fixed Y coordinate} \tag{15}$$

- Unoccupied unit constraints determine the unusable area. This constraint helps an architect design various orthogonal boundary shapes by avoiding the room positioning placement on an unoccupied unit. We use two binary variables (s_{ik}, t_{ik}) to identify the location of unoccupied unit cell, k^{th}, see Figure 9.

$$x_i \ge x^u_k + 1 - W^*(s_{ik} + t_{ik}) \quad \text{unoccupied space to left of } i \tag{16}$$

$$x^u_k \ge x_i + w_i - W^*(1 + s_{ik} - t_{ik}) \quad \text{unoccupied space to right of } i \tag{17}$$

$$y_i \ge y^u_k + 1 - H^*(1 - s_{ik} + t_{ik}) \quad \text{unoccupied space to top of } i \tag{18}$$

$$y^u_k \ge y_i + h_i - H^*(2 - s_{ik} - t_{ik}) \quad \text{unoccupied space to bottom of } i \tag{19}$$

where x^u_k, y^u_k are unoccupied positions in X, Y coordinate of the unoccupied k^{th} cell.

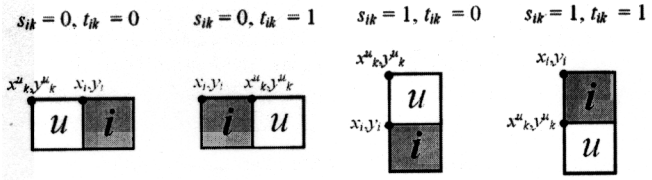


Figure 9. x_i and y_i represent an unoccupied unit (u) of room i on the various directions using the decision variables s_{ik} and t_{ik}.

The decision variables s_{ik} and t_{ik} make sure that location of the room i does not cover a unit cell k, (see Figure 9 for illustrations).

- Boundary constraints force a room to be inside a boundary.

$$x_i + w_i \leq W \quad \text{a room within the horizontal boundary} \quad (20)$$

$$y_i + h_i \leq H \quad \text{a room within the vertical boundary} \quad (21)$$

- Fixed border constraints address the absolute placement of the room. This constraint is divided into four types: the north, the south, the east and the west. For example, a room is positioned to the north if its touch the top border, see Figure 10.

$$y_i = 0 \quad \text{touch the north border} \quad (22)$$

$$y_i + h_i = H \quad \text{touch the south border} \quad (23)$$

$$x_i + w_i = W \quad \text{touch the east border} \quad (24)$$

$$x_i = 0 \quad \text{touch the west border} \quad (25)$$

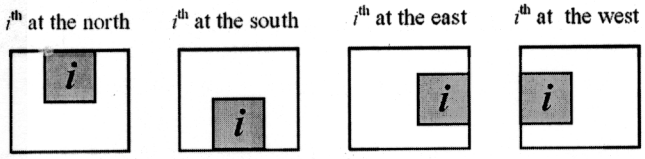


Figure 10. The fixed room i^{th} touch at each layout boundary.

2) Dimensional constraints

Dimensional constraints determine the adjustment of room geometry according to proportional requirements.

- Connectivity constraint forces two connecting rooms to be placed next to one another. We use the same two binary variables p_{ij} and q_{ij} with different set of constraints, see Figure 11.

$$x_i + w_i \geq x_j - W^*(p_{ij} + q_{ij}) \quad i \text{ to the right of } j, \; p_{ij} = 0, q_{ij} = 0 \quad (26)$$

$$y_j + h_j \geq y_i - H^*(1 + p_{ij} - q_{ij}) \quad i \text{ above } j, \quad p_{ij} = 0, q_{ij} = 1 \quad (27)$$

$$x_j + w_j \geq x_i - W^*(1 - p_{ij} + q_{ij}) \quad i \text{ to the left } j, \quad p_{ij} = 1, q_{ij} = 0 \quad (28)$$

$$y_i + h_i \geq y_j - H^*(2 - p_{ij} - q_{ij}) \quad i \text{ below } j, \quad p_{ij} = 1, q_{ij} = 1 \quad (29)$$

Applying the location constraints with connectivity constraints, the room i^{th} is forced to contact room j^{th} at the right, the top, the left and the bottom corresponding the four possible cases of (0, 0), (0, 1), (1, 0) and (1, 1), see Figure 11 for illustration.

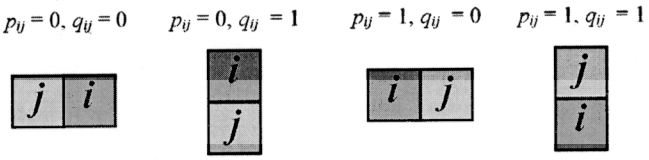


Figure 11. p_{ij} and q_{ij} are reused to formulate the non-intersecting relation between room i and room j.

- Access-way constraints force the minimal contact length between two connected rooms. Two rooms are touching each other with the minimal contact length defined by the value (T_{ij}). For example, the junction between room i and j must be wide enough to accommodate an access way, the same binary variables q_{ij} have been reused. Only q_{ij} has been used due to the fact that the vertical contact is allowed to be placed on the left ($p_{ij} = 0$) or on the right of the room j ($p_{ij} = 1$). This also true for the horizontal contact which ignores the placement of the room i above ($p_{ij} = 0$) and below ($p_{ij} = 1$) the room j, (Figure 12).

$$H^*(q_{ij}) \geq y_i + T_{ij} - y_j - h_j \quad i \text{ upper-vertical contact } j \quad (30)$$

$$H^*(q_{ij}) \geq y_j + T_{ij} - y_i - h_i \quad i \text{ lower-vertical contact } j \quad (31)$$

$$W^*(1 - q_{ij}) \geq x_i + T_{ij} - x_j - w_j \quad i \text{ left-horizontal contact } j \quad (32)$$

$$W^*(1 - q_{ij}) \geq x_j + T_{ij} - x_i - w_i \quad i \text{ right-horizontal contact } j \quad (33)$$

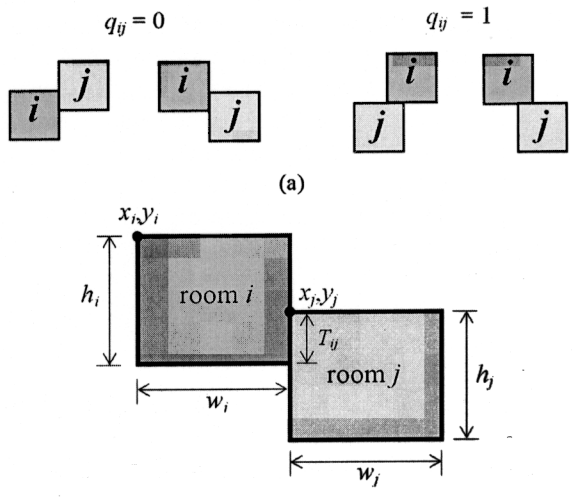


Figure 12. (a) q_{ij} is reused to formulate the overlapping region between room i and room j and (b) T_{ij} represent a minimal contact length between room i and room j.

- Length constraints determine minimal and maximal lengths of the bounded size of each room. A room width and height must be fitted according to its corresponding dimensions between the horizontal range of $w_{min,i}$,$w_{max,i}$ and the vertical range of $h_{min,i}$, $h_{max,i}$ respectively.

$$w_{min,i} \leq w_i \leq w_{max,i} \quad \text{range of width of room } i \qquad (34)$$

$$h_{min,i} \leq h_i \leq h_{max,i} \quad \text{range of height of room } i \qquad (35)$$

3. Non-circular connectivity and guided constraints

Due to the work of Keatruangkamala and Sinapiromsaran (2006), the solution time to solve the architectural layout design problem as the multiobjective mixed integer programming model is prohibitive for a medium to large problem size, even with the state-of-the-art optimization solver. This paper proposes two modeling techniques to alleviate this problem. The first technique is to add all valid inequalities of non-circular configuration of the three consecutive rooms to the base AL-MIP model, called non-circular AL-MIP. The second is to apply guided room location based on architect's preferences that helps eliminate some alternative solutions.

3.1 Non-Circular constraints

The first technique uses valid inequalities (Das *et al.*, 2003; Pefferkorn, 1975) to tighten the feasible region. By which, the LP-relaxation region has been cut off while all integral points are maintained. The remaining LP-relaxation region is strictly smaller than the LP-relaxation of the original one with the corner extreme points are forced to be integral. The notion of the valid inequality can be formulated as follows.

Given the IP (Integer programming problem) as

(IP) $\max\{ c^T x : x \in X \}$

$X = \{ x : Ax \leq b , x \in Z^+ \}$

The inequality $\pi^T x \leq \pi_0$ is called a valid inequality for X if $\pi^T x \leq \pi_0$ for all $x \in X$, (Figure 13).

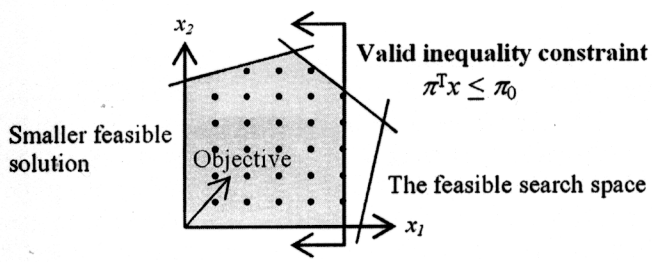


Figure 13. Valid inequality constraints cut off the non-integral feasible region.

The valid inequality for the architectural layout design problem is generated using the concept of non-circular constraints. These inequality constraints are defined among three consecutive rooms i, j and k, connected in this order. The binary variables p_{ij}, p_{ik}, p_{ik}, q_{ij}, q_{ik} and q_{ik} from the base AL-MIP model are used to present room connectivities. The consecutive connectivity of room i and j prohibits the placement of room i between room j and k, see Figure 14. Therefore, the valid inequalities force the non-circulation of the room i and k which eliminate configuration formed by four different directions, top, left, right and bottom of the i^{th} room and the j^{th} room. The non-circular constraints for each direction have been illustrated as follows.

$$p_{ik} - q_{ik} \leq W^*(p_{ij} + q_{ij}) \quad i \text{ to the left of } j, \quad p_{ij}=0, q_{ij}=0 \qquad (36)$$

$$p_{jk} + q_{jk} - 1 \leq W^*(1 + p_{ij} - q_{ij}) \quad i \text{ above } j, \quad p_{ij}=0, q_{ij}=1 \qquad (37)$$

$$1 - p_{ik} - q_{ik} \leq W^*(1 - p_{ij} + q_{ij}) \quad i \text{ to the right of } j, \quad p_{ij}=1, q_{ij}=0 \qquad (38)$$

$$q_{ik} - p_{ik} \leq W^*(2 - p_{ij} - q_{ij}) \quad i \text{ below } j, \quad p_{ij}=1, q_{ij}=1 \qquad (39)$$

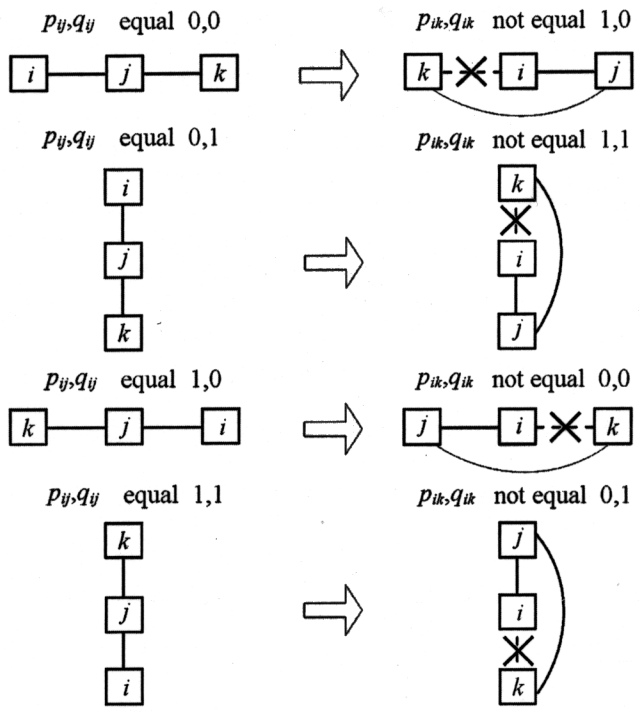


Figure 14. (Left) four possible scenarios defined by consecutive rooms i, j and k and (right) and four corresponding scenarios that are eliminated by valid inequality constraints.

3.2 Guided constraints

Other inequality constraints correspond to the architect's preferences. Traditionally, some room positions in as architectural layout design may often be placed to the

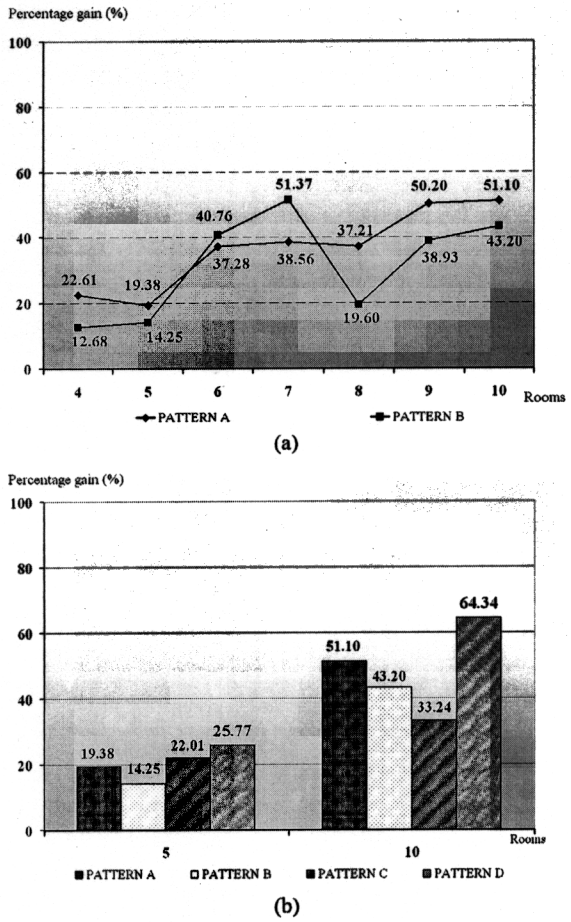


Figure 16. (a) iteration comparisons between pattern A and B and (b) iteration comparisons between room 5 and room 10 for pattern A, B, C and D.

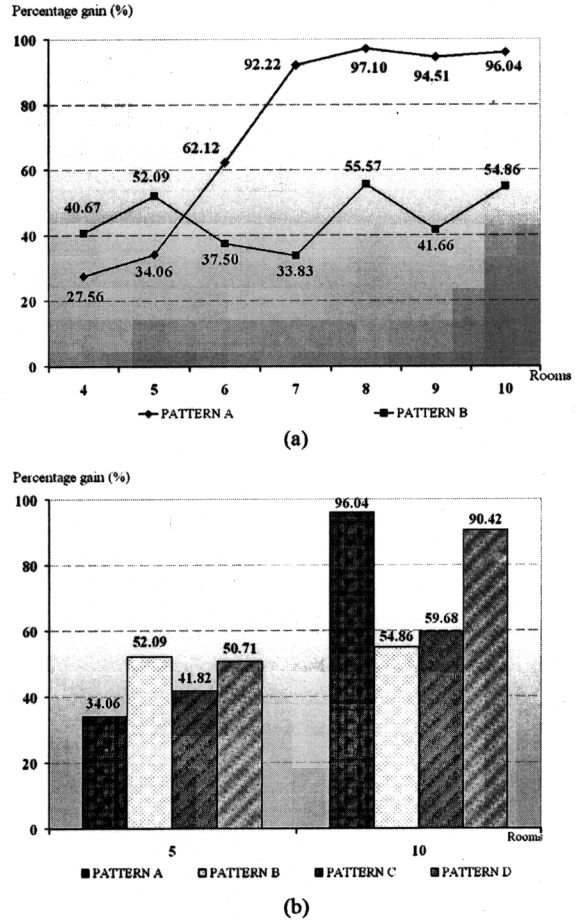


Figure 17. (a) iteration comparisons between pattern A and B and (b) iteration comparisons between 5 rooms and 10 rooms for pattern A, B, C and D.

Table 3. Two stories house example

No.	Room	Width (m.)		Height (m.)		Connect
		min	max	min	max	
1st Floor						
1.	Garage x 2 cars	5	6	5	7	2, 4, South
2.	Living Rm.	5	8	5	8	1, 6
3.	Dining Rm.	5	7	5	7	4, 6
4.	Kitchen	5	6	5	7	1, 3
5.	Staircase	4	4	3	3	6
6.	Hall 1	4	6	4	6	2, 3, 5, 7
7.	Bath	3	4	3	4	6
2nd Floor						
8.	Master bedroom	6	7	6	7	10, 11, East
9.	Bedroom 2	5	7	5	7	10, 11
10.	Hall 2	3	5	3	5	8, 9, 11, 12
11.	Bath	3	4	3	4	8, 9, 10
12.	Staircase	4	4	3	3	10

Remark: unit scale in meter.

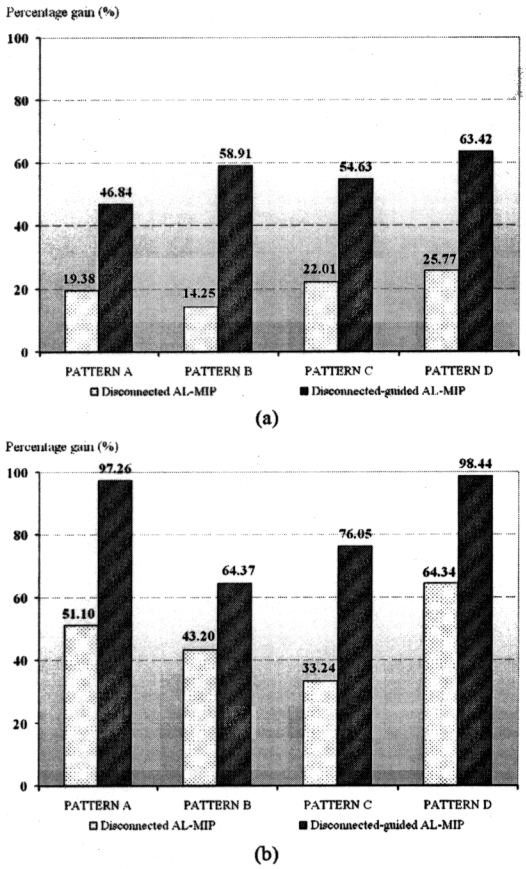


Figure 18. Percentage gain of the number of iterations between non-circular AL-MIP and non-circular and guided AL-MIP for (a) 5 rooms and (b) 10 rooms.

iterations and computational time of the second floor are 48369 and 12.843 seconds, respectively. The optimal layout design is shown in Figure 19.

4.3.2 Case study

The drastic reduction of the computational time shows a practical use of the non-circular and guided AL-MIP to handle the realistic architectural layout design. We compared the performance of our model with a standard one-storey house solved by the genetic algorithm (Chen *et al.*, 1993; Goldberg, 1989) from Romualdas *et al.* (2005). The initial specification of the layout is shown in Table 4 This layout consists of three bedrooms, a bathroom, dining and living room and two hallway connecting among different rooms.

From their paper (Romualdas *et al.*, 2005), the local optimal solution from the genetic algorithm (GA) used 800 generations with 156 seconds. Their experiments have been performed with 94 variables, 156 constraints, crossover probability (P_c) is 0.6 and mutation probability (P_m) is 0.125. The initial population of solution was generated randomly with no feasible initial starting points while crossover and mutation synthesized new solutions. After 300 and 650 generations the intermediate feasible layouts have been reported.

Our AL-MIP model using non-circular and guided constraints determines the global optimal solution of the architectural layout design based on CPLEX solver used 178537 computational iterations, 12575 branch nodes and 105 seconds of computational time. Even though these measures are not comparable between GA and AL-MIP model with non-circular and guided constraints, both solutions are similar.

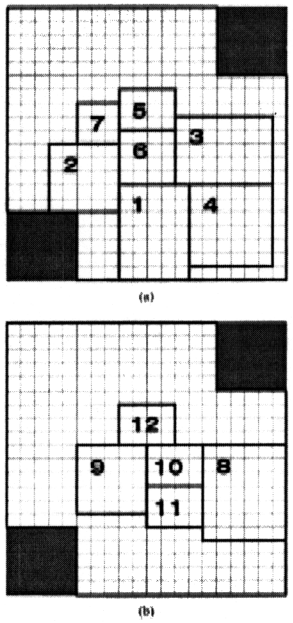


Figure 19. The realistic of the two-storey house solved by the non-circular AL-MIP cooperate with three adjustable constraints and the gray color presents the unoccupied unit spaces.
(a) the computational time of 1st floor plan is 41.625 seconds.
(b) the computational time of 2nd floor plan is 12.843 seconds.

Table 4. Room specifications

No.	Room	Width (m.)		Height (m.)		Connect
		min	max	min	max	
1.	Living Rm.	4	6	5	6	6
2.	Dining Rm.	2	5	4	6	6
3.	Bedroom 1	5	6	3	6	7
4.	Bedroom 2	3	6	4	6	7
5.	Bedroom 3	3	5	4	5	7
6.	Hallway 1	2	5	5	6	1, 2, 7
7.	Hallway 2	3	5	3	5	3, 4, 5, 6, 8
8.	Bathroom	3	5	3	5	7

Remark: unit scale in meter.

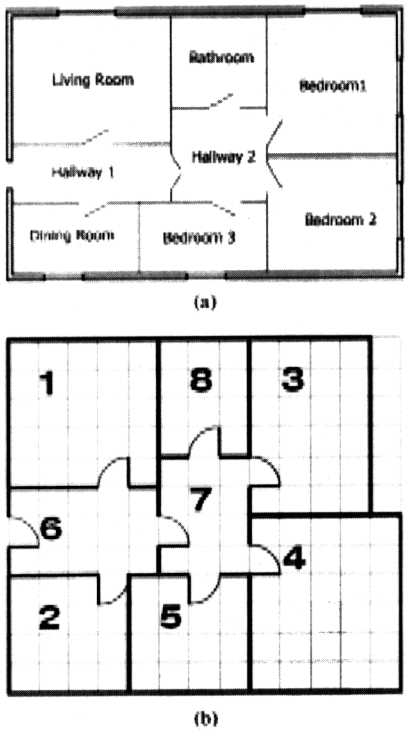


Figure 20. Comparison between (a) a layout design using genetic algorithm (GA) presents 800 generations with 156 seconds and (b) a layout design using non-circular and guided AL-MIP model presents 105 seconds of computational time.

5. Conclusions

From our experiments, the non-circular and guided AL-MIP model performs substantially well with respect to the medium-sized problems. Our approaches incorporate the mathematical concept of valid inequalities and the architect intuition of placing a specific room in the desired direction. The combined model can handle medium-sized problems efficiently gaining an average of ninety percent, taking less than 1,000 seconds or less. The result of our method is comparable with the genetic algorithm (GA) and suitable for practical case studies. Furthermore, the separation of the AL-MIP model into the model section and the data section from GAMS can easily be extended to larger number of connected rooms without affecting the model formulation.

Finally, the non-circular and guided AL-MIP model exhibits significant potential to be used in the early conceptual stage of a design process to help architects balance all their preferences and constraints.

Acknowledgements

The author would like to thank Sinapiromsaran K. from Chulalongkorn University for very useful advice and critical comments concerning the theory of mathematical programming which encouraged me to successively proceed with these techniques. Besides, I would like to thank all staff from the master program in computer-aided architectural design, Rangsit University, which supported me financially for this research.

References

Bloch, C.J. and Krishnamurti, R. 1978. The counting of rectangular dissections. Environment and Planning 1978; 2: 207-214.

Chen, Y.H. and Wang, Y.Z. 1993. Genetic algorithms for optimized retriangulation in the context of reverse engineering. Compute Aid Des; 31(4): 261-71.

Das, A.K., Marks, R.J., El-Sharkawi, M., Arabshani, P. and Gray, A. 2003. Minimum power broadcast trees for wireless networks: integer programming formulations. In Processing of INFOCOM.

Flemming, U. 1978. Representation and generation of rectangular dissections. Annual ACM IEEE Design Automation Conference; 15: 138-144.

Flemming, U. Coyne, R., Glavin, T. and Rychener, M. 1988. A generative expert system for the design of building layouts, Artificial Intelligence in Engineering: Design. Proceedings of the Third International Conference, Palo Alto, CA., J. Gero ed., New York: Elsevier; 2: 445-464.

George, L.N. and Laurence, A.W. 1988. Integer and combinatorial optimization. New York: A Wiley-Interscience Publication. USA.

Gero, J.S. 1990 Design prototypes: a knowledge representation schema for design. AI Magazine;11(4):26-36.

Goldberg, D.E. 1989. Genetic algorithms in search, optimization and machine learning. Massachusetts: Addison-Wesley Pubblishing Company.

Homayouni, H. 2006. A Survey of Computational Approaches to Space Layout Planning (1965-2000). Computational Approaches to Space Layout Planning.

Honda, K. and Mizoguchi, F. 1995. Constraint-based approach for automatic spatial layout planning. 11th Conference on Artificial Intelligence for Applications; 10: 38-48.

Keatruangkamala, K. and Sinapiromsaran, K. 2005. Optimizing Architectural Design via Mixed Integer Programming. Proceeding in CAADFutures;11: 175-184.

Keatruangkamala, K. and Sinapiromsaran, K. 2006. Heuristic cut for identifying the solution of the architectural layout design optimization. Proceeding in the Second Graduate Congress of Mathematics and Physical Science; 2: 138-139.

Li, S.P., Frazer, J.H. and Tang, M.X., 2000. A constraint based generative system for floor layouts. Proceeding in CAADria; 10: 441-450.

Ligett, R.S. and Mitchell, W.J. 1981. Optimal space planning in practice. Computer Aided Design;13(5): 277-288.

Ligett, R.S. 1992. Designer-automated algorithm partnership: an interactive graphic approach to facility lay-

out. *In*: Kalay YE, editor. Evaluating and predicting design performance, New York: Wiley Interscience; 2: 101–123.

Linderoth and Savelsbergh M.W.P. 1999. A computational study of search strategies for mixed integer programming. INFORMS Journal on Computing11: 173-187.

Medjdoub, B. and Yannou, B. 1998. Topological enumeration heuristics in constraint-based space layout planning. AI in Design'98; 271-290.

Medjdoub, B. and Yannou, B. 2000. Seperating topology and geometry in space planning. Computer-Aided Design; 32: 39-61.

Michalek, J. and Papalambros, P.Y. 2002. Interactive layout design optimization. Engineering Optimization; 34 (5): 461-184.

Michalek, J., Choudhary, R. and Papalambros, P.Y. 2002. Architectural layout design optimization. Engineering Optimization; 34(5): 485-501.

Mitchell, W.J. and Steadman, J.P., and Liggett, R.S. 1976. Synthesis and optimization of small rectangular floor plans. Environment and Planning B; 3(1): 37-70.

O'Sullivan, B. 1999. Constraint-aided conceptual design. PhD thesis, Dept of Computer Science, University College Cork, Ireland.

Pefferkorn, G.E. 1975. A Heuristic problem solving design system for equipment or furniture layouts. Communications of the ACM; 18: 286-297.

Romualdas, B., and Ina, P. 2005. Optimization of architectural layout by the improved genetic algorithm. Journal of Civil Engineering and Management; 11(1): 13-21.

Russell, D.M., Venkat, N. and Pamela, H.V. 1999. Optimal facility layout design. Operational Research Letters; 23: 117-127.

Schwarz, A., Berry, D.M. and Shaviv, E. 1994. On the use of the automated building design system. Computer Aided Design; 26: 747-762.

Schwarz, A., Berry, D.M. and Shaviv, E. 1994. Representing and solving the automated building design problem. Computer-Aided Design; 26(9): 689-698.

Scott, A. and Donald, H. 1999. Making Design Come Alive: Using Physically Based Modelling Techniques in Space Layout Planning. CAADFutures; 7: 245-262.

Simon, H.A. 1973. The structure of ill-structured problems. Artificial Intelligence; 4: 181-201.

Willoughby, T., Paterson, W. and Drummond, G. 1970. Computer aided architectural planning. Operational Research Quarterly; 21: 91-98.

north, the south, the east and the west directions; for example, architects fix a bedroom on the north or the east direction to avoid the sunlight corresponding to the benign Feng Shui (Chinese belief). According to this traditional belief, we define constraints to allocate the bedroom on a required direction. By which, we proposed the North-South-East-West guided constraints, called guided AL-MIP to allocate the room positioning based on an architect's preference. The following constraints present the allocation of the guided room i' for all possible room j.

$y_{i'} \le y_j$ guided room i' to the north direction (40)

$x_{i'} \le x_j$ guided room i' to the west direction (41)

$x_j + w_j \le x_{i'} + w_{i'}$ guided room i' to the east direction (42)

$y_j + h_j \le y_{i'} + h_{i'}$ guided room i' to the south direction (43)

where $x_{i'}$ and $y_{i'}$ are coordinates of guided room i' in a required direction,
x_j and y_j are coordinates of room j.

4. Experiments

The experiments presented in this paper have been carried out on a PC computer using Pentium 1.6 MHz and 1.5 GB of memory using GAMS with CPLEX 9.0. In order to measure the performance growth, we simulated architectural layout design instances with 4, 5, 6, 7, 8, 9 and 10 rooms based on four distinct configurations which are 1) a linear configuration, 2) a rail configuration, 3) a connected wheel configuration and 4) a nested wheel configuration, (Figure 15). Each configuration is composed of five instances and the average measurements were recorded. Total of 140 experimental runs were tested and gathered. The boundary area is set on 100×100 square meters and defined the minimum and maximum room widths and heights between 5 to 10 meters. We collected objective values, the number of iterations, the computational time and the memory usage, which are illustrated in Table 1 and Table 2.

4.1 Non-circular AL-MIP experiments

We run the experiments for medium-sized instances. Table I illustrates the experimental results between base AL-MIP and non-circular AL-MIP.

Table 1 shows the objective value, the number of iterations, the computational time in seconds, the memory usage (MB) and the percentage gain from the base AL-MIP of four architectural patterns varying from 4 to 10 rooms. The column of the objective value is used to compare the optimal solutions from the base AL-MIP and the non-circular AL-MIP. All experiments have the same objective values even though they have different coordinates.

To measure the performance between base AL-MIP and non-circular AL-MIP, we report the number of iterations and total computational time of both models. The percentage gain is computed by subtracting a measure (iteration or computational time) of the non-circular AL-MIP from the base AL-MIP. The larger the positive value is, the better the gain will be. Different patterns have different percentage gains. For a linear pattern (Pattern A), the average iteration gain and computational time gain are 50.34 and 50.33, respectively. For a rail pattern (Pattern B), the average iteration gain and computational time gain are 41.11 and 34.51. For a connected wheel pattern (Pattern C), the average iteration gain and computational time gain are 30.85 and 27.46. For a nested wheel pattern (Pattern D), the average iteration gain and computational time gain are 65.09 and 64.05. Overall patterns, the iteration gains are more than 20% and 51% for non-circular AL-MIP of 5 rooms and 10 rooms respectively. For the linear configuration (pattern A) and the rail configuration (pattern B), we can achieve more than 30% of the iteration improvement over 7 rooms. This is due to the structural connectivity which is composed of a large number repeated patterns of connections. Moreover, the memory usages also improve for larger problem sizes due to the small number of iterations.

Figure 16(a), a linear configuration (pattern A) presents the increasing percentage gains from 4 to 10 rooms.

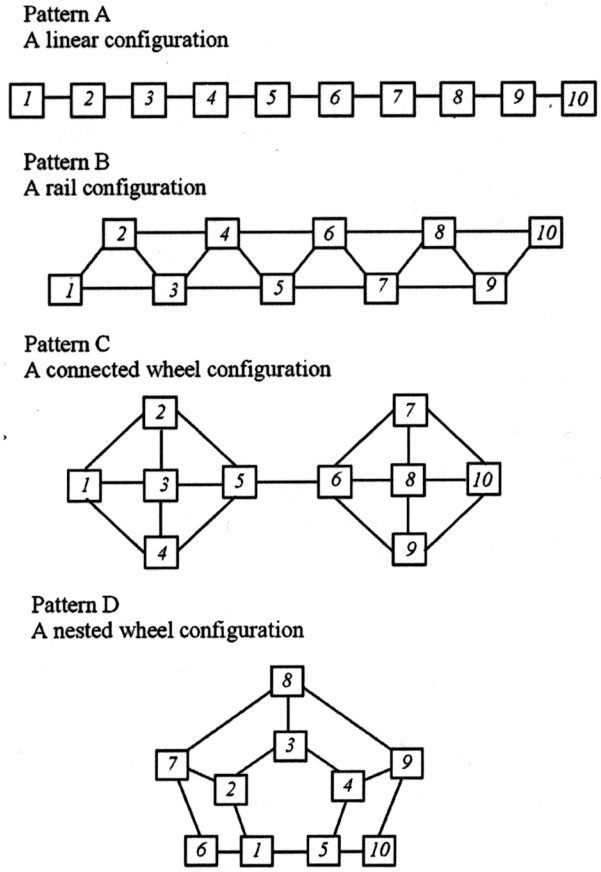


Figure 15. The distinct pattern A, B, C and D present 10 room configurations.

Table 1. Comparison between base AL-MIP and non-circular AL-MIP

Room No.	Patterns	Objective value		Iterations		Computational time (sec)		Memory usage (MB)		Percentage gains (%)	
		Base AL-MIP	Non-circular AL-MIP	Base AL-MIP	Non-circular AL-MIP	Base AL-MIP	Non-circular AL-MIP	Base AL-MIP	Non-circular AL-MIP	Iteration (%)	Time (sec)
4 rm.	A	15	15	1097	849	0.156	0.125	0.00	0.00	22.61	19.87
	B	17	17	1191	1040	0.203	0.187	0.00	0.00	12.68	7.88
	C	18	18	1390	1212	0.156	0.140	0.00	0.00	12.61	10.26
	D	15	15	1071	1006	0.140	0.125	0.00	0.00	6.07	10.71
5 rm.	A	50	50	13147	10599	0.899	0.765	0.04	0.04	19.38	14.91
	B	50	50	8884	7618	0.686	0.609	0.02	0.01	14.25	11.22
	C	55	55	11101	8658	0.796	0.609	0.07	0.02	22.01	23.49
	D	50	50	15184	11271	1.031	0.750	0.02	0.05	25.77	27.26
6 rm.	A	90	90	184446	115680	14.828	9.046	0.37	0.28	37.28	38.99
	B	93	93	43795	25943	2.875	1.937	0.07	0.02	40.76	32.63
	C	104	104	61113	26227	4.656	1.968	0.07	0.05	57.08	57.73
	D	94	94	226719	126238	18.093	10.187	0.27	0.19	44.32	43.70
7 rm.	A	150	150	5251485	3226400	802.437	449.265	10.90	7.21	38.56	44.01
	B	166	166	175561	85368	13.312	6.953	0.19	0.11	51.37	49.29
	D	165	165	201216	135517	16.375	10.578	0.20	0.18	32.65	35.40
	C	151	151	1110458	1014039	112.015	101.046	1.08	1.15	8.68	9.79
8 rm.	A	225	225	164927909	103553047	28611.765	19742.890	483.49	336.28	37.21	31.00
	B	239	239	779762	626950	67.156	54.859	0.41	0.45	19.60	18.31
	C	250	250	868866	442377	91.125	44.750	0.83	0.40	49.09	50.89
	D	232	232	9066768	4862432	1487.656	792.000	8.69	4.87	46.37	46.76
9 rm.	A	310	310	1355233860	674859635	191713.715	95905.441	783.49	520.05	50.20	49.97
	B	350	350	2895371	1768215	287.437	179.640	1.00	1.20	38.93	37.50
	C	353	353	3571721	2849157	430.171	347.203	2.87	1.86	20.23	19.29
	D	324	324	240923666	75485112	44748.906	14949.437	241.00	84.21	68.67	66.59
10 rm.	A	425	425	3178839384	1524459635	784127.587	383167.441	1466.30	921.90	51.10	51.13
	B	462	462	14148776	8037023	1874.984	1274.984	5.35	2.45	43.20	32.00
	C	496	496	10758972	5650376	1455.859	904.078	4.69	3.44	47.48	37.90
	D	449	449	823668688	293742408	182469.374	66402.515	455.35	320.14	64.34	63.61

The non-circular constraints can effectively reduce the feasible search space of consecutive room connections, for example, the five room configuration 1-2-3-4-5 is used to construct three non-circular constraints, the first constraint deals with rooms 1-2-3, the second constraint deals with rooms 2-3-4 and the last constraint deals with rooms 3-4-5.

A rail configuration (pattern B) shows a significant drop of iteration percentage gain when there are eight rooms which differs from a linear configuration. This phenomenon is caused by the presolved modules in the preprocess of CPLEX solver that reduces the small number of cuts in the case of eight room configuration.

4.2 Non-circular and guided AL-MIP experiments

Additional North-South-East-West guided constraints are added to the non-circular AL-MIP formulation, called non-circular and guided AL-MIP to abandon some alternative solutions. The concept is to utilize the architect's expert opinion to place a specific room in the required direction. In this experiment, we specify the first room at top left corner (North-West corner).

Table 2 shows the number of iterations and the computational time in seconds among the non-circular AL-MIP and the non-circular and guided AL-MIP for four architectural patterns varying rooms from 4 to 10 rooms. The result is that the non-circular AL-MIP uses more iterations and time than the non-circular and guided AL-MIP. The percentage gains of the number of iterations of the non-circular and guided AL-MIP model are reported in the last column. Moreover, in pattern A, an average iteration gain of more than 70% while in pattern B and pattern C, average iteration gains of 45% and 38%, were achieved respectively. In pattern D, the average significant iteration gain presents more than 70%.

Figure 17(a), a linear configuration (pattern A) presents the increasing percentage gains from 4 to 10 rooms.

Table 2. The comparison results comparison between non-circular AL-MIP and non-circular and guided AL-MIP

Rooms		Iterations		Computational time (sec.)		Iteration gains (%)
		Non-circular AL-MIP	Non-circular and guided AL-MIP	Non-circular AL-MIP	Non-circular and guided Al-MIP	Percentage
4 rm.	A	849	615	0.125	0.093	27.56
	B	1040	617	0.187	0.140	40.67
	C	1212	655	0.187	0.140	45.96
	D	1006	551	0.125	0.125	45.23
5 rm.	A	10599	6989	0.765	0.578	34.06
	B	7618	3650	0.609	0.406	52.09
	C	8658	5037	0.609	0.515	41.82
	D	11271	5555	0.750	0.421	50.71
6 rm.	A	115680	43814	9.046	3.375	62.12
	B	25943	16214	1.937	1.343	37.50
	C	26227	25500	1.968	1.953	2.77
	D	126238	27074	10.187	2.171	78.55
7 rm.	A	32226400	251128	449.265	23.546	92.22
	B	85368	56489	6.750	6.500	33.83
	C	135517	92226	10.578	7.343	31.95
	D	1014039	134698	101.046	10.797	86.72
8 rm.	A	103553047	3002986	19475.625	287.250	97.10
	B	626950	278537	54.859	24.546	55.57
	C	442377	362324	44.750	36.875	18.10
	D	4862432	1746258	792.000	200.406	64.09
9 rm.	A	674859635	37080970	95905.441	7156.046	94.51
	B	1768215	1031533	179.640	107.828	41.66
	C	2849157	855421	347.203	73.453	69.98
	D	75485112	3767553	14949.437	527.390	95.01
10 rm.	A	1554459635	61545177	383167.441	33568.822	96.04
	B	8037023	3628086	1274.984	474.468	54.86
	C	5650376	2278195	904.078	313.984	59.68
	D	293742408	28143987	66402.515	5132.046	90.42

The guided constraints help reduce the feasible search space of consecutive room connections drastically which the medium room number (7-10 rooms) shows the percentage raises up to 90 percents, while the rail configuration (pattern B) shows a fluctuated trend of percentage gain. Similar to the non-circular constraints effect, the presolved modules in the preprocess of CPLEX solver reduces a small number of cuts.

Finally, Figure 18 illustrates the iteration percentage gains among the base AL-MIP, non-circular AL-MIP and non-circular and guided AL-MIP for 5 and 10 rooms. The non-circular and guided AL-MIP presents more than 40% and 75% achievement of the number of iterations and computational time reduction among all patterns. Especially, more than 90% gain is achievable for the pattern A and the pattern D while more than 55% gain for the pattern B and the pattern C.

4.3 Practical experiments

4.3.1 Two stories house experiments

In this section, we present the usage of fixed position constraint, fixed border constraint and unoccupied unit constraints for a non-circular AL-MIP model. This example allows us to construct non-rectangular boundary shape which is motivated by the staircase area.

To exhibit the flexibility of these three constraints, a realistic two-storey house allocated on asymmetric boundary, has been solved using the non-circular AL-MIP model. The initial specification of the requirements is shown in Table 3.

Based on our non-circular AL-MIP model, the total computational time of this two- storey house is 54.468 seconds. The number of iterations and computational time of the first floor are 179282 and 41.625 seconds while the

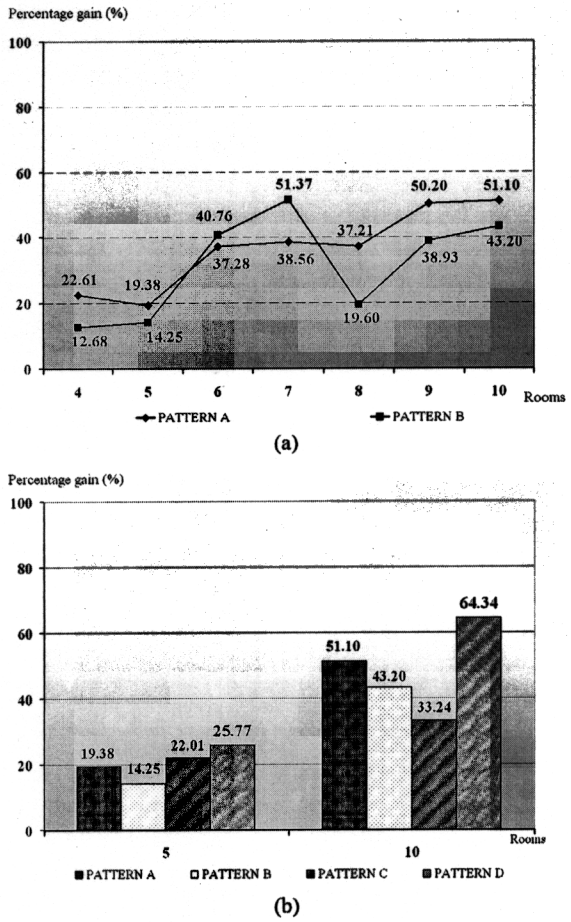


Figure 16. (a) iteration comparisons between pattern A and B and (b) iteration comparisons between room 5 and room 10 for pattern A, B, C and D.

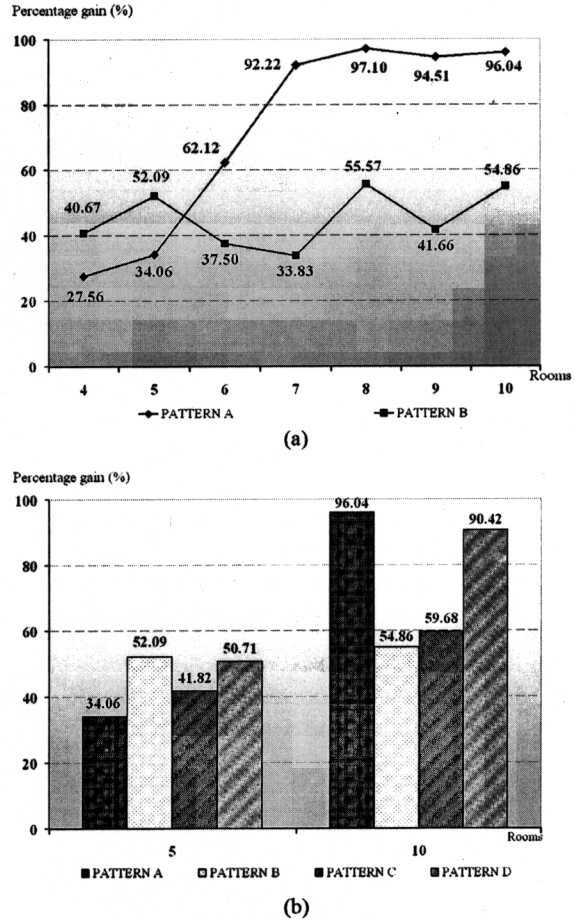


Figure 17. (a) iteration comparisons between pattern A and B and (b) iteration comparisons between 5 rooms and 10 rooms for pattern A, B, C and D.

Table 3. Two stories house example

No.	Room	Width (m.)		Height (m.)		Connect
		min	max	min	max	
1st Floor						
1.	Garage x 2 cars	5	6	5	7	2, 4, South
2.	Living Rm.	5	8	5	8	1, 6
3.	Dining Rm.	5	7	5	7	4, 6
4.	Kitchen	5	6	5	7	1, 3
5.	Staircase	4	4	3	3	6
6.	Hall 1	4	6	4	6	2, 3, 5, 7
7.	Bath	3	4	3	4	6
2nd Floor						
8.	Master bedroom	6	7	6	7	10, 11, East
9.	Bedroom 2	5	7	5	7	10, 11
10.	Hall 2	3	5	3	5	8, 9, 11, 12
11.	Bath	3	4	3	4	8, 9, 10
12.	Staircase	4	4	3	3	10

Remark: unit scale in meter.

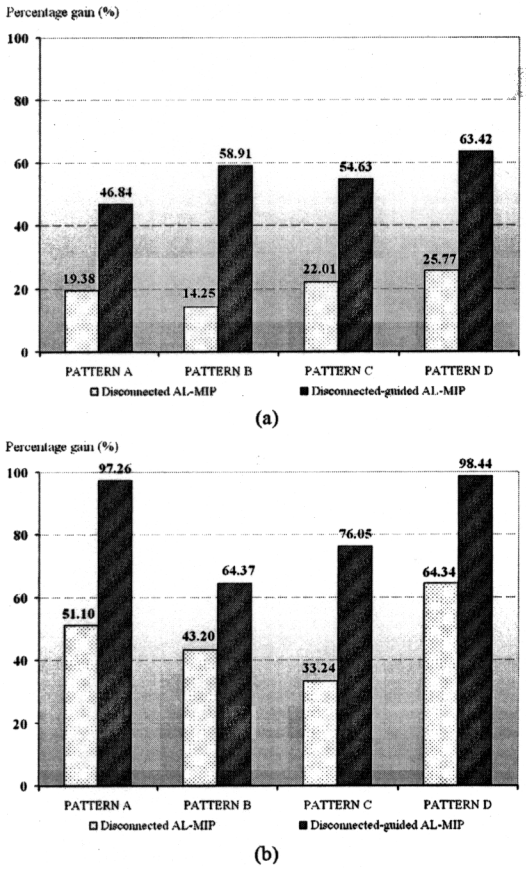


Figure 18. Percentage gain of the number of iterations between non-circular AL-MIP and non-circular and guided AL-MIP for (a) 5 rooms and (b) 10 rooms.

iterations and computational time of the second floor are 48369 and 12.843 seconds, respectively. The optimal layout design is shown in Figure 19.

4.3.2 Case study

The drastic reduction of the computational time shows a practical use of the non-circular and guided AL-MIP to handle the realistic architectural layout design. We compared the performance of our model with a standard one-storey house solved by the genetic algorithm (Chen *et al.*, 1993; Goldberg, 1989) from Romualdas *et al.* (2005). The initial specification of the layout is shown in Table 4 This layout consists of three bedrooms, a bathroom, dining and living room and two hallway connecting among different rooms.

From their paper (Romualdas *et al.*, 2005), the local optimal solution from the genetic algorithm (GA) used 800 generations with 156 seconds. Their experiments have been performed with 94 variables, 156 constraints, crossover probability (P_c) is 0.6 and mutation probability (P_m) is 0.125. The initial population of solution was generated randomly with no feasible initial starting points while crossover and mutation synthesized new solutions. After 300 and 650 generations the intermediate feasible layouts have been reported.

Our AL-MIP model using non-circular and guided constraints determines the global optimal solution of the architectural layout design based on CPLEX solver used 178537 computational iterations, 12575 branch nodes and 105 seconds of computational time. Even though these measures are not comparable between GA and AL-MIP model with non-circular and guided constraints, both solutions are similar.

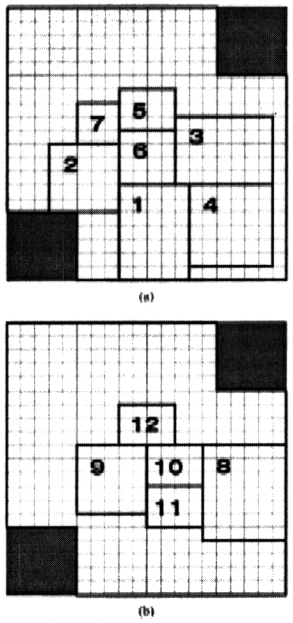


Figure 19. The realistic of the two-storey house solved by the non-circular AL-MIP cooperate with three adjustable constraints and the gray color presents the unoccupied unit spaces.
(a) the computational time of 1st floor plan is 41.625 seconds.
(b) the computational time of 2nd floor plan is 12.843 seconds.

Table 4. Room specifications

No.	Room	Width (m.)		Height (m.)		Connect
		min	max	min	max	
1.	Living Rm.	4	6	5	6	6
2.	Dining Rm.	2	5	4	6	6
3.	Bedroom 1	5	6	3	6	7
4.	Bedroom 2	3	6	4	6	7
5.	Bedroom 3	3	5	4	5	7
6.	Hallway 1	2	5	5	6	1, 2, 7
7.	Hallway 2	3	5	3	5	3, 4, 5, 6, 8
8.	Bathroom	3	5	3	5	7

Remark: unit scale in meter.

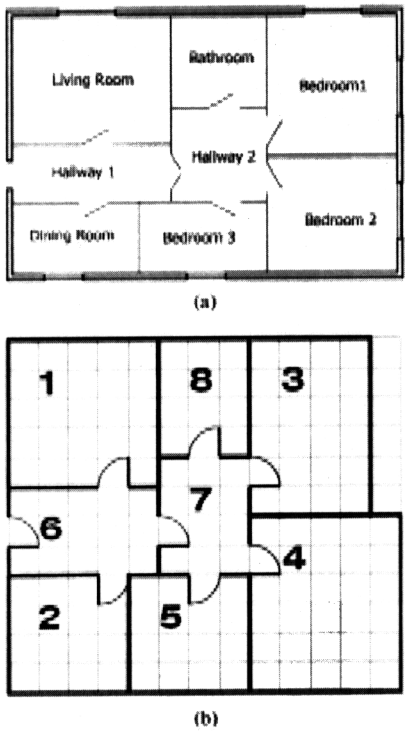


Figure 20. Comparison between (a) a layout design using genetic algorithm (GA) presents 800 generations with 156 seconds and (b) a layout design using non-circular and guided AL-MIP model presents 105 seconds of computational time.

5. Conclusions

From our experiments, the non-circular and guided AL-MIP model performs substantially well with respect to the medium-sized problems. Our approaches incorporate the mathematical concept of valid inequalities and the architect intuition of placing a specific room in the desired direction. The combined model can handle medium-sized problems efficiently gaining an average of ninety percent, taking less than 1,000 seconds or less. The result of our method is comparable with the genetic algorithm (GA) and suitable for practical case studies. Furthermore, the separation of the AL-MIP model into the model section and the data section from GAMS can easily be extended to larger number of connected rooms without affecting the model formulation.

Finally, the non-circular and guided AL-MIP model exhibits significant potential to be used in the early conceptual stage of a design process to help architects balance all their preferences and constraints.

Acknowledgements

The author would like to thank Sinapiromsaran K. from Chulalongkorn University for very useful advice and critical comments concerning the theory of mathematical programming which encouraged me to successively proceed with these techniques. Besides, I would like to thank all staff from the master program in computer-aided architectural design, Rangsit University, which supported me financially for this research.

References

Bloch, C.J. and Krishnamurti, R. 1978. The counting of rectangular dissections. Environment and Planning 1978; 2: 207-214.

Chen, Y.H. and Wang, Y.Z. 1993. Genetic algorithms for optimized retriangulation in the context of reverse engineering. Compute Aid Des; 31(4): 261-71.

Das, A.K., Marks, R.J., El-Sharkawi, M., Arabshani, P. and Gray, A. 2003. Minimum power broadcast trees for wireless networks: integer programming formulations. In Processing of INFOCOM.

Flemming, U. 1978. Representation and generation of rectangular dissections. Annual ACM IEEE Design Automation Conference; 15: 138-144.

Flemming, U. Coyne, R., Glavin, T. and Rychener, M. 1988. A generative expert system for the design of building layouts, Artificial Intelligence in Engineering: Design. Proceedings of the Third International Conference, Palo Alto, CA., J. Gero ed., New York: Elsevier; 2: 445-464.

George, L.N. and Laurence, A.W. 1988. Integer and combinatorial optimization. New York: A Wiley-Interscience Publication. USA.

Gero, J.S. 1990 Design prototypes: a knowledge representation schema for design. AI Magazine;11(4):26-36.

Goldberg, D.E. 1989. Genetic algorithms in search, optimization and machine learning. Massachusetts: Addison-Wesley Pubblishing Company.

Homayouni, H. 2006. A Survey of Computational Approaches to Space Layout Planning (1965-2000). Computational Approaches to Space Layout Planning.

Honda, K. and Mizoguchi, F. 1995. Constraint-based approach for automatic spatial layout planning. 11th Conference on Artificial Intelligence for Applications; 10: 38-48.

Keatruangkamala, K. and Sinapiromsaran, K. 2005. Optimizing Architectural Design via Mixed Integer Programming. Proceeding in CAADFutures;11: 175-184.

Keatruangkamala, K. and Sinapiromsaran, K. 2006. Heuristic cut for identifying the solution of the architectural layout design optimization. Proceeding in the Second Graduate Congress of Mathematics and Physical Science; 2: 138-139.

Li, S.P., Frazer, J.H. and Tang, M.X., 2000. A constraint based generative system for floor layouts. Proceeding in CAADria; 10: 441-450.

Ligett, R.S. and Mitchell, W.J. 1981. Optimal space planning in practice. Computer Aided Design;13(5): 277-288.

Ligett, R.S. 1992. Designer-automated algorithm partnership: an interactive graphic approach to facility lay-

out. *In*: Kalay YE, editor. Evaluating and predicting design performance, New York: Wiley Interscience; 2: 101–123.

Linderoth and Savelsbergh M.W.P. 1999. A computational study of search strategies for mixed integer programming. INFORMS Journal on Computing11: 173-187.

Medjdoub, B. and Yannou, B. 1998. Topological enumeration heuristics in constraint-based space layout planning. AI in Design'98; 271-290.

Medjdoub, B. and Yannou, B. 2000. Seperating topology and geometry in space planning. Computer-Aided Design; 32: 39-61.

Michalek, J. and Papalambros, P.Y. 2002. Interactive layout design optimization. Engineering Optimization; 34 (5): 461-184.

Michalek, J., Choudhary, R. and Papalambros, P.Y. 2002. Architectural layout design optimization. Engineering Optimization; 34(5): 485-501.

Mitchell, W.J. and Steadman, J.P., and Liggett, R.S. 1976. Synthesis and optimization of small rectangular floor plans. Environment and Planning B; 3(1): 37-70.

O'Sullivan, B. 1999. Constraint-aided conceptual design. PhD thesis, Dept of Computer Science, University College Cork, Ireland.

Pefferkorn, G.E. 1975. A Heuristic problem solving design system for equipment or furniture layouts. Communications of the ACM; 18: 286-297.

Romualdas, B., and Ina, P. 2005. Optimization of architectural layout by the improved genetic algorithm. Journal of Civil Engineering and Management; 11(1): 13-21.

Russell, D.M., Venkat, N. and Pamela, H.V. 1999. Optimal facility layout design. Operational Research Letters; 23: 117-127.

Schwarz, A., Berry, D.M. and Shaviv, E. 1994. On the use of the automated building design system. Computer Aided Design; 26: 747-762.

Schwarz, A., Berry, D.M. and Shaviv, E. 1994. Representing and solving the automated building design problem. Computer-Aided Design; 26(9): 689-698.

Scott, A. and Donald, H. 1999. Making Design Come Alive: Using Physically Based Modelling Techniques in Space Layout Planning. CAADFutures; 7: 245-262.

Simon, H.A. 1973. The structure of ill-structured problems. Artificial Intelligence; 4: 181-201.

Willoughby, T., Paterson, W. and Drummond, G. 1970. Computer aided architectural planning. Operational Research Quarterly; 21: 91-98.