N. Gu, S. Watanabe, H. Erhan, M. Hank Haeusler, W. Huang, R. Sosa (eds.), *Rethinking Comprehensive Design: Speculative Counterculture, Proceedings of the 19th International Conference on Computer-Aided Architectural Design Research in Asia CAADRIA 2014,* 275–284.

ENUMERATION OF FLOOR PLANS BASED ON ZERO-SUPPRESSED BINARY DECISION DIAGRAM

ATSUSHI TAKIZAWA[1], YUSHI MIYATA[2] and NAOKI KATOH[2]
[1] *Osaka City University, Osaka, Japan*
takizawa@arch.eng.osaka-cu.ac.jp
[2] *Kyoto University, Kyoto, Japan*
{as-miyata, naoki}@archi.kyoto-u.ac.jp

Abstract. Open Building (Habraken, 1972) has been the focus of attention due to growing interest in the sustainable society. For Open Building, it is important to preserve the diversity of feasible floor plans in order to adapt to various lifestyles of residents. Capacity analysis is a method for evaluating the potential diversity. We propose a novel method that evaluates the potential diversity of floor plans by enumerating all feasible floor plans satisfying given constraints based on zero-suppressed binary decision diagram (ZDD) (Minato, 1993).

Keywords. ZDD; floor plan; enumeration; Open Building; diversity.

1. Introduction

Open Building has been the focus of attention due to growing interest in the sustainable society. Open Building regards space as three levels; town block (called urban tissue), buildings (support or skeleton), floor plan, interior and facilities (infill). Design and maintenance of such buildings are divided into those three levels. The length of the lifetime much differs in three levels, and the former is longer. Dwellers can live in such buildings over a long period of time according to the change of living needs only by changing elements of infill. This leads to resource saving and economy. The variability and diversity of infill, especially floor plans, are important indices for open buildings. For example, Kadowaki et al. (2000) analyzed statistically the degree of renovation potential of existing multidwellings. Though their study focuses on evaluating the variability of infill, few researches on the evaluation of diversity of floor plan seem to have been done ever. The evaluation of the diversity seems not to be easy because the number of possible floor plans might sometimes become large and difficult to imagine for human. Therefore, it is worth to consider the possibility of the computer aided approach such as floor planning problems.

The floor planning problem has been widely studied from the early days when the computer was applied in practical use (Manning, 1964). Lobs and Donath (2010) reviewed those studies and categorized them from some viewpoints such as approach and implementation. Our proposed method is categorized as the constraint based approach and is implemented by enumeration algorithms. A lot of previous studies formalized this problem as optimization problems. Since the main goal of those formalizations is to find one solution that indicates the best score and satisfies all constraints, those formalizations are inadequate for evaluating the diversity of floor plans. Meanwhile, since the enumeration algorithm used in this study finds all floor plans satisfying given constraints, we might evaluate the diversity of the floor plans by simply counting the number of enumerated floor plans. However, the number of solutions of an enumeration problem increases in an exponential order of the number of variables, and combinatorial explosion tends to occur even with the small number of variables.

In order to tackle this problem, we propose the method for enumerating, indexing and extracting floor plans satisfying given constraints within practical amount of time based on a compressed data structure called ZDD that is a compact expression of logic functions, and a frontier method (Saitoh et al., 2011) that directly constructs a ZDD without requiring a full binary tree to be generated. Mitchell et al. (1976) proposed the method that adjacency graphs of rooms are enumerated by depth-first search and then room shapes are optimized by mathematical programming. However, in their study the room shape is limited to be rectangular, no efficient data structure adequate for enumeration was proposed and floor plans with only eight rooms could be enumerated. Nakano (2002) has also proposed an enumeration method for floor plans. Nakano's method is based on the reverse search (Avis and Fukuda, 1996) and efficient. Though, the room shape is also limited to be rectangular. On the other hand, our proposed method can deal with complex room shapes, more than ten rooms and, in addition, can execute several types of queries.

The remainder of the present paper is organized as follows. In the next section, we explain the outline of ZDD. Section 3 explains the proposed method. Section 4 explains the result of case study. Section 5 concludes the article.

2. ZDD

Here, we explain a ZDD with a simplified example of the proposed method described in Section 3. Figure 1(a) illustrates the problem. There are two empty cells numbered with 0 and 1 and two blocks labelled with A and B. The problem is to enumerate all assignments that each block is assigned to exactely one cell without duplication. This problem can be solved naively by building the binary decision tree illustrated in Figure 1(b). The label on a node represents a combination of a cell number and the kind of a block; e.g. 0A denotes that block A is assigned to the 0th cell. The outgoing dotted and solid lines from a node indicate 0-arc and 1-arc, respectively. If and only if the line from a node is solid (i.e. 1-arc), the corresponding block is assigned to the corresponding cell according to the label of the node. A Boolean value inside a rounded square is called 1-terminal node or 0-terminal node. If an assignment is feasible, 1-terminal node is connected from the last node of the path. If not, 0-terminal node is connected. A binary tree requires up to $\sum_{i=0}^{mn+1} 2^i$ nodes to represent all assignments where m and n represent the numbers of cells and blocks respectively. On the other hand, a ZDD can compress the set of assignments as illustrated in Figure 1(c).

A ZDD can be constructed by applying the following three reductions to a binary tree: pruning, removing the node whose 1-arc is connected to 0-terminal node and merging equivalent nodes. In order to construct a ZDD, we use a frontier method (Saitoh et al., 2011) that is an efficient search method without explicitly constructing binary tree on the way for enumeration.

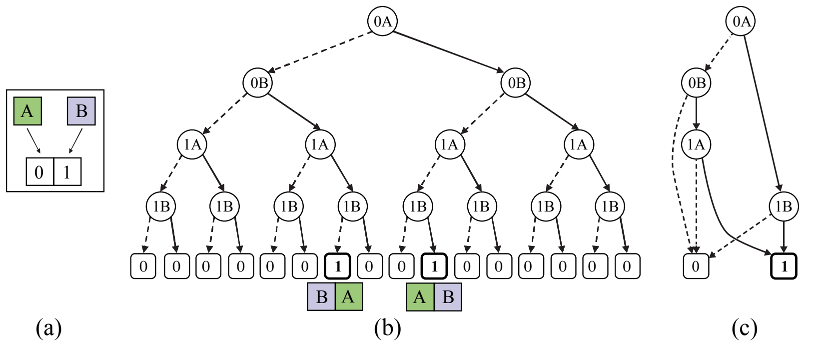


Figure.1 Representation of all cell layouts for problem by (a) a binary tree, and a ZDD (c).

Yet an even more important and widely appreciated virtue of ZDDs is that we can efficiently perform fundamental set operations directly over ZDDs. For example, basic set operations for ZDDs of sets A and B are union ($A \cup B$), product ($A \cap B$), difference ($A \setminus B$), direct product $A \times B$ and so on.

In this study, we use the ZDD library developed by Minato Discrete Structure Manipulation System Project (Hokkaido University, 2012). The efficiency of the set operations depends on the size of the ZDDs rather than the cardinality of the combinations in the target sets. We can execute these operations in a time almost proportional to the size of the number of nodes of a ZDD. We can do it efficiently without restoring the compressed data to the original one. In Section 3.4 we define some queries with ZDD operations for extracting user specific floor plans.

3. Enumeration method of floor plans

3.1. CONFIGURATION SPACE AND ROOMS

Throughout this study, we explain the proposed method with the floor plan of an actual building, UR Hanabatake apartment built in 1964 in Tokyo. As illustrated in Figure 2, the floor is divided into unit cells and which we call a configuration space. Let C be a set of cells and $c \in C$ be a cell. The width and the height of a cell are set to 0.95m and 0.93m respectively. A cell is assigned a room usage such as a living room etc. Let U be a set of room usage and $u \in U$ be a room usage. A cell on or besides an external wall may represent one of facilities $B = \{$simple exterior wall, window, alcove and balcony$\}$. Cells are indexed from top left to bottom right. The index corresponds to the order of ZDD nodes. The ZDD representation of a floor plan is described in the next section.

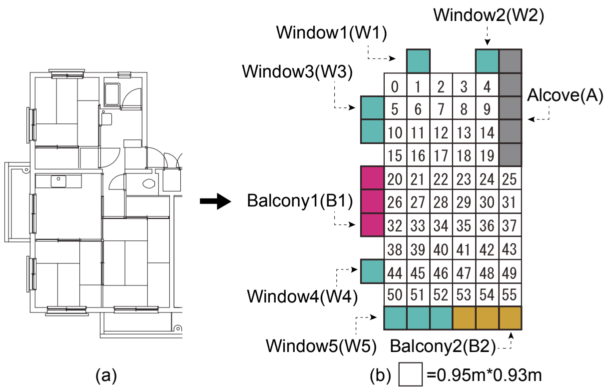
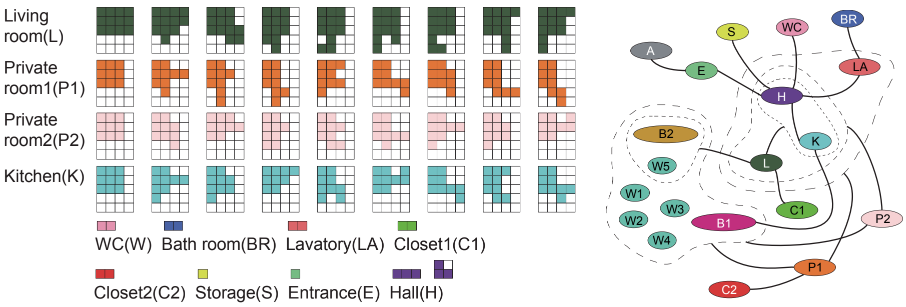


Figure 2. (a) A floor plan of UR Hanabatake apartment and (b) its configuration space.

We have to provide room usages and shapes in advance. In reference to the actual floor plan of Hanabatake apartment, ten kinds of room usages are considered for the case study which are illustrated in Figure 3. The total number of rooms is twelve. A room is composed of cells. A room usage is allowed to have multiple room shapes. In this study, some complex room shapes are given for some room usages to test the performance of the proposed method. In addition to the shapes illustrated in Figure 3(a), we consider their rotated shapes in every 90 degrees and mirror ones. Therefore, up to seven transformed shapes are given in addition to their original shape. Furthermore, adjacency relationships that affect the flow line and environmental condition are defined between different room usages and boundary conditions. Figure 3(b) illustrates the given adjacency relationships for this study. We may consider more constraints for actual use, but in this article we limit the basic ones because our main purpose is to demonstrate the enumeration.



(a) Room shapes. *(b) Adjacency relationships between room usages.*
Figure 3. Rooms used in this study.

The floor planning problem for a single floor plan is categorized as a tiling problem. That is, we place a piece of each room shape on the configuration space satisfying the adjacency relationships without any voids or overlapping. Finally, the floor plan enumeration problem is to output all floor plans that are the solutions of the floor planning problem.

3.2. ZDD REPRESENTATION FOR FLOOR PLANS

A node of a search tree such as a binary tree and a ZDD represents a room usage assigned to a corresponding cell. Figure 4 illustrates upper nodes of a search tree. These nodes correspond to the cell marked with 0 in Figure 1. While a general ZDD deals with Boolean functions, our problem needs multiple values for room usages. That is, more than two node levels are needed to express the assignment of a room usage to a cell if the number of room usages is more than two. The number of total levels through all cells is $|U| \times |C|$. Let n_{cu} denote the 0-1 variable that represents whether the room usage of a cell c is u or not. If $n_{cu} = 1$, the usage of cell c is u, otherwise the usage is not u. From the constraint described in Section 3.1, $\sum_{u \in U} n_{cu} = 1$.

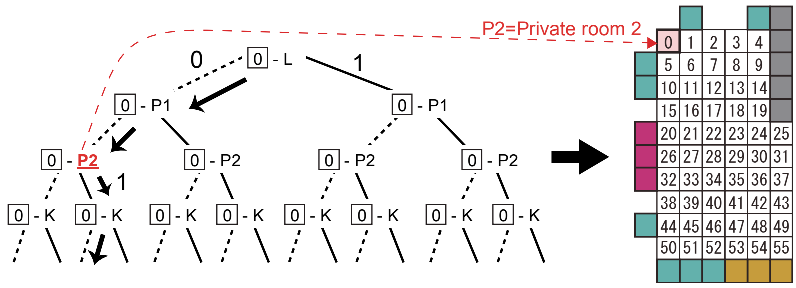


Figure 4. Node representation of floor plans over a search tree. (For easy understanding, the search tree is expressed as a binary tree but not as ZDD.)

3.3. ENUMEATION WITH FRONTIER METHOD

Each room usage is assigned to a cell from the upper left to the lower right in breath first order. If the assignment does not satisfy constraints before a floor plan is completed, the succeeding assignment is terminated and another series of assignment starts to be searched. In order to check whether an assignment satisfies the constraints or not, information of assigned cells is required. We call the set of assigned cells that are adjacent with unassigned cells as frontier. Frontier method requires only the information of cells in the frontier for checking constraints (see Figure 5).

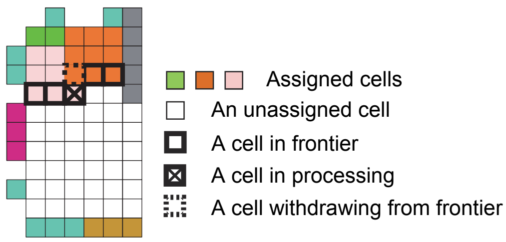


Figure 5. Distinction of cells around frontier.

Information of a cell in the frontier are as follows: (1) kind of assigned room usage, (2) the area of each assigned usage, (3) possible shapes of each room inferred from cells assigned and (4) sufficiency of adjacency relationships. That information is stored in nodes of frontier cells. Nodes storing the same information can be merged to simplify the search tree. Constraints are checked before and after determining to assign each room usage to a cell. If a constraint is not satisfied, the corresponding node is connected to 0-terminal node of a ZDD. That is, the ZDD records that the assignment is infeasible. Table 1 lists the flow of the assignment.

Table 1. Flow of assigning a room usage to a cell.

<table>
<tr><th>Phase</th><th colspan="2">Condition</th><th>Action</th></tr>
<tr><td rowspan="2">(1) Before the assignment</td><td colspan="2">A room usage u is no assigned to a cell c and c is not assigned any room usage.</td><td rowspan="2">Corresponding node is connected to 0-terminal node.</td></tr>
<tr><td colspan="2">u is assigned to c and c has been already assigned the other room usage.</td></tr>
<tr><td rowspan="2">(2) During the assignment</td><td colspan="2">u is not assigned to c.</td><td>Information of frontier nodes on possible shapes for u is updated.</td></tr>
<tr><td colspan="2">u is not assigned to c.</td><td>Information of frontier nodes on possible shapes, area and adjacency conditions for u are updated.</td></tr>
<tr><td rowspan="3">(3) After the assignment</td><td colspan="2">Regardless of whether u has been assigned to c or not, no possible room shape for u remains.</td><td rowspan="3">Corresponding node is connected to 0-terminal node.</td></tr>
<tr><td rowspan="2">u has been assigned to c.</td><td>The area of the room exceeds the maximal area for u.</td></tr>
<tr><td>c and the other cell with a different room usage faces the same window.</td></tr>
<tr><td rowspan="2">(4) When cell c leaves the frontier</td><td rowspan="2">The room usage u of cell c is not included in the cells on frontier.</td><td>Area of the room for u is less than its minimal area.</td><td rowspan="2">Corresponding node is connected to 0-terminal node.</td></tr>
<tr><td>An adjacency relationship for u has not been satisfied.</td></tr>
</table>

3.4. SEARCH QUERIES FOR EXTRACTING SPECIFIED FLOOR PLANS

By combining ZDD operations described in Section 2, we can extract the floor plans that satisfy a search condition directly from all enumerated solutions stored in a ZDD. We prepare three search queries: (1) extract floor plans for which the room usage of a given cell matches the one specified, (2) extract floor plans for which two specified rooms are adjacent with each other, (3) extract floor plans for which the shape of a specified room matches the one specified. Due to limitations of space, we explain only search query (1) used in Section 4. Assigning a room usage to a cell means that the value of the corresponding node is set to 1. Let F be the ZDD that preserves all floor plans and $G(c,u)$ be the other ZDD representing that a room usage u is assigned to a cell c. Search query (1) is formulated as $F' \leftarrow F \setminus G(c,u) \times G(c,u)$ where F' is the ZDD extracted. "$F \setminus G(c,u)$ " returns a set eliminating $G(c,u)$ from the subset of F that contains $G(c,u)$. Then, "$\times\ G(c,u)$" add $G(c,u)$ to each floor plan in the subset extracted just before.

4. Experiments

We apply the method to the problem described in Section 3.1. We use a desktop PC, whose CPU is Intel Core i7-3820, memory size is 64GB, OS is windows 7 Professional and compiler is Visual C++ 2010.

Table 2 lists computational time with different kinds of search trees. While both ZDD based trees can find all solutions satisfying constraints, the binary tree based method cannot find them because the number of nodes has increased to a large extent. However, even the shortest computation time of ZDD based method took about six hours. Comparing the computation time of both ZDDs results, the time of node merging is much longer than that of the ZDD without node merging and their numbers of nodes do not differ so much. The node merging operation is not effective for the problem.

Table 2. Results of numerical experiments.

Sort of search trees	Num. of nodes	Num. of solutions	Computation time
ZDD with node merging	464468021	1806	About 26 hours
ZDD without node merging	491140760	1806	About 6 hours
Binary tree	1.44×10^{17}	-	-

Figure 6 shows a part of the enumerated floor plans. All enumerated plans can be downloaded from the URL(http://p.tl/UibG). The location of private rooms 1 and 2 sometimes changes. On the other hand, those of kitchen and living room are relatively stable. Figure 7 shows the ratio of the number of assignments of each cell for each room usage against the maximal number of the room usage assignments to a cell. If a room usage never been assigned to a cell , the cell shows its cell index. Almost all room usages are located near in the original position of UR Hanabatake apartment because the number of rooms, the sort of room usages, the adjacency relationship and the boundary condition used for this case study were decided with reference to the apartment. However, the existence of another floor plan easy to ensure diversity than the original one is suggested. For example, while the entrance of the actual plan is located in the left side of the alcove, the most assigned positions shown in Figure 7 are below the alcove. If the entrance was located there below the alcove, the diversity of floor plans would increase more without changing the position of the entrance. Thus, the proposed method can provide useful information for the capacity analysis not only the degree of diversity but also the changeability of each room usage.

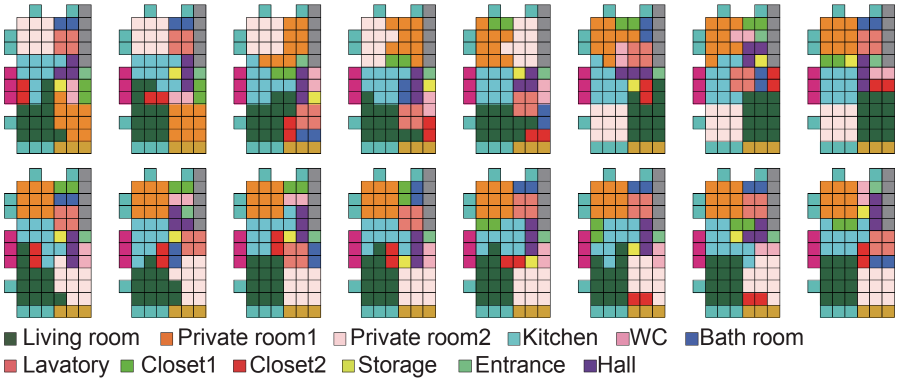


Figure 6. A part of the enumerated floor plans.

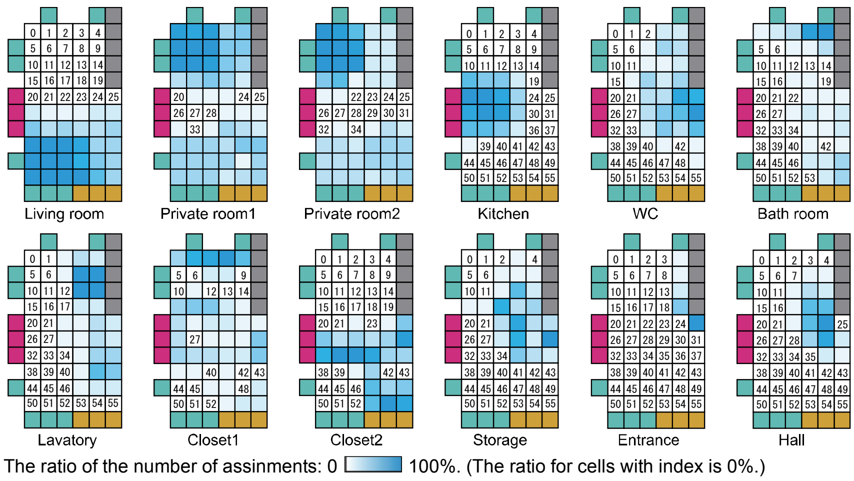


Figure 7. Ratio of the number of assignments of each cell for each room usage against the maximum number of the assignments to a cell.

5. Conclusion

The variability and diversity of infill are important indices for open buildings. While the evaluation of variability has been already proposed, the evaluation of diversity has not been studied yet. In this article, we proposed the novel method to enumerate floor plans satisfying constraints by using a ZDD and a frontier method. Through the numerical experiment, we confirmed that the enumeration is possible within practical amount of time even when the number of rooms whose shape is not limited to be rectangular is more than ten. In addition, the enumeration approach clarifies the position where each room usage tends to be located. This information might be useful because if a room usage tends to be located in the same position, the room usage might be fixed while preserving the total diversity. Though the variability of Open Building sometimes conflicts with economic issues, enumeration approach has the possibility to relax this conflict.

Future works are as follows: (1) improvement of the efficiency of enumeration and checking the information of frontier nodes to be merged, (2) more realistic constraints and (3) experiments with a larger configuration space.

Acknowledgements

This study is supported by members of Minato Discrete Structure Manipulation System Project, Assoc. Prof. Horiyama in Saitama Univ. and a Grant-in-Aid for Scientific Research (A) (No. 25240004).

References

Avis, D. and Fukuda, K.: 1996, Reverse Search for Enumeration, *Discrete Applied Math*, **6**, 21-46.

Habraken, N. J.: 1972, *Supports: An Alternative to Mass Housing*, Praeger.

Hokkaido University, Graduate School of Information Science and Technology: 2012, "Minato Discrete Structure Manipulation System Project web pages". Available from: Open Source Repository <http://www-erato.ist.hokudai.ac.jp/index.php?language=en> (accessed 1, December 2012).

Lobos, D. and Donath, D.: 2010, The problem of space layout in architecture: A survey and reflections, *Arquiteturarevista*, **6**, 136-161.

Manning, P.: 1964, An approach to the Optimum Layout of Single-storey Buildings, *The Architect's Journal Information Library*, **17**, 1373-1380.

Minato, S.: 1993, Zero-suppressed BDDs for Set Manipulation in Combinatorial Problems, *Proceedings of the 30th International Design Automation Conference, Dallas*, 272-277.

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

Nakano, S.: 2002, Enumerating Floorplans with n Rooms, *IEICE transactions on fundamentals of electronics, communications and computer sciences*, **E85-A**(7), 1746-1750.

Saitoh, T., Kawahara, J., Yoshinaka, R., Suzuki, H. and Minato, S.: 2011, Path Enumeration Algorithms Using ZDD and Their Performance Evaluations, *IPSJ SIG Notes*, **2011-AL-134**(17), 1-6.