

Layout Algorithm for an EC-based Room Layout Planning Support System

Inoue, Makoto
Graduate School of Design, Kyushu University

Takagi, Hideyuki
Faculty of Design, Kyushu University

<https://hdl.handle.net/2324/1808899>

出版情報 : IEEE Conference on Soft Computing in Industrial Applications (SMCia). 2008, pp.165-170, 2008-06-25. IEEE

バージョン :

権利関係 :



Layout Algorithm for an EC-based Room Layout Planning Support System

Makoto INOUE

Graduate School of Design, Kyushu University
4-9-1, Shiobaru, Minami-ku, Fukuoka, 815-8540 JAPAN
<http://inoue.s201.xrea.com/>

Hideyuki TAKAGI

Faculty of Design, Kyushu University
4-9-1, Shiobaru, Minami-ku, Fukuoka, 815-8540 JAPAN
<http://www.design.kyushu-u.ac.jp/~takagi/>

Abstract—We introduce a new layout algorithm, along with its underlying growth rules, and propose its use in generating room layout plans. We apply it to room layout planning, investigate its characteristics by observing the generated room layout plans, and discuss our results. We also introduce a framework for a room layout planning support system consisting of the layout algorithm, evolutionary multi-objective optimization (EMO), interactive evolutionary computation (IEC), and a graphical user interface. The proposed layout algorithm, inspired by cellular automata, grows rooms from the initial subspace seed coordinates according to a set of growth rules and fills out the entire provided space with the grown subspaces. The room layout planning support system uses the layout algorithm to generate rooms which fill a specified residential space and evolves the plans using the multiple objectives of by way of EMO and human visual inspection through IEC until a satisfactory layout plan is obtained.

I. INTRODUCTION

As an expert's experience, skills, and domain knowledge are necessary for layout planning, a demand exists for a layout support system to be developed for more efficient planning. There are several objectives in layout planning, and sometimes tradeoffs must be made amongst these objectives. Furthermore, the objectives are sometimes qualitative in addition to being quantitative. This is why layout planning is difficult, and planners are required to have both experience in layout planning and *KANSEI*, or artistic sense.

Layout planning is a broad field which can include including IC layout for electronic devices, the editorial design of publications and web pages, architectural design, landscaping, and regional and urban planning. Computational models for space layout are necessary to make layout planning support systems, and various models have been proposed so far. However, these models have left some problems unsolved; for example, some models cannot remove gap spaces from the layout and/or cannot embed a designer's experience, domain knowledge or even *KANSEI* into the support systems.

Several algorithmic-based room layout planners have been proposed. One such proposed system for architectural planning optimization used a combination of cells [8]. The target of the layout planning in this case was detached houses. The remaining limitations with the model were the lack of controllability of the gaps between rooms and room shapes.

In [2], the target application was hospital architecture, and a simple approach was used where rooms were of certain fixed

shapes. The lack of freedom in specifying the room shapes limited the number of generated room layout plans, however.

If the goal of optimization is only to optimize the size of each assigned rooms while maintaining a fixed topological room relationship, the task becomes simple [1]. This represents only one part of room layout planning, however, and the approach is unable to generate several varieties of room layout plans.

Another approach is to recursively divide the space into twos and increase the number of rooms this way [5]. The shapes of obtained rooms are consequentially, however, limited to being only rectangular and square.

The *bin packing* problem aims to minimize gap spaces and maximize the number of items which can be fit into a space by optimizing their directions and positions. However, unlike layout planning, it does not change the actual sizes and shapes of the items under consideration.

One of the simplest ways to specify the shapes and locations of subspaces in a layout plan is to specify each of the rectangular or square subspaces by a coordinate pair corresponding to its diagonal corners. If the pairs of coordinates are given randomly, many layout plans will include overlapped subspaces as shown in Fig. 1, and it is difficult to minimize gap spaces which crop up between these subspaces. An evolutionary computation (EC)-based layout planning system typically handles overlapped subspaces in layout plans, especially in the case of room layout planning, by considering their genes as lethal and removing them. Another limitation of the approach is that the subspace shapes are again limited to being either rectangular or square. With these drawbacks, this approach does not seem suitable for room layout planning.

The objective of this paper is to propose a new layout algorithm, discuss its usefulness, and present a framework for an EC-based layout planning support system that embeds planner experience, domain knowledge, and *KANSEI* into room layout plans. The algorithm and framework presented are targeted at architectural room layout planning applications.

Following the overview of layout planning research that we have presented in this section, we propose a layout algorithm that generates room layout plans in sections II and III, and discuss the generated plans in section IV. Finally, we present the entire framework for an EC-based room layout planning support system consisting of the proposed layout

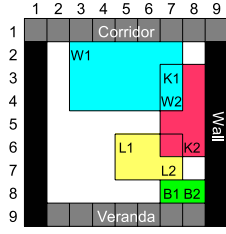


Fig. 1. An sample room layout produced using a two-coordinate approach. The shape and position of a living room, for example, is specified by $L1$ and $L2$. The number of overlapped rooms and gap spaces are apt to increase.

algorithm, evolutionary multi-objective optimization (EMO), interactive evolutionary computation (IEC) [7], and graphical user interface (GUI). Using these evolutionary techniques, our experience and domain knowledge can be embedded into designing room layout plans.

II. ALGORITHMIC LAYOUT PLANNING

A. Proposal of Layout Algorithm

We propose a layout algorithm that satisfies the following two requirements: (1) size, shape and location of rooms within the residential space should be changeable, and (2) gap spaces between rooms in layout plans should be minimized. These specifications are difficult to meet with the algorithms discussed in section I.

The outline of our algorithm is as follows. We create seeds for the subspaces located within the area where we plan to create a layout. A chromosome is defined as the set of coordinates for these initial subspace seeds and a subspace is defined as the set of points close to a subspace seed. A space layout is planned by controlling two factors; one is the definition of distance and the rules by which the distance is used to determine growth, and the other is the chromosomes, i.e. the coordinates of the subspace seeds, which we describe in detail in sections II-B and II-C, respectively.

The below four steps outline the concrete procedures that comprise our layout algorithm. Assume that the entire layout space is discretized so that it consists of a collection of cells, and coordinates and distances can be expressed in terms of these cells. We will call the procedure of incrementally forming subspaces from their seeds based on distance as *growth* in this paper.

- Step (1) Subspace seeds, which are the starting points from which subspaces for grown, are randomly or manually placed on cells within the whole space consisting of $m \times n$ cells.
- Step (2) Each subspace grows into its neighboring cells based on the below-mentioned growth rules.
- Step (3) The growth of each subspace is controlled by the growth rules such that growth is hindered by obstacles, e.g. walls, pillars, and other rooms.
- Step (4) Steps (2) and (3) are iterated until the subspaces occupy the entire layout space and the growth of all the subspaces has completed.

When the layout algorithm is used as a room layout planning support system, the *whole* layout area corresponds to the residential area and the subspaces correspond to rooms. Fig. 2 illustrates an example of how the growth process is used to generate room layout plans from subspaces.

Our layout algorithm can be understood by its similarity to Voronoi diagrams. Since the shapes of the subspaces generated by our algorithm are dominated by the initial coordinates of their subspace seeds, they have similar characteristics to discrete Voronoi diagrams and rectangular Voronoi diagrams [6]. Although we describe the incremental growth of subspaces in the above concrete procedure, this does not contradict the nature of Voronoi diagrams. When we make a subspace layout using Voronoi diagrams directly instead of the previously mentioned incremental procedure, subspace shapes can be controlled by using different distance definitions, such as Euclidian distance, Manhattan distance and Karlsruhe distance [6]. Thus boundary lines between two subspaces need not be limited to the horizontal and vertical, but may for example also take on angles of 45 degrees or 135 degrees when Manhattan distance is used. Alternatively, the boundaries become a combination of radial lines and arc lines when Karlsruhe distance is used.

B. Growth Rules

Under the framework of our layout algorithm, it is through the application of the growth rules that we form subspaces from subspace seed coordinates. The incremental growth rule essential involves increasing the size of the subspace one cell at a time and limiting growth when one subspace expands against another. As there are several different methods of defining and measuring distance from seed coordinates, many different layouts can be produced by varying how distance is used in the growth rules.

A variety of different layouts can also be generated by changing the method of growth. The following represent some ways in which this can be accomplished: expand a subspace into n of four directions ($n = 1, 2, 3$, or 4); expand a subspace into neighboring oblique cells or prohibit said expansion; expand a subspace into n of four directions ($n = 2, 3$, or 4) either at the same time, or sequentially; expand all subspaces at the same speed, or vary speed by subspace; expand a subspace at the same speed in all its directions, or make speed dependent on the n direction. There are also several variations to rules the rules we can use when stopping the growth of a subspace which has struck another subspace. When one part of a subspace's edge strikes other subspace, there are two choices: (a) the growth of the struck edge segment stops or (b) only the growth of the struck cell stops. This difference in the stop rules results in different sizes of gaps and subspace shapes in the produced layout plans, such as whether gaps arise in the layout or whether the subspaces are rectangular in shape or deformed.

Because the growth rules affect the subspace shapes and gap spaces, it is necessary to adjust or change the growth rules according to the layout tasks and layout planning specifications.

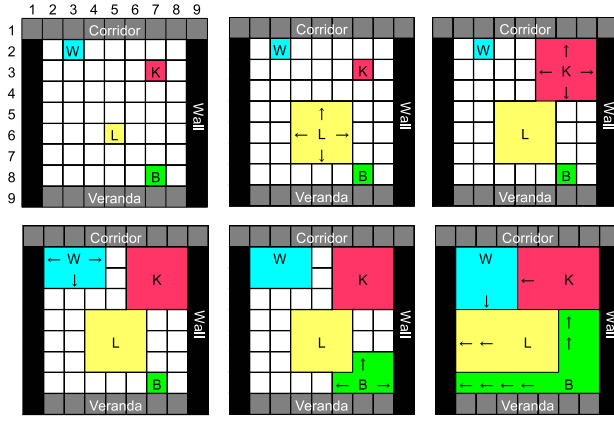


Fig. 2. Room layout process based on our layout algorithm. L, K, W, and B are the initial seed positions of a living room, a kitchen, a water area, and a bedroom, respectively. Correspondence of a sample gene coding and its phenotype, a room layout. The gene coding of [(6,5), (3,7), (2,3), (8,7)] represents the coordinates of the initial seeds of the rooms.

For example, if the real-world specifications of a room layout task specify that room shapes should be rectangle, square, *L*-shape, cross-shape, or another shape, with some set order of priority, then we need to update the growth rules based on these specifications.

The growth rules used in the room layout experiment of section III are:

- IF there is no obstacle or other room neighboring the cells of a room,
- THEN the room expands its area in all four directions,
- ELSE the room does not expand in the direction where there are obstacles or other rooms.
- IF a room expands in two adjacent directions (e.g., North and West or South and East),
- THEN the diagonal cell which is now touching on two sides with the expanded room is included in that room as the combination of the cells which were formed by the expansion.

Other layout tasks, such as garden planning, may demand different boundary lines between subspaces, such as curves, and different shapes of subspaces, such as non-rectangular shapes. In those cases, we would adjust the growth rules to meet the specifications.

C. Gene Coding of the Layout Plan

As our evolutionary room layout plan support system in section V evolves room layouts using EMO and IEC, so the layout plans must be expressed in terms of chromosomes. We use the initial coordinates of room seeds as genes. For example, [(6,5), (3,7), (2,3), (8,7)] expresses the initial coordinates of four subspaces, i.e. rooms, in a chromosome. See the upper left of Fig. 2.

A restriction of the initialization is that no two room seeds may have the same coordinates; otherwise, the room which possesses a duplicate seed will not appear in the layout. This case will be explained in the next section.

Here, we describe how the room layout support system can be used to model a residential space. The residential space is surrounded by obstacles such as walls and pillars, and a room layout plan is restricted to be inside of this area. The proposed layout algorithm grows each room from its initial seed coordinate until the growth rules stop its growth when it strikes either an obstacle outside the residential space or other rooms. Room layout in this paper is defined by the obtained rooms and their spatial relations when the layout algorithm completes.

Note that our layout algorithm always creates the same room layout when the same chromosome is given. However, it is also possible for different chromosomes to generate the same room layout. We discuss this further in the next section III-B.

III. EXPERIMENTAL GENERATION OF ROOM LAYOUT PLANS AND OBSERVATIONS

A. Experimental Conditions

There are many fields where space layout planning is used. In this study, we focus on building architecture and apply our layout algorithm to design room layout plans for the residential unit of a condominium as shown in Fig. 3 in this section.

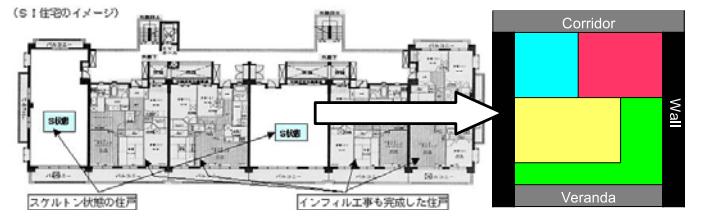


Fig. 3. Example of real room layout. The image on the left is referenced from Japan Ministry of Land, Infrastructure and Transport, *Skeleton-Infill Residence*.

The objective in this section is to generate several room layout plans using our proposed layout algorithm and observe their characteristics. From this, we expect to be able to determine the strengths, weaknesses and peculiarities of the algorithm.

We use a 1 m^2 cell as the basic unit for the room layout of a $7\text{ m} \times 7\text{ m}$ residential unit in this experiment. The basic unit size depends on layout planning applications. The residential unit consists of four rooms: a living room (L), a kitchen (K), a water area (W) and a bedroom (B). The surroundings of the residential area include a common corridor at the north side (upper side), walls at the east and west sides, and a veranda at the south side. An entrance door is assigned to the common corridor side. The room type at this entrance door side influences the evaluation of room layout plans from the perspective of room accessibility.

B. Observing Generated Room Layout Plans

Here, we evaluate our proposed layout algorithm through observing obtained room layout plans.

Several varieties of room shape can be generated by changing the growth rules. Room shapes generated by the proposed

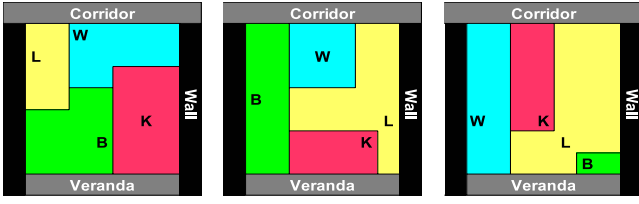


Fig. 4. The shapes of several generated rooms.

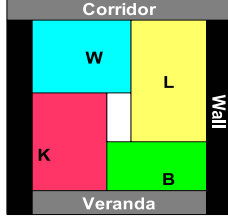


Fig. 5. A gap space is generated between rooms.

layout algorithm are basically rectangle or square, and room shapes whose corners are cut-off in part can be generated by changing the stop conditions of the growth rules. Likewise for e.g. *L*-shape, convex-shape, cross-shape, and other irregularities (see Fig. 4).

The number of room layout plans generated by our layout algorithm must be more than are provided by conventional approaches. Usually, conventional room layout plans are based on rectangular or square shaped rooms, and this reduced variety in room shape limits the number of room layout plans. Although it desirable to compare the number of layouts produced by several approaches numerically, it seems difficult to calculate the total number of combinations possible with the arbitrarily shaped rooms obtained using our proposed methods, so we must postpone this quantitative evaluation to a future work.

The gap space size between rooms can be controlled by the growth rules. It is possible to make the gap zero if we do not restrict the shapes of rooms. The biggest gaps are made when the stop condition of the growth rules is the strictest: *IF there is an obstacle or other room neighboring the cells outside the room, THEN the entire edge of the room facing the direction of the obstacle or other room does not expand*. Even if in this case, the gap size is only the width of one basic unit, i.e. 1m wide in this experiment (see Fig. 5).

The layout algorithm does not generate two overlapped rooms when two identical pairs of initial coordinate room seeds are expressed in a chromosome. When we apply the algorithm to a room layout planning support system, we must avoid overlapped initial coordinates in order to generate all the rooms of the planning specification; otherwise the case illustrated in Fig. 6 may occur.

Same room layout plans can be generated by different combinations of seed coordinates. Basically, the border of two subspaces passes through the middle point between two seed coordinates of the subspaces. Although the border location

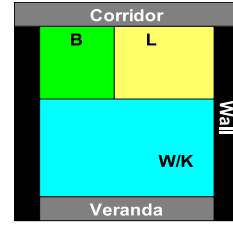


Fig. 6. N rooms cannot be generated when some of n initial coordinates of room seeds are the same.

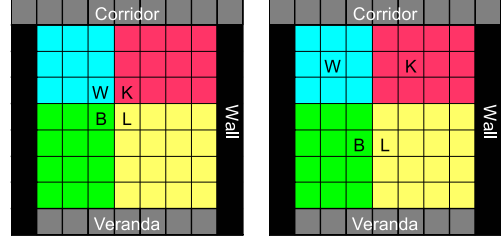


Fig. 7. Different initial coordinates of room seeds may result the same room layout plan.

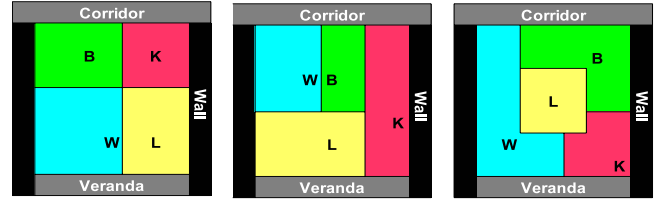


Fig. 8. Room layout plans of simply combined rectangular rooms (left and middle) and that with an *island* room (right).

depends on growth rules, there are different pairs of the seed coordinates that generate same subspace border. Fig. 7 is one of its examples.

Sometimes, the long part of an *L*-shaped room can become too thin (see the bedroom in Fig. 2), which is not suitable for a residential room from the architectural planning point of view. It is therefore necessary to introduce a stop condition into the growth rules to prevent the generation of such shapes.

The layout algorithm and growth rules can generate room layout plans that include *L*-shaped rooms such as at the left of Fig. 4. However, there are also combinations of two *L*-shaped rooms that cannot be generated when the sample growth rules of section III-A are used.

Further examples of other room layout plans that can be generated by the algorithm and growth rules are illustrated by the simply combined rectangular rooms shown at the left and middle of Fig. 8, and the island room surrounded by other rooms and not touching the exterior walls as shown at the right of Fig. 8.

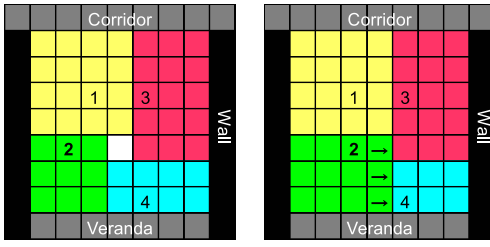


Fig. 9. A slight difference in the initial coordinates for the room seeds results in different room layout plans.

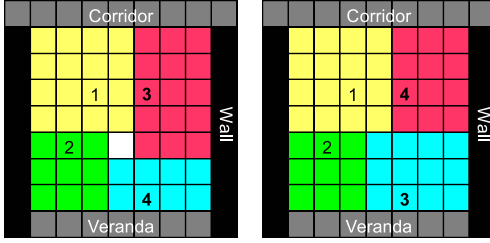


Fig. 10. A difference in growth order affects room layout plans. Both room layout plans are obtained by iteration of the growth in numeric order.

IV. DISCUSSIONS

A. Controlling Room Layout Plans

Our layout algorithm does not require information regarding room size or direction, but only the initial coordinates of the room seeds. This results in a fewer number of genes, i.e. a shorter chromosome length. A fewer number of optimization parameters helps preferred room layouts be found more quickly.

There are three factors for controlling room layout plans: the initial coordinates of room seeds, the growth order, and the growth rules.

The positions of the subspaces depend on the initial coordinates of the room seeds, and the relationship amongst the initial coordinates of the room seeds is the dominant factor for deciding room sizes.

The two different room layout plans shown in Fig. 9 are obtained from the same initial room seed coordinates, except for a slight difference in room seed No.2.

The order of room growth also influences the generated room layout plans. Since each room grows by discrete increments, i.e. by a basic unit of $(1m \times 1m)$ in our experiment, if there are an odd number of cells between two room seeds, the border between the two rooms is decided by the growth order of the rooms; since it cannot be placed in the middle of a cell, the midpoint between the two seeds will end up in one room or the other. The unit size influences the generated room layout plans as well. Fig. 10 is an example illustrating how different room growth orders result in different room layout plans.

The growth rules mainly control the shapes of rooms by enabling the generation of not only rectangular or square shapes but also L -shaped or convex-shaped rooms. If the edge

of a room is controlled to stop its growth when any one part of the edge strikes an obstacle or other room, all the rooms will become rectangular or square in shape and gap spaces may be generated between the rooms. On the other hand, if all the cells of the room edge grow into the adjacent cells in their direction of growth except for the cell that struck an obstacle or other room, the gap space becomes smaller or is eliminated entirely and the room shapes gain more variety.

By changing the stop rules and/or growth speed, we can control the balance of the constraint on room shape variety and permissible range for gap spaces.

B. Effectiveness of the Proposed Layout Algorithm

In this section, we discuss the number of ways in which cells can be combined, i.e. the number of possible layout plans, and then discuss the usefulness and advantages provided by the proposed layout algorithm.

How many room layout plans can be generated? When each of the 49 ($=7 \times 7$) cells takes 1 of 4 rooms, we calculate a total of 4^{49} possible combinations. This is the number of, for example, color patterns which can be generated when one of four colors is used for each cell. This is not, however, the correct answer for room layout plans, since cells that are not connected cannot form the same single room. The number of room layout plans must therefore be fewer than 4^{49} .

To make four rooms, Nakano's binary-division method divides a residential space into two rooms first; second, it divides either of the two rooms into two and makes three rooms; lastly, it divides either of three rooms into two and makes four rooms [5]. All room shapes made by this method become either rectangular or square. Total number of room layout plans becomes 49,344 ($= ({}^6C_3 \times 2 + ({}^6C_2 \times {}^6C_1) \times 18 + ({}^6C_1 \times {}^6C_1 \times {}^6C_1) \times 2 - ({}^6C_1 \times {}^6C_1)) \times 4!$).

In our layout algorithm's case, there are ${}_{49}P_4$ (5,085,024= $49 \times 48 \times 47 \times 46$) patterns corresponding to the combination of initial subspace seed coordinates. The number of the maximum possible room layout patterns would then become ${}_{49}P_4 \times 4! = 122,040,576$, if we take into account the growth order amongst the four rooms. However, differing initial subspace seed coordinates may nevertheless result in the same room layout plan, as mentioned, so the real number of generated room layout plans must be fewer than this number. Further analysis and experimental confirmation are necessary.

In comparison to a random search, our algorithm maintains the variety of subspace shapes while at the same time optimizing to limit the search space. Table I is a comparison of the three approaches in term of the numbers of the possible combinations.

TABLE I
THE NUMBER OF LAYOUT PLANS.

random model	4^{49}
division model [5]	49,344
our algorithm	fewer than 122,040,576

V. ROOM LAYOUT PLANNING SUPPORT SYSTEM

The next step is to make a room layout planning support system in which the layout algorithm and growth rules can be embedding the layout algorithm and growth rules into EMO and IEC framework.

There are many terms on which an architectural room layout may be evaluated. They include legality of each subspace's area; its proportions, i.e. the aspect ratios and forms of each subspace; convenience and comfortableness for residents, i.e. resident's circulation; architectural laws, e.g. requirements or restrictions on daylighting; and architectural costs such as total length of walls and ducts. Additionally, the subjective evaluations of e.g. the architects and residents, is an important factor in room layout planning. This means that room layout planning is a multi-objective optimization task, and a framework is therefore required to combine multi-objective optimization with optimization based on subjective evaluation.

We combined EMO and IEC [1], [4] and developed an EC-based room layout planning support system recently. Fig. 11 shows the framework of an EC-based room layout planning support system. We are going to report its evaluation at other opportunity.

The EC-based room layout planning support system works as follows:

- 1) Chromosomes are initialized randomly or set with prepared coordinates manually.
- 2) Our proposed layout algorithm generates room layout plans from the chromosomes.
- 3) EMO evaluates the multi-objective fitness of the generated room layout plans quantitatively.
- 4) IEC lets a layout planner visually evaluate the generated room layout plans and input his or her subjective evaluations.
- 5) EMO calculates Pareto ranks using the obtained quantitative and subjective fitness values.
- 6) Parent individuals are selected using fitness based on the Pareto ranks.
- 7) The selected parent individuals evolve using evolution-

ary operations and generate offspring for next generation.

- 8) Go to 2 until a desired room layout plan is obtained.

One of the important research themes is how we can use users' evaluations, qualitative factors or human favorites effectively for EMO in the future. Embedding a function for learning human preference to assist EMO would be another research topic.

VI. CONCLUSION

We proposed a layout algorithm and evaluated it by applying it to room layout planning. Through observing the generated room layout plans, we have shown that it is flexible at changing the size, shape, and location of rooms and is capable of generating a variety of room layout plans.

Parameters of the growth rules which should be controlled in future studies include the growth priority of rooms, handling of gap spaces between rooms and growth speed. In order to obtain preferred room shapes, we have to understand the relationship between these parameters. We also outlined a framework for an EC-based room layout planning support system which we will use in the next step of our research. Other future extensions of this research include layout planning in 3-dimensional (3-D) spaces for tasks such as whole building design or curved partitioning for tasks such as garden planning.

Acknowledgments

This work was supported in part by the 21st Century COE program and the project to develop *innovative seeds* of Japan Science and Technology Agency.

REFERENCES

- [1] A. M. Brintrup, H. Takagi, A. T. and J. J. Ramsden, "Evaluation of sequential, multi-objective, and parallel interactive genetic algorithms for multi-objective optimization problems," J. of Biological Physics and Chemistry, vol.6, pp.137–146 (2006).
- [2] T. Inoue, Y. Kohama and T. Takada, "Study on Architectural Space Planning by Optimality Method," Japan Society of Mechanical Engineers, OPTIS2000, vol.2000, no.4, pp.281–285 (Oct., 2000) (*in Japanese*).
- [3] M. Inoue and H. Takagi, "Computational Geometry Model for Evolutionary Spatial Planning," IEICE General Conference, D-8-29 (March, 2008) (*in Japanese*).
- [4] R. Kamalian, H. Takagi, and A. M. Agogino, "Optimized Design of MEMS by Evolutionary Multi-objective Optimization with Interactive Evolutionary Computation," Genetic and Evolutionary Computation (GECCO2004), Seattle, WA, USA, pp.1030–1041 (June, 2004).
- [5] Shin-ichi Nakano, "Enumerating Floorplans with n Rooms," IEICE Trans. on Fundamentals, vol.E85-A, no.7, pp.1746–1750, (2002).
- [6] A. Okabe, B. Boots, K. Sugihara, and S. N. Chiu, "Spatial Tessellations: Concepts and Applications of Voronoi Diagrams," John Wiley & Sons Inc., Chichester, (2000).
- [7] Hideyuki Takagi, "Interactive Evolutionary Computation: Fusion of the Capabilities of EC Optimization and Human Evaluation," Proceedings of the IEEE, vol.89, no.9, pp.1275–1296 (2001).
- [8] S. Tanigaki, A. Tani and Y. Yamabe, "Circulation and Shape Planning of Dwelling House by Multiple-Optimization System," Architectural Institute of Japan, 30th Symposium on Computer Technology of Information, Systems and Applications, pp.7–12 (Dec., 2007) (*in Japanese*).

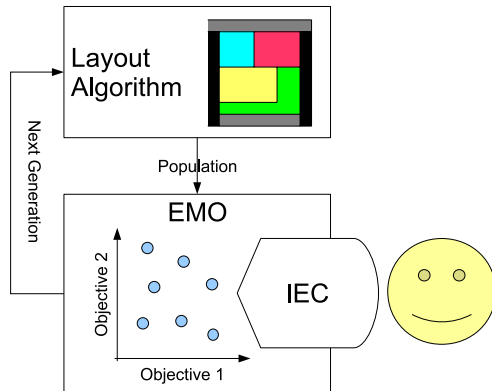


Fig. 11. The framework of an EC-based room layout planning support system consisting of EMO, IEC, and a graphical user interface.