An Interior Layout Support System with Interactive Evolutionary Computation using Evaluation Agents

Yusuke Bamba and Junji Kotani and Masafumi Hagiwara
Department of Information and Computer Science, Keio University
3-14-1 Hiyoshi, Kohoku-ku, Yokohama 223-8522, Japan
{banba,kotani,hagiwara}@soft.ics.keio.ac.jp

Abstract – *In this paper, we propose an interior layout support system using interactive evolutionary computation. The conventional system is only concerned with layout of a room, and is not concerned with a usability of layouts. In the proposed system, layouts are evaluated using newly proposed evaluation agents, which behave as virtual residents. As a result, a usefulness value can be calculated as an estimated usability for each layout. The proposed system has the interface to show layouts in a 3-D virtual space. The system evolutionarily generates layouts according to two factors: user's evaluation and usefulness values. Evaluation agents move and work virtually in layouts. As the result of these activities, the system can estimate feature values such as a quantity of noise, a degree of passage jam and so on for each layout. In addition, the system calculates a usefulness value of each layout from these feature values. Since the user can observe not only layouts but also behaviors of evaluation agents, he/she can imagine the usage of layouts easily. Results of experiments show that the layouts that meet the user's demand are obtained.*

Keywords: genetic algorithm, interior layout, evaluation agents.

1 INTRODUCTION

Recently, many people are interested in interior layout. For example in business, office layout influences the operating efficiency. On the other hand, in an everyday life, the concern of the interior layout as self-assertion increases. And many books on interior have been published. We focus on an interior layout in this paper [1] [2] [3].

Because of development of computers, many layout support systems have been developed. However, layout is left to a user's knowledge in general. So, a layout support system that reduces user's burden and can get good layout is required.

There have been many researches adopting evolutionary computation. For example, some interior layout systems employed artificial intelligence. These systems can reduce human error for layout. However, to get good layout, many rules are required [4] [5] [6]. Other interior layout systems adopted Conceptual Space. Conceptual Space arranges idea about the layout, which a user has. These systems can reflect intention of a user in a layout. However, it is difficult to reflect user's preference in a layout [7].

On the other hand, to treat user's particular preference and sensitivity, usage of interactive evolutionary computation is effective. This method is based on genetic algorithm and can reflect user's evaluation to fitness of population. This method is useful for layout reflecting user's preference and sensitivity [8]. In an interior layout, it is difficult to imagine an actual usage of the layout. So, it is important to develop interior layout system.

Therefore, we employ an interactive evolutionary computation. In order to evaluate a usability of layouts, we propose the idea of evaluation agents.

2 INTERIOR LAYOUT SUPPORT SYSTEM

2.1 The outline of the proposed system

The proposed system has an interface that shows layout in 3D virtual space. And this system employs interactive evolutionary computation based on evaluation agents and user's subjective evaluation. Fig.1 shows the processing flow of the proposed system. First, a user inputs some data about the room for layout. For example, the size and form of the room, the kind and the number of furniture, the number of employees, and so on. Second, the proposed system generates initial individuals. Then evaluation agents evaluate the room. For example, they evaluate a quantity of noise, a degree of passage jam. The usefulness value and user's subjectivity evaluation are the degree of adaptation of a layout. Third, based on the degree of adaptation, the proposed system performs genetic operation (selection, cross over, mutation). And then, next individuals are generated. These cycles last until a user is satisfied.

At the time of system starting, the proposed system shows to user in the state looking down at a layout. In addition, a user can walk virtually the inside of a layout at any time.

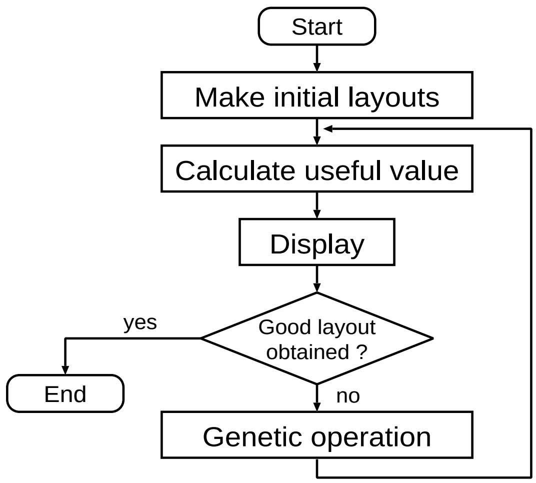


Figure 1: Outline of the proposed system.

2.2 Arrangement of furniture

In this section, we explain arrangement of furniture. In the followings, coding of gene and decoding of gene are explained.

2.2.1 Coding of gene

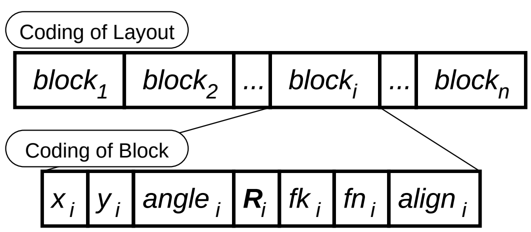


Figure 2: Coding.

Fig.2 shows coding of an individual. One code consists of some parts named “block”. A block consists of

- x_i,y_i: relative coordinates over the room of a block
- $angle_i$: rotation angle of the block
- R_i: rule of arrangement of a block
- fk_i: the kind of furniture, which constitutes a block
- fn_i: the number of furniture, which constitutes a block
- $align_i$: adjustment of arrange for block i or not

R_i is represented by double ended queue. It consists of 8 rules.

Table 1 shows these 8 rules. These 8 rules named “subrules” decide the position of furniture to the standard furniture placed before. R_i of the first generation is determined at random.

Table 1: subrules

rules	The position to standard furniture	The angle to standard furniture
s_1	forward	same direction
s_2	forward	reverse direction
s_3	backward	same direction
s_4	backward	reverse direction
s_5	right	same direction
s_6	left	same direction
s_7	right	rotates on the left 90 degrees.
s_8	left	rotates on the right 90 degrees.

$align_i$ is a parameter for adjustment of arrange for block.

$$align_i = \begin{cases} 1 & \text{: do adjustment} \\ 0 & \text{: otherwise} \end{cases} \tag{1}$$

2.2.2 Decoding of gene

Decoding is performed sequentially from the head of a gene in a block. Arrangement block i to the room is performed as follows.

1. place the block i in the room so that a center is set to (x_i,y_i).
2. rotate $angle_i$
3. if $align_i$ is 1, block i is arranged so that it touches the nearest block may be touched.
4. if block i overlap a wall or the other blocks, only furniture not overlapping are arranged.

2.3 Evaluation Agents

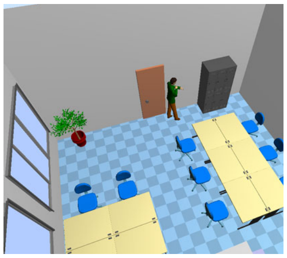
Evaluation agents move on lattice-like nodes and work virtually based on the predefined Markov process. A A* algorithm is used for course determination [9]. A user can determine the number of evaluation agents. The agents act and evaluate the usefulness value of layout in the background of the system. In addition, a user can observe the agents' actions.

Fig.3 shows examples of evaluation agent's action. First, an agent appears from the door (a). Second, he moves to the locker and works (b). Then, he moves to his desk and works (c). After that, he moves to the copier or the bookrack and works (d). After a certain time, several agents appear and discuss (e). Last, he gets out the room. These actions continue about 60 seconds.

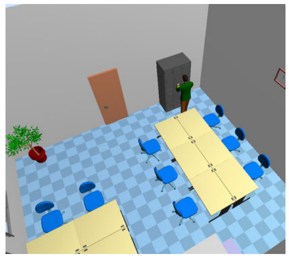
The usefulness value is represented by 5 elements of the usefulness value of the room.

$$U = U_r(a_s U_s + a_d U_d + a_f U_f + a_n U_n) \tag{2}$$

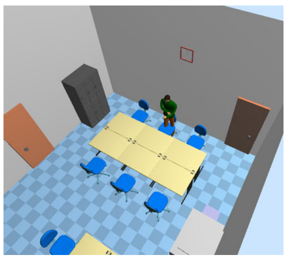
Hereinafter, 5 elements of the usefulness of the room will be described.



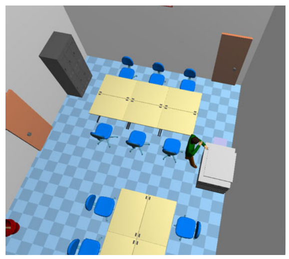
(a) agent appears



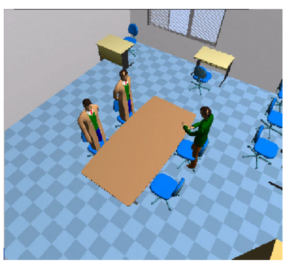
(b) works at the locker



(c) works at the desk



(d) works at the copier



(e) discussion

Figure 3: Examples of agents' actions

- U_r : Can all furniture be reached or not?
- U_s : Is the surrounding quiet or not?
- U_d : Is the narrow part crowded with agents or not?
- U_f : Can agents talk each other in the room or not?
- U_n : How long is the total move distance?

a_s, a_d, a_f, a_n are the constants. A user can change these constant numbers by the use of the room.

$$U_r = \begin{cases} 0 & : \text{furniture which agents can't reach exists} \\ 1 & : \text{agents can reach all funriniture} \end{cases} \tag{3}$$

U_s is an average value of the noise per unit time. Only when agents need silence, U_s is computed.

$$U_s = \frac{\sum_{t=0}^{t_{max}} \sum_{i=1}^{A_{max}} \sum_{j=1, j \neq i}^{A_{max}} O_i(t) noise_{ij}(t)}{1 + t_{max} A_{max} \sum_{t=0}^{t_{max}} \sum_{i=1}^{A_{max}} O_i(t)} \tag{4}$$

t_{max}	:	maximum step
A_{max}	:	the number of agents
$noise_{ij}(t)$	:	agent i hears from agent j's noise at time t

$$O_i(t) = \begin{cases} 1 & : \text{agent } i \text{ needs silent when } t \\ 0 & : \text{otherwise} \end{cases} \tag{5}$$

One node connects neighboring 8 nodes. Evaluation agents can move eight directions. U_d means non-crowdness and is expressed as

$$U_d = \frac{1}{1 + \sum_{i=1}^{X_{max}} \sum_{j=1}^{Y_{max}} traffic(i,j)} \tag{6}$$

$$traffic(i,j) = \begin{cases} 1 & : \text{When node(i,j) connect} \\ & \ \ \text{less than 2 nodes and} \\ & \ \ \text{the amount of passings} \\ & \ \ \text{exceeds fixed quantity} \\ 0 & : \text{otherwise} \end{cases} \tag{7}$$

U_f is expressed with the following formulas.

$$U_f = \frac{\sum_{i=1}^{A_{max}} \sum_{j=1, j \neq i}^{A_{max}} F(i,j)}{A_{max}(A_{max}-1)/2} \tag{8}$$

$$F(i,j) = \begin{cases} 0 & , \quad i \neq j \text{and ajent i is not} \\ & \quad \ \text{in the distance which} \\ & \quad \ \text{can talk with agent j} \\ 1 & , \quad \text{otherwise} \end{cases} \tag{9}$$

When an agent's number is one, U_f is not computed.
U_n is expressed with the following formulas.

$$U_n = \frac{\sum_{i=1}^{A_{max}} \sum_{j=1}^{P_{max}^i} d_i^j}{\sum_{i=1}^{A_{max}} P_{max}^i} \tag{10}$$

P_{max}^i	:	agent i's number of times of movement
d_i^j	:	agent i's i-th time move distance

2.4 Genetic Algorithm

In the proposed system, one individual corresponds to one layout, and some individuals form one generation. Fitness is represented as weighting summation of user's evaluation and usefulness value.

2.4.1 Fitness

The fitness is calculated as follows.

$$f = b_1 U + b_2 E \tag{11}$$

U	:	usefulness value
E	:	user's evaluation
b_1, b_2	:	weighting factor

2.4.2 Genetic Operation

Genetic operations are as follows.

- Selection : Roulette selection, Elitist preserving selection.
- Cross Over : Exchange two blocks.
- Mutation : Randomize one block.

3 EXPERIMENTS

18 subjects used the proposed system and verified the validity of this system. Similar experiments were conducted without evaluation agents. Table2 shows genetic algorithm parameters.

Table 2: Parameters in the experiments

cross over rate	0.8
mutation rate	0.1
the number of individuals	10
end conditions	10 generations

3.1 Experimental Results

We prepared data about four kinds of rooms in advance. All subjects chose one data from them and evaluated rooms that the proposed system generated in five stages (very good, good, even, bad, very bad).

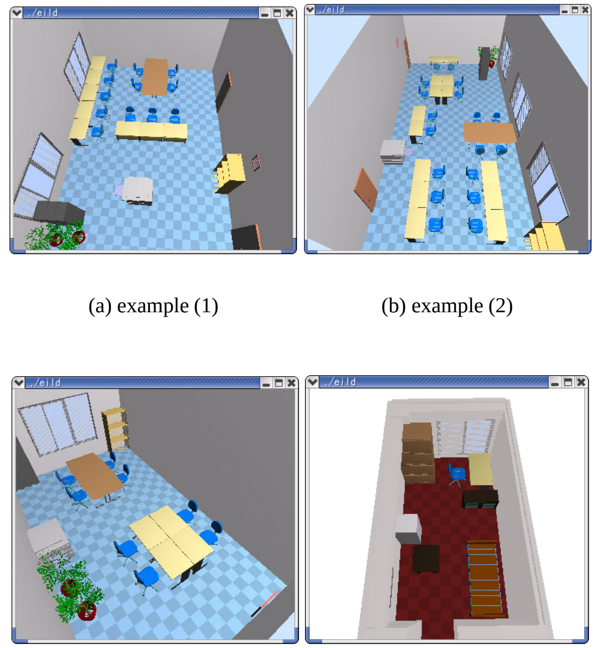
Examples of the acquired layout are shown in Fig.4.

Table 3: Average and distribution of the degree of satisfaction

	Average	Variance
Proposed system	4.23	0.86
Without evaluation agents	3.23	1.19

Fig.5 shows the distribution of the degree of satisfaction. Table 3 shows an average and distribution of the degree of satisfaction. These results show the validity of evaluation agents.

These are qualitative opinions from subjects.



(a) example (1) (b) example (2)

(c) example (3) (d) example (4)

Figure 4: Examples of an obtained

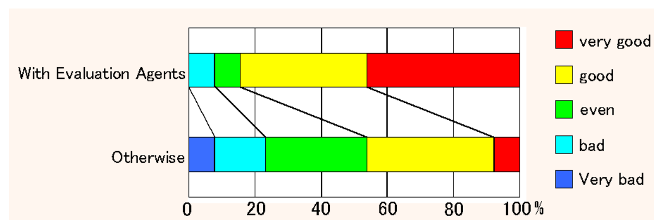


Figure 5: Degree of satisfaction

- It was easy to layout. Because we had only to push the button.
- By observing agents' activity, problems in a room which is difficult to could be found.
- Without evaluationary agents, furniture might been arranged in front of a door.
- Without evaluationary agents, furniture have been arranged where agents could't reach.
- Without evaluationary agents, direction of a desk scatters and lacks in a feeling of unification.

Also, these opinions show the validity of evaluation agents.

Fig.6 shows the transition of average fitness. This result shows the validity of the evolutionary computation.

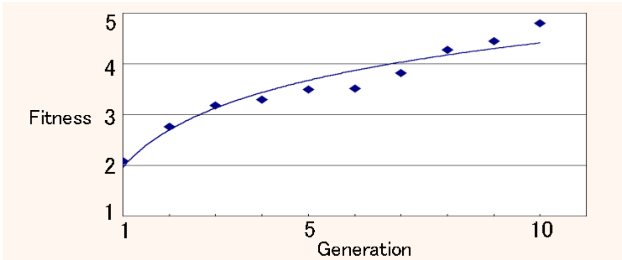


Figure 6: The transition of average fitness

Table 4: Average time and the number of average generations until subjects acquire a satisfactory layout

Average time [m]	The number of average generations average generations
22.7	7

Table 4 shows average time and the number of average generations until subjects acquire a satisfactory layout

4 CONCLUSION

We proposed an interior layout support system with interactive evolutionary computation using evaluation agents in this paper. The proposed system has the interface to show layouts in a 3-D virtual space. The system evolutionarily generates layouts according to user's evaluation and usefulness value. Evaluation agents move and work and, evaluate usefulness value of layouts.

From experimental results, effectiveness of the proposed system is shown.

Acknowledgment

This work is supported in part by a Grant in Aid for the 21st century Center Of Excellence for Optical and Electronic Device Technology for Access Network from the Ministry of Education, Culture, Sport, Science, and Technology in Japan.

References

[1] tsuneaki nishigashira:"A meeting revolution is the source of energy," Nikkei buisiness, 2003.

[2] Comfortable office creation http://www.netricoh.com/contents/antenna/office/data/0018.html,

[3] An office layout and collaboration http://www.periander.co.jp/Column1/Columun1.html,

[4] Hon Wai Chun, Edmund Ming-Kit Lai:"Intelligent critic system for architectural design," *IEEE Transaction on Knowledge and Data Engineering*, vol.9, no.4, pp.625-639, 1997.

[5] Yasuyuki Nishioka, Shinichi Nakasuga, Kouichi Hori:"An Interactive Design Support System Concerning with Manufacturing : An Application to An Aired Pump Design," Journal of Japanese Society for Artificial Intelligence, vol.13, no.2, pp.300-311, 1998.

[6] Yumi Iwasaki:"Qualitative reasoning and the science of design," *IEEE Expert*, vol.12, no.3, pp.2-4, 1997.

[7] Kaname Kakihara, Toshihiro Nagata, Takenao Ohkawa, Norihisa Komoda:"Character Design System by Concept Space Formational Design," Transactions of Information Processing Society of Japan, Vol.37, No.6, pp.1227-1234, 1996.

[8] Hiroaki Kitano: "Genetic Algorithm 1,2,3,"Sangyoto-sho, 1993, 1995, 1997.

[9] Hart, P. E., N. J. Nilsson, and B. Raphael: "A Formal Basis for the Heuristic Determination of Minimum Cost Paths in Graphs," IEEE Trans. on Systems Science and Cybernetics, Vol. SSC-4, No. 2, pp 100-107, July 1968.