

FUNCTION-SPACE ASSIGNMENT AND MOVEMENT SIMULATION MODEL FOR BUILDING RENOVATION

Ren-Jye DZENG, Wei-Chih WANG, Fan-Yi HSIAO

Department of Civil Engineering, National Chiao-Tung University, 1001 Ta-Hsieu Rd., Hsinchu, 30050, Taiwan

Received 24 Jul 2012; accepted 09 Nov 2012

Abstract. Building renovation is an effective way to revive the use of a building, the use efficiency of which is primarily determined by its layout. However, in architectural practice, architects and building owners renovate buildings based on their personal subjective perceptions of how occupants use the building instead of systematically analyzing their use behaviors. This study proposes a model, called the Function-space Assignment and MOVement Simulation (FAMOS) model, which integrates radio frequency identification (RFID), fast messy genetic algorithms (fmGA), and movement simulation techniques to solve the function-space assignment problem. The RFID equipment is specifically used to track the occupants' movement data in a building, the fmGA is employed to identify the optimal result of function assignment, and the movement simulation technique is adopted to verify the result and support the decision-making of function-space assignment. This study presents a real case study to demonstrate the use of FAMOS and compare its assignments with those generated by a renovation architect. The objective function showed that FAMOS's version had a 14.80% higher objective value than the architect's version. The experiment also showed that FAMOS helped the architect find the best assignment or improve their assignment based on desired objectives such as preferred space size, minimized movement distance, or removal of corridor congestion.

Keywords: function-space assignment, RFID, optimization, fmGA, genetic algorithms, movement simulation.

Introduction

Assigning appropriate functions to building spaces is one of the most important factors in determining the use performance of an existing building. Using an educational building as an example, there are several types of occupants, and they move around the spaces in the building based on the activities which they have been formally assigned or in which they are personally interested. The function-space layout (such as classrooms, administration offices, laboratories, library, meeting room and so on) affects how occupants move and the distance they must cover to participate in their activities in the building. Kalay (2004) noted that function assignment only works in limited areas of architectural design, primarily because of the lack of quantifiable data. Instead of relying on the trial and error attempts of the facility administrator to find the optimum layout, mathematical optimization algorithms can be used to design the layout of hospitals, factory assembly lines, and construction sites once quantifiable data on the occupants' movements are available. Another prevailing problem is that the administrator may lack the means to quantitatively verify their proposed function-assignment to assess its performance. This verification may be difficult for new buildings because the intended occupants are unknown or no data

on their behaviors are available. However, the solution to this problem may be feasible for an existing building because occupants are known and it is possible to collect data on how they use the building.

Instead of relying on subjective experience or opinions, a researcher can actually monitor the activities of building occupants manually or by using location-tracking technologies to collect quantitative data on occupants' movement. These technologies include Wi-Fi-based systems, infrared systems, ultrasound, scene analysis, and RFID (Tesoriero *et al.* 2010). This study investigates a location technology capable of tracking occupants' movements between multiple partitioned indoor spaces in a multi-floor building using corridors that could be exposed to sunlight. Based on these conditions, Wi-Fi, infrared, ultrasound, and scene analysis technologies are unsuitable for movement tracking. Lionel *et al.* (2004) noted that an active RFID system is a viable and economical option for indoor location sensing. Therefore, this study uses RFID as the location technology.

However, the optimized function assignment may be unsuitable for occupants. This is because human crowds exhibit highly complex behavior driven by individual decisions based on individual goals, environmental obstacles, and the surrounding crowd (Rahul *et al.* 2009).

To quantitatively verify the function-space assignment, one may set up experiments in which sampled occupants mimic the use of the facilities in an ad hoc environment. However, this approach is costly, and thus, is seldom used. Another approach is to use a computer simulation program. Simulations have previously been used to monitor building performance (Hong 2000) in areas such as renovation scheduling simulation (Lee 2012), and occupant evacuation planning (Yang *et al.* 2005), etc. In addition, the majority of renovation research in existing buildings has focused on the multi attribute assessment (Zavadskas *et al.* 2009), multiple criteria evaluation (Kaklauskas *et al.* 2005; Zavadskas, Antucheviciene 2007), and decision support system (Kaklauskas *et al.* 2008). Relatively little research simulates occupant movements in the function-space assignment of existing buildings. Dzung *et al.* (2012) proposed a model that simulated occupant movement in an educational building based on the assumption that occupants move to participate in designated activities or for private needs.

This study proposes a model, called the Function-space Assignment and MOvement Simulation (FAMOS) model, which attempts to optimize the function assignment by sensing occupant movement data, data-mining the function-space relationship, finding optimized function-space assignment using fmGA, and verifying its achievement on the objective function and other objectives not included in the function using simulation.

1. Facility layout modeling and problem solving techniques

Optimization techniques have been used in architecture primarily for solving problems of facility layout, structural design, and building performance (Choudhary *et al.* 2005). Facility layout optimization involves finding feasible topology and the dimensions of interrelated objects that meet all of the design requirements and maximizing design preferences (Liggett, Mitchell 1981). Previous research has developed several formulations for the optimization of facility plans. For discrete formulations, the QAP is the most commonly encountered in the literature.

QAP was first proposed by Koopmans and Beckman in 1957. The QAP is a well-known classical combinatorial optimization problem, which can be described below. A set of n distinct facilities are to be placed uniquely in m distinct spaces, where $m \geq n$ (Koopmans, Beckman 1975). The QAP can be described as the task of finding the minimum allocation costs as well as some constraints to be satisfied (Jo, Gero 1998). In the earlier studies of manufacturing planning, the objective of QAP can be to minimize the handling cost of total material, and contribute to the overall efficiency of operations. Since then, the QAP model have been widely applied to many different real situations, such as the planning of buildings in university campuses, arrangement of departments in hospitals, warehouse management and distribution strategies, minimization of the total wire length in electronic

circuits, ordering of correlated data in magnetic tapes and others (Ramkumar *et al.* 2009). The QAP model was adopted in this research for the optimization of a building layout.

For the QAP, several types of problem-solving approaches have been proposed, such as exact, heuristics and meta-heuristics approaches. Examples of exact approaches are branch and bound (Solimanpur, Jafari 2008), cutting planes (Bazaraa, Sherali 1980), and dynamic programming (Christofides, Benavent 1989). Because exact solutions require large expenditures of time and money, it may not be worthwhile to search for the optimum solutions except in rare circumstances (Hahn, Jrarup 2000). For this reason, several heuristic and meta-heuristic approaches have been developed to search for sub-optimal solutions within a reasonable time limit.

There are two types of heuristic approaches, construction methods and improvement methods. Construction methods generate sub-optimal permutations from scratch by assigning functions to spaces one by one based on prioritized criteria. Examples are CORE-LAP (Lee, Moore 1967), ALDEP (Seehof, Evans 1967), COFAD (Tompkins, Reed 1967) and SHAPE (Hassan *et al.* 1986). Instead of starting from scratch, improvement methods begin with a feasible solution and try to systematically improve it by searching for other nearby solutions. The process is continued until no improvement can be found. Examples of this method are CRAFT (Armour, Buffa 1963), FRAT (Khalil 1973) and DISCON (Drezner 1987).

Before the end of the 1980s, most of the proposed heuristic approaches for combinatorial optimization problems were specific and dedicated to a given problem. Since that time, this paradigm has changed. More general methods have appeared, known as meta-heuristics (Loiola *et al.* 2007). Several of these methods are based on some type of simulation of a natural process studied within another field of knowledge. Recently, numerous researchers have developed meta-heuristics approaches for the QAP. Solimanpur *et al.* (2005) developed an ant algorithm for a sequence-dependent single row machine layout problem. Yeh (2006) adopted annealed neural networks and Hopfield neural networks to solve preferences in a hospital building layout problem. Liang and Chao (2008) developed the multi-searching technique of tabu algorithms to improve facilities layout performance through several previous examples, including a pre-cast yard, construction site and hospital. Cheung *et al.* (2002) developed the swap method of simple genetic algorithms to determine the least cost arrangement for a pre-cast yard layout. Jang *et al.* (2007) also employed simple genetic algorithms to optimize the layout of multi-floor construction material.

The simple genetic algorithm (sGA) is one of the meta-heuristic approaches. First developed by Holland (1975), sGA is an efficient and popular algorithm. Goldberg *et al.* (1989) subsequently developed the messy genetic algorithm (mGA) in 1989 to improve the sGA.

Several experiments have proven that the mGA is much better at solving permutation problems than the sGA. In 1993, Goldberg *et al.* (1993) developed the fast messy genetic algorithm (fmGA) to reduce the high memory consumption of operation processes. Over the years, mGAs and fmGA have been used successfully in water distribution system design (Halhal *et al.* 1999), the dispatching of ready mixed concrete trucks (Feng, Wu 2006), the design of fuzzy control systems (Hoffmann, Pfister 1996), solutions for clustering problems (Mohan 1993), and learning classifier systems (Lanzi 1999; Lanzi, Perrucci 1999; Skurikhin, Surkan 1996; Cheng *et al.* 2010). Because of its advantages, we used fmGA to search for the optimal solution in this research.

2. Research problem

This study proposes an FAMOS model, which integrates RFID, fmGA, and movement simulation techniques to solve the function-space assignment problem. Consider a scenario in which an existing public building with multiple-stories requires renovation, and the owner and the architect plan to adjust the functions of the spaces in the building during renovation based on their usage experience in order to accommodate the usability problems. The function provided by a space is fixed once the function-space assignment is finalized. There are several types of occupants, and they move around the spaces in the building based on the activities which they have been formally assigned or in which they are personally interested. Some activities occur periodically and some do not. Some activities require an occupant to participate at specific times, and some allow them to participate at will

and at their preferred times. Each occupant has an identification object so that their individual movement can be detected. The objective of the adjustment of space functionality is to minimize the total occupants' movement distance and the interference between certain spaces. Finally, the movement simulation technique is adopted to verify the result and support the decision-making of function-space assignment.

3. FAMOS model

The FAMOS consists of 4 modules, namely data collection, data analysis, optimization, and simulation as illustrated in Figure 1. This paper focuses on the description of function-space assignment optimization and simulation.

3.1. Data collection

For tracking the occupants' movements, we used an active RFID positioning approach, and the movement tracking devices included readers, tags, and gateways. As shown in Figure 1, a reader was installed at each tracked space, and occupants carrying RFID-tagged ID cards were detected once they moved in and out of the space. The gateway was responsible for acquiring the terminal data from the readers. The number of required gateways, usually operating at 2.4 GHz, depended on the accessibility of the wireless signals. The manufacturer's proprietary algorithm considered both the signal strength received and the time of arrival to determine the actual position of the tag. In the data screening, we also eliminated data outside of the tracking time range and noise, such as "passing" a space instead of "using" a space.

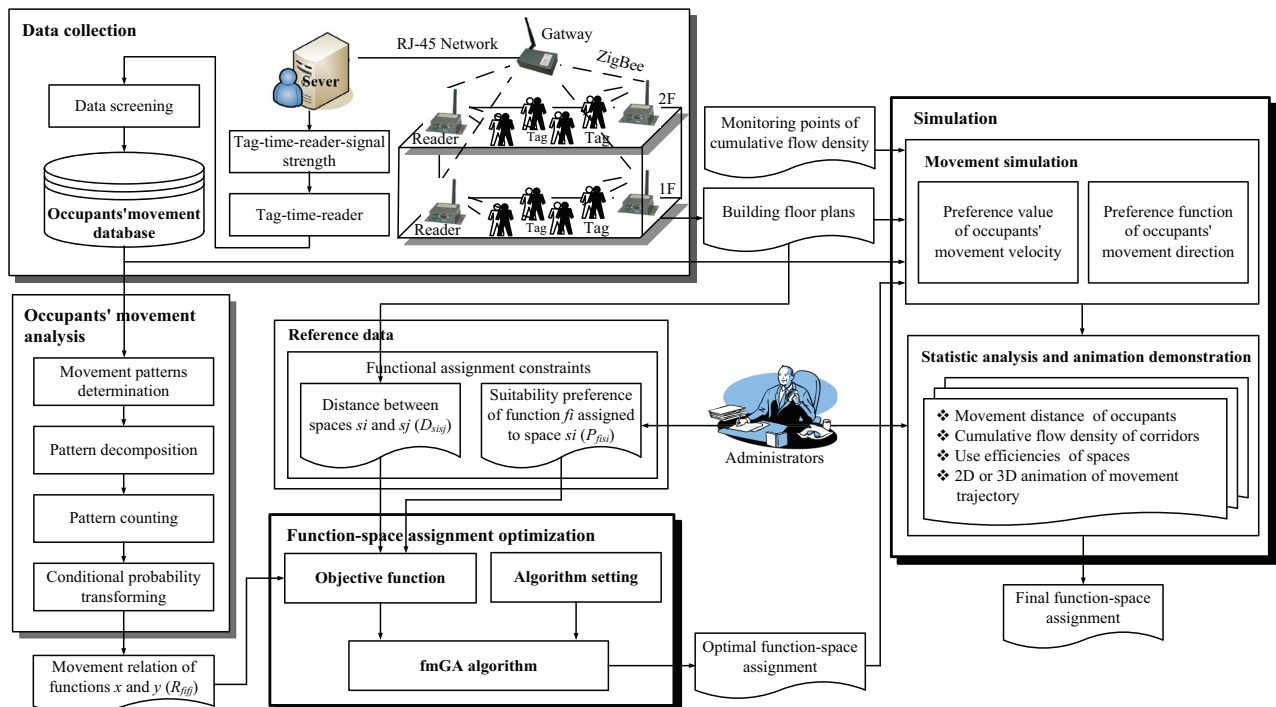


Fig. 1. Framework of FAMOS model

3.2. Occupants' movement analysis

The analysis of occupants' movement involves four steps, i.e., movement pattern determination, pattern decomposition, pattern counting, and conditional probability transforming, as shown in Figures 1 and 2. Movement pattern determination data mined an occupant's movement pattern between spaces on each day. One can set up a threshold of *maximum break time* for a movement between two spaces to be considered as part of a pattern. For example, one may set the *maximum break time* to be 30 minutes. Thus, if the time interval of the detections of a tag at two different spaces is smaller than 30 minutes, the corresponding occupant's movement between the two spaces will be considered as a movement pattern; otherwise, the uses of two spaces will be considered as independent usages.

Pattern decomposition breaks down the daily movement pattern of an occupant into pairs of spaces for later counting purposes. For example, as shown in the movement pattern table in Figure 2, Tag 120 has used space *a*, *a*, *b*, *c*, *c*, and *d* on 1 March 2011. The breakdown results in pairs of *aa*, *ab*, *bc*, *cc*, and *cd*.

Pattern counting calculates the occurrence of each pair of spaces in the breakdown result of the previous step in preparation for constructing the interactive preference table. As shown in the pattern counting table of Figure 2, each number represents the number of occurrences of the corresponding pair (from a column space to a row space). For example, the use pattern *aa* (the use of space *a* following space *a*) occurs 4 times, *ab* occurs 1 time, and so on.

The fourth step is to calculate the conditional probability of a row space given a column space, as shown in the movement relation table of Figure 2. For example, the likelihood of using space *b*, *c*, and *d* after using space *c* is 1/5 (0.2), 3/5 (0.6), and 1/5 (0.2), respectively.

3.3. Function-space assignment

The optimization module maximizes the objective function under defined constraints by finding the best assignment of functions to spaces. The objective function used in this research is based on the concept proposed by

Koopmans and Beckman (1975), which was applied by Jo and Gero (1998) in assigning functions to spaces of equal size in an office building. Yeh (2006) modified the objective function to enable the assignment of functions to spaces with different sizes in a hospital. We used the objective function proposed by Yeh (2006), but with two primary differences. First, the objective function of this research is based on the interactive preference derived from the tracking of real occupants' movements instead of subjective judgments. Secondly, while Yeh (2006) optimized the objective function using an annealed neural network, this research used a newly developed algorithm called the fast messy genetic algorithm.

For a building layout problem, facilities can be regarded as architectural functions; spaces can be regarded as spaces allocated for specific architectural functions (Yeh 2006). Eqn (1) is the objective function, which is a weighted average of two parts. First, $(X_{f_i s_i} \times P_{f_i s_i})$ represents the assessment of the suitability of a function f_i assigned to a space s_i . For example, a library (f_1) assigned to a large space (s_3) is more suitable than to a small space (s_5). Secondly, $(X_{f_i s_i} \times X_{f_j s_j} \times D_{s_i s_j} \times R_{f_i f_j})$ represents the assessment of a function assigned to a space from the perspective of the moving distance ($X_{f_i s_i} \times D_{s_i s_j}$) based on the movement relation ($X_{f_j s_j} \times R_{f_i f_j}$). For example, strong related functions assigned to neighboring spaces may have a higher assessment value than that to spaces at a distance:

$$\text{Max } O = W_1 \left(\frac{\sum_{f_i} \sum_{s_i} X_{f_i s_i} \times P_{f_i s_i}}{n} \right) + W_2 \left(\frac{\sum_{f_i} \sum_{s_i} \sum_{f_j} \sum_{s_j} X_{f_i s_i} \times X_{f_j s_j} \times D_{s_i s_j} \times R_{f_i f_j}}{n} \right) \quad (1)$$

Subjected to:

$$X_{f_j s_i} = 0 \text{ if } X_{f_i s_i} = 1 \text{ and } f_j \neq f_i,$$

$$X_{f_i s_j} = 0 \text{ if } X_{f_i s_i} = 1 \text{ and } s_j \neq s_i,$$

where O – objective function; $X_{f_i s_i}$ – permutation matrix

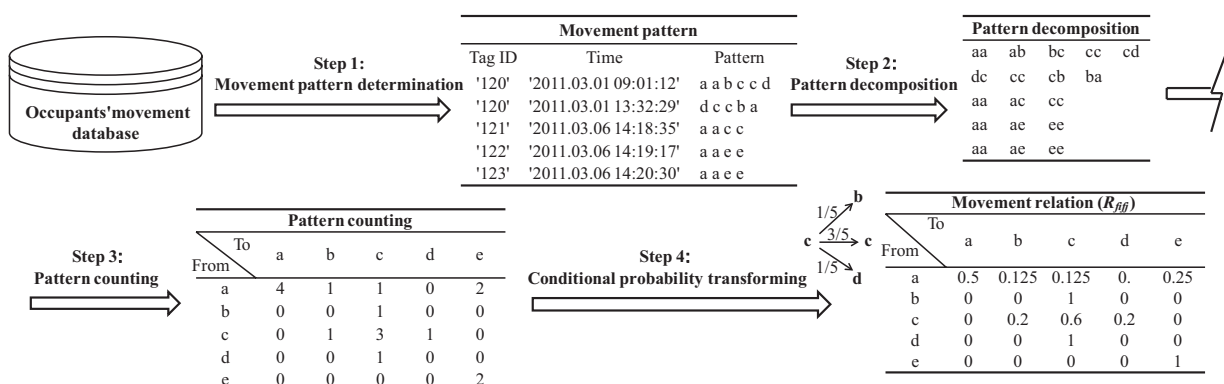


Fig. 2. Occupants' movement analysis

variable (i.e., the value is 1 if function f_i is assigned to space s_i , and is 0 if not assigned to s_i); P_{f_i, s_i} – suitability preference of function f_i assigned to space s_i ; D_{s_i, s_j} – distance between spaces s_i and s_j ; R_{f_i, f_j} – movement relation of functions f_i and f_j (i.e., the value is between 0 and 1, where 0 represents no sequential movement pattern exists between functions f_i and f_j , and 1 represents the use of f_i always followed by the use of f_j); n – total number of functions; W_1, W_2 – the weights between 0 and 1.

3.4. Simulation

Assessing and improving the quality of the indoor space for occupants is an important issue in the design of a public building (Lee *et al.* 2012). The goal of applying computer simulation to occupants' mobility is to reproduce and predict possible system behavior in hypothetical environments, thus facilitating the design of buildings and service location (Brambilla, Cattelani 2009). The FAMOS module allows decision makers to visualize occupant movements, verify the function assignment, or adjust the optimal assignment generated by the system for any reason. For example, one may want to adjust the building layout for qualitative reasons such as the consid-

eration of interference between functions (e.g., classroom should not be adjacent to a professor's office to avoid noise interference), scenic view, safety, and theft concerns. Other reasons may be quantitative, such as those that are difficult to express in an objective function. For example, avoiding movement flow congestion may be difficult to express in the objective function because there are theoretically an infinite number of congestion points in continuous corridors. The adjustment for new considerations may also come to mind when decision makers see simulation results.

As Figure 1 shows, the simulation module takes occupants' movement data, building floor plans, function assignment data as its input, and allows the user to set the monitoring point at a specific location to observe its cumulative movement flow density. The simulation then generates statistics such as the total movement distance for various types of occupants, usage density of spaces, and cumulative flow density of corridors. These statistics may be viewed from the perspective of spaces, activities, or times. In addition, it can also demonstrate the occupants' movement trajectory in 2D and 3D animations (Fig. 3).

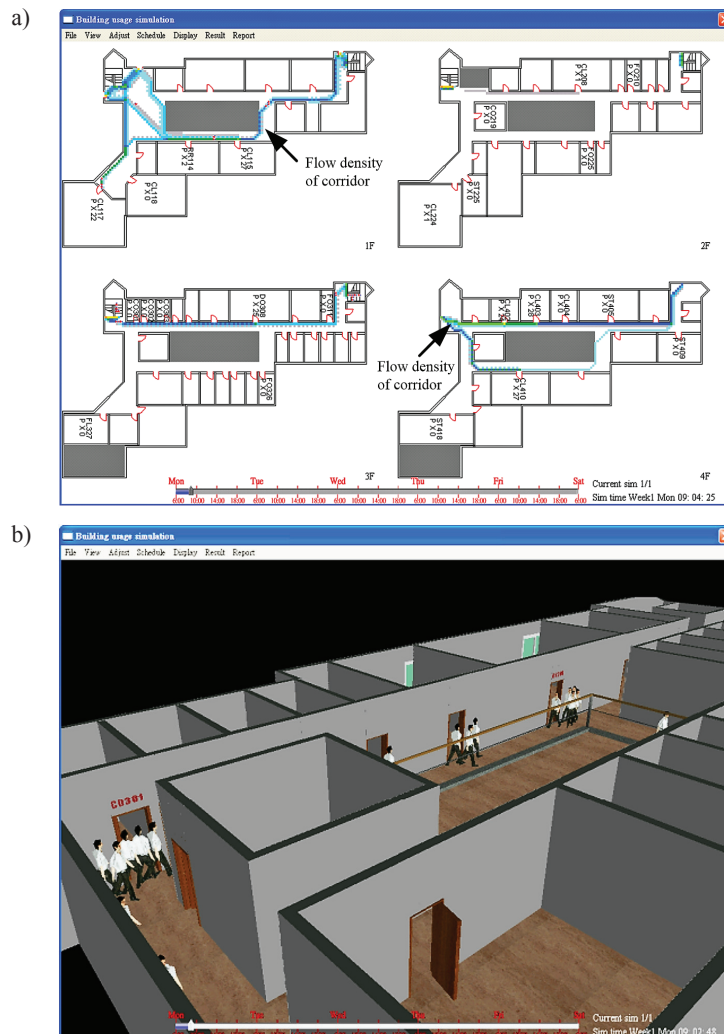


Fig. 3. Occupants' movement trajectory: (a) 2D animation, (b) 3D animation

This simulation module uses the cellular automata (CA) model (Yue *et al.* 2007) to simulate occupants' movement trajectories in a building. The CA model is a discrete dynamical system that simulates complex behaviors based on simple computational models. In addition, it requires the modeling of a discrete $W \times W$ cell grid in a two-dimensional area (i.e., the size of the space a human occupies at any point in time), the moving field (i.e., the possible cells that a human can move in each discrete time step during a simulation), and a preference function governing the preferred moving direction of a human.

The FAMOS is based on a $0.4 \times 0.4 \text{ m}^2$ cell grid size for a typical space occupied by a human in a dense crowd, as suggested by Burstedde *et al.* (2001). Each cell can be empty or occupied by no more than one occupant or obstacle. Figure 4 shows a 3×3 moving field, which represents the nine movement alternatives available for an occupant at the center position if only one cell is allowed for each step. The vision-conscious field defines the area where vision may affect movement. The vision-conscious field is larger than the moving field, and measures 3×5 (if not next to a boundary) or 2×5 (if next to a boundary), as shown by Figures 5(a) and 5(b), respectively.

The simulation divides continuous time into discrete time steps. In each time step, an occupant can remain in the original position or move one cell in the movement field. The occupant's movement direction depends on the

preference values of moving alternatives, as shown by the numbers (P_{ij}) in Figure 4. The occupant moves toward the cell with the highest preference value.

Eqn (2) is the preference function of movement direction (P_{ij}), which is a weighted average of four parameters: Straightness (S_{ij}), Category (C_{ij}), Forward (F_{ij}), and Empty (E_{ij}), each of which ranges between -1 and 1 . The sum of weights (i.e., w_S , w_C , w_F , and w_E) equals 1 . S_{ij} represents the proximity to the occupant's destination for a target cell ij . The cell closer to the destination has a higher S_{ij} . C_{ij} represents the proportion of empty cells and occupants homogeneous with the subject in the current direction of movement in the vision-conscious field around the target cell. F_{ij} indicates the proportion of empty cells unoccupied by other occupants ahead of the target cell in the vision-conscious field. E_{ij} calculates the proportion of empty cells unoccupied by obstacles ahead of the target cell in the vision-conscious field. Detail description regarding preference function of movement direction can be found in Dzung *et al.* (2012):

$$\text{Max } P_{ij} = w_S S_{ij} + w_C C_{ij} + w_F F_{ij} + w_E E_{ij}, \quad (2)$$

where: P_{ij} – preference function of movement direction; S_{ij} – straightness parameter; C_{ij} – category parameter; F_{ij} – forward parameter; E_{ij} – empty parameter; w_S , w_C , w_F , w_E – the weights between 0 and 1 .

4. Problem-solving process for function-space assignment optimization and movement simulation

The problem-solving process for function-space assignment optimization and movement simulation are illustrated in Figure 6. The fmGA is adopted to optimize function-space assignment and its output assignment is then sent to the movement simulation module to verify the achievement of the objective function as well as other objectives that were not included in the function. The process consists of 8 primary steps described as follows.

Step 1. Randomly generate a competitive template

The first step is to randomly generate a competitive template, which is a problem-specific, fixed-bit string that is randomly generated or found during the search process (Goldberg *et al.* 1993). The competitive template is used to make up for the missing genes in the latter process when chromosomes are under-specified. The fmGA process consists of inner and outer loops. Each inner loop is called an era and each outer loop is called an epoch. Thus, the execution of the maximum number of eras defined by era_max completes an epoch. The execution of the maximum number of epochs defined by epoch_max terminates the fmGA evolution process.

The inner loop consists of three phases (Goldberg *et al.* 1993): (1) the initialization phase – a population with sufficient chromosomes is created to contain all possible building blocks (BBs) of the order k , where BBs refer to partial solutions of a problem; (2) the primordial

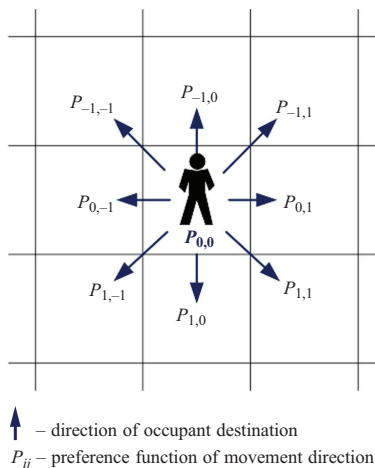


Fig. 4. Preference function of movement direction

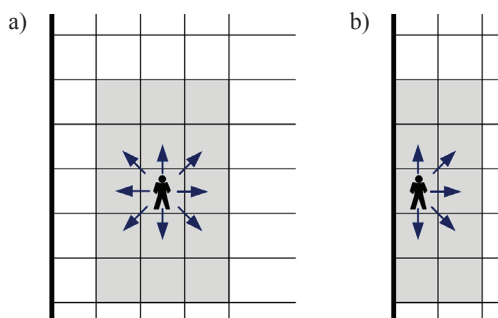


Fig. 5. Moving field vs. vision-conscious field without/with a boundary

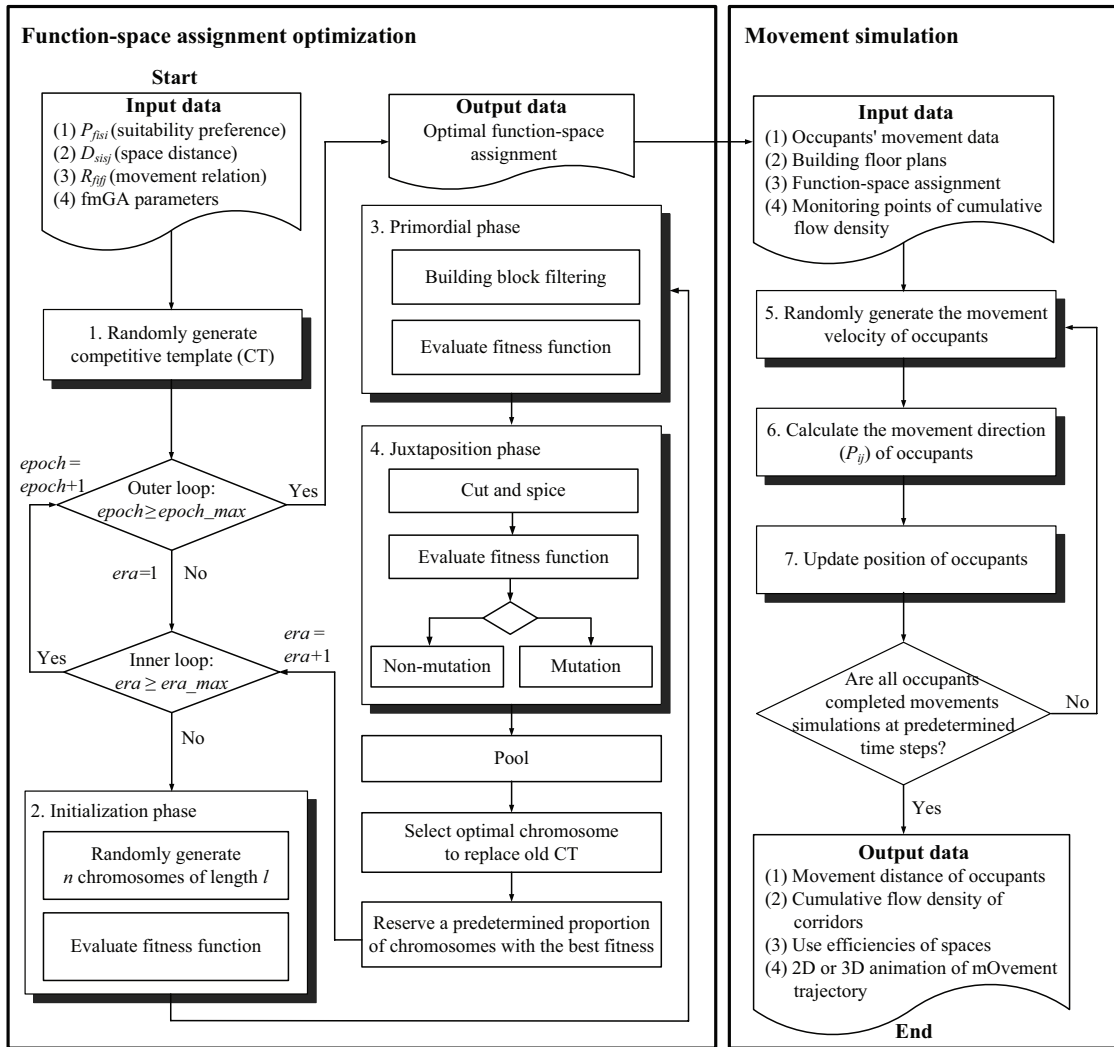


Fig. 6. Problem-solving process for function-space assignment optimization and movement simulation

phase – bad genes are filtered out to maintain only the chromosomes with good fitness; and (3) the juxtaposition phase – those good alleles (BBs) are rebuilt by cut-splice and mutation operations to form a high quality generation, which tends to generate an optimal solution.

Step 2. Initialization phase

To ensure a sufficient quantity of chromosomes, the population size of each era is determined by Eqn (3), as suggested by Goldberg *et al.* (1993). In addition, n chromosomes are randomly generated in this phase, so the fitness of each chromosome is evaluated based on the objective function, defined by Eqn (1):

$$n = \frac{\binom{l}{\lambda}}{\binom{l-k}{\lambda-k}} 2c(\alpha)\beta^2(m-1)2^k, \quad (3)$$

where: l – the problem length; k – the order of BBs; λ – a random value, generally set to be $l - k$, $k < \lambda \leq l$; $c(\alpha)$ – the square of a normal random deviate corresponding to

a tail-probability α ; β – the signal-to-noise ratio which is the ratio of the fitness deviation to the difference between two competing BBs; m – BBs coefficient.

Step 3. Primordial phase

There are two operations in the primordial phase, namely building-block filtering and threshold selection. Building-block filtering includes building-block selection and random gene deletion. The key to building-block filtering is to pump up enough copies of the good building blocks so that even after random deletion eliminates a number of copies, there remain one or more copies for subsequent processing (Goldberg 2002).

According to Goldberg *et al.* (1991), having enough good building blocks provides more good chromosomes for subsequent processing. Thus, they used a generic threshold mechanism, where the selection between two strings was only permitted if they shared a greater than expected number of genes in common, which restricts the competition between building blocks with little in common. However, the threshold was not needed in our

case because all of the chromosomes share the same set of genes (i.e., the same set of functions are assigned to a set of genes).

Step 4. Juxtaposition phase

The purpose of the juxtaposition phase is to change chromosomes, and it includes the cut-splice and mutation operations. At first the cut-splice operation is applied to a predetermined proportion (i.e., crossover rate P_c) of the chromosomes. After performing the cut-splice operation, the fitness value of a chromosome may be higher or lower than (or equal to) that of the competitive template in the previous era. The mutation is applied to a predetermined proportion (i.e., mutation rate P_m) of the chromosomes with lower (or equal) fitness values because they are less competitive solutions and vice versa.

The newly generated and existing chromosomes are stored in a pool, representing the population of the era. The best chromosome with the highest fitness value will be selected and replaces the competitive template if its fitness is higher. In addition, a predetermined proportion of the best population is kept and carried to the next era. Steps 2 to 4 are iterated for a predetermined number of times, which completes an epoch. Such a process is iterated until the fitness value of the best chromosome converges or the predetermined maximum number of epochs is reached.

Step 5. Randomly generate the movement velocity

The simulation module randomly generates a movement velocity for each occupant. Based on the Federal Highway Administration (2012), the pedestrian's average walking speed is approximately 1.22 m/s. Thus, the default setting is a random number between 1 and 4 cells/s (approximately 0.4 to 1.6 m/s).

Step 6. Calculate the movement direction

The module calculates the movement direction for each occupant. In a 3×3 moving field, the module chooses one cell with the highest preference value of movement direction (P_{ij}) as the target position for each time step. The module also uses the A* algorithm (Bourg, Seemann 2004) to calculate the occupants' shortest path from the original position to the final position.

Step 7. Update positions of occupants

The module randomly selects one occupant for the position update function within each time step. To avoid a conflict when any two or more occupants attempt to move to the same position (Keßel *et al.* 2002), FAMOS adopts a sequential update pattern at each time step during a simulation. The update rules are described as follows. The occupant in the cell with the highest value P_{ij} stays at the original position and does not move to another cell. If there are empty cells, the occupant moves to the cell with the highest value P_{ij} . If there are no empty cells, the occupant's moving direction is compared with

the occupant who has the highest value P_{ij} . Based on a 50% exchange probability (Blue, Adler 2001), two occupants mutually exchange positions if they have opposite movement directions. Steps 5 to 8 are repeated until all occupants have moved for the current time step.

5. Case and experiment

This section describes an experiment with a real case to prove the research concept. A simulation system, previously developed by the authors, is used to verify the result and support the decision making of function-space assignment to accommodate the desirable goals that are not included in the objective function. The following sections first introduce the case and the experiment with the setting of the system variables, then presents the results of function assignment and movement simulation.

5.1. Case

The case used in this experiment is the building of the Civil Engineering Department of National Chiao-Tung University in Hsinchu, Taiwan. The building is a 4-story courtyard building with a total floor area of 6 616 m². The 4 plans of this building (1F to 4F) are shown in Figure 7. There are 3 main entrances on the ground floor and two staircases. The spaces of the building include a garden, a library, an auditorium, an administration office, classrooms, laboratories, meeting rooms, seminar rooms, faculty offices, mechanical rooms, and storage rooms. The building was renovated by an architect in 2010.

Limited by the availability of RFID readers, we only tracked 10 key spaces (numbered from s_1 to s_{10}), as shown in Figure's 7 of highlights. These 10 spaces were situated separately on the 4 floors of the building, and its functions consisted of an administration office (f_6), a library (f_1), a seminar room (f_3), 2 laboratories (f_8 and f_9), 2 meeting rooms (f_4 and f_5), and 3 classrooms (f_2 , f_7 and f_{10}). There were 98 students (23% of the total number of college and graduate students) participating in this experiment. Each of them carried a Helicomm IP-Link 5110 active-RFID tag while performing their daily activities in the building for 8 weeks during the middle and end of a semester in 2011. Although the movement was tracked 24 hours a day during the experiment period, only the data obtained within 8:00 AM and 10:00 PM on weekdays (Monday to Friday) were used.

5.2. Variables of function-space assignment

1. Suitability preference of function f_i assigned to space s_i ($P_{f_i s_i}$). The spaces were divided into 3 groups, i.e., large (L: 90–135 m²), medium (M: 45–80 m²), and small (S: 15–20 m²) spaces. Different functions require difference sizes of space. The library and seminar rooms prefer large spaces. The administration office and 2 laboratories prefer medium spaces. The 2 meeting rooms prefer small spaces. In addition, one of the classrooms prefers a medium space, and the rest of the two classrooms prefer large spaces. Table 1 provides the types of space size and the functions' preferred size.

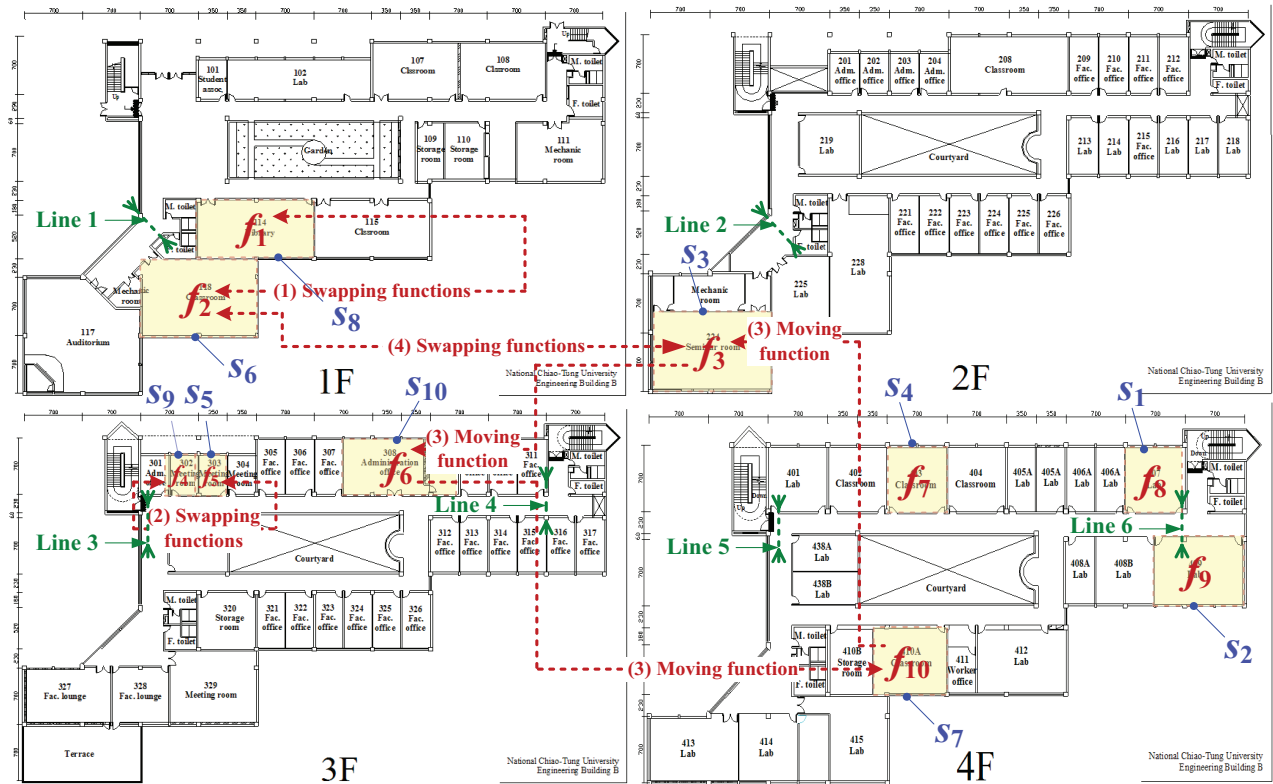


Fig. 7. Comparison of Architect's function assignment and the assignment result R_2 (Objective value = 1.10471 vs. Objective value = 1.25849)

Table 1. Space size and functions' preferred size

| Space | s_1 | s_2 | s_3 | s_4 | s_5 | s_6 | s_7 | s_8 | s_9 | s_{10} |
|----------------|---------|------------|--------------|--------------|-------------|------------|------------|------------|------------|------------|
| Size | M | M | L | M | S | L | M | L | S | L |
| Function | f_1 | f_2 | f_3 | f_4 | f_5 | f_6 | f_7 | f_8 | f_9 | f_{10} |
| Preferred size | L | L | L | S | S | M | M | M | M | L |
| Description | Library | Class room | Seminar room | Meeting room | Adm. office | Class room | Laboratory | Laboratory | Laboratory | Class room |

The suitability preference for each possible pair of function-space assignment was given based on the following principles. A function requiring a large space can only be assigned to large space. Thus, the suitability preferences ($P_{f_i s_i}$) for the function assigned to large, medium, and small spaces were 1, 0, and 0, respectively. A function requiring a medium space can only be assigned to a large or medium (with less preference) space. Thus, $P_{f_i s_i}$ for the function assigned to large, medium, and small spaces was 0.5, 1, and 0, respectively. A function requiring a small space can be assigned to any space with different preferences. Thus, $P_{f_i s_i}$ for the function assigned to large, medium, and small spaces was 0.1, 0.5, and 1, respectively. The preference values between 0 and 1 were arbitrary depending on the decision maker, who, in this case, is the head of the department. FAMOS treated the zero preference as a constraint, and never assigned a function to a space with $P_{f_i s_i} = 0$.

2. Distance between spaces s_i and s_j ($D_{s_i s_j}$). The distance between spaces s_i and s_j was given based on the normalization of the sum of the actual geographical distance and weighted floor difference value of the two spaces. The weighted floor difference value was equivalent to 0 m if the two spaces are located on the same floor, such as s_5 and s_{10} in Figure 7. The weighted value is equivalent to 20 m if the two spaces were located on two different consecutive floors, such as s_5 and s_4 . Similarly, the weighted value is 40 m and 60 m for the spaces situated 2 and 3 floors apart, respectively. Equation (4) was used to normalize the weighted distance sum to ensure that the value falls between 0 and 1. Table 2 shows the normalized distances between the 10 spaces:

$$D_{s_i s_j} = \frac{D_x - D_{\min}}{D_{\max} - D_{\min}} \quad (4)$$

Table 2. Distance between space s_i and s_j

| $D_{s_i s_j}$ | s_1 | s_2 | s_3 | s_4 | s_5 | s_6 | s_7 | s_8 | s_9 | s_{10} |
|---------------|-------|-------|-------|-------|-------|-------|-------|-------|-------|----------|
| s_1 | 0.00 | 0.97 | 0.09 | 0.78 | 0.38 | 0.00 | 0.72 | 0.04 | 0.43 | 0.60 |
| s_2 | 0.97 | 0.00 | 0.11 | 0.80 | 0.40 | 0.02 | 0.74 | 0.06 | 0.44 | 0.58 |
| s_3 | 0.09 | 0.11 | 0.00 | 0.32 | 0.57 | 0.49 | 0.24 | 0.53 | 0.61 | 0.40 |
| s_4 | 0.78 | 0.80 | 0.32 | 0.00 | 0.61 | 0.22 | 0.75 | 0.26 | 0.66 | 0.45 |
| s_5 | 0.38 | 0.40 | 0.57 | 0.61 | 0.00 | 0.47 | 0.53 | 0.51 | 0.96 | 0.84 |
| s_6 | 0.00 | 0.02 | 0.49 | 0.22 | 0.47 | 0.00 | 0.14 | 0.90 | 0.51 | 0.30 |
| s_7 | 0.72 | 0.74 | 0.24 | 0.75 | 0.53 | 0.14 | 0.00 | 0.18 | 0.57 | 0.37 |
| s_8 | 0.04 | 0.06 | 0.53 | 0.26 | 0.51 | 0.90 | 0.18 | 0.00 | 0.55 | 0.34 |
| s_9 | 0.43 | 0.44 | 0.61 | 0.66 | 0.96 | 0.51 | 0.57 | 0.55 | 0.00 | 0.80 |
| s_{10} | 0.60 | 0.58 | 0.40 | 0.45 | 0.84 | 0.30 | 0.37 | 0.34 | 0.80 | 0.00 |

3. Movement relation of functions f_i and f_j ($R_{f_i f_j}$). The movement relationship plays an important role in the optimization process. Table 3 shows the values of the interactive preference ($R_{f_i f_j}$), which was based on the movement analysis of the RFID tracking data. It signifies a usage pattern of an occupant between two functions. Basically, two functions with a large $R_{f_i f_j}$ should be located closer together to reduce the moving distance. The RFID reader read every half minute to determine which tags existed in a functional space. The reader might falsely read a tag if the tag was close to the space but did not enter the space. Therefore, data screening was essential to determine if a tagged student was really entering, using the space or just passing by.

The *minimum stay time* required to be considered as using the space was set to 1 minute for the library and administration office and 5 minutes for the rest of the functions. Thus, a tagged student who entered the administration office to retrieve mail and left after two minutes was considered to have used the office once. A student who entered a classroom to collect some textbooks he forgot during a previous lecture and left the classroom after two minutes was not considered to have used the classroom.

Table 3. Movement relation of functions f_i and f_j

| $R_{f_i f_j}$ | f_1 | f_2 | f_3 | f_4 | f_5 | f_6 | f_7 | f_8 | f_9 | f_{10} |
|---------------|-------|-------|-------|-------|-------|-------|-------|-------|-------|----------|
| f_1 | | 0.13 | 0.03 | | | | 0.01 | | 0.01 | |
| f_2 | 0.16 | | 0.02 | | | | 0.11 | 0.01 | 0.02 | 0.06 |
| f_3 | 0.01 | 0.04 | | | | 0.02 | | 0.04 | 0.06 | |
| f_4 | | | | | | | | | | |
| f_5 | 0.01 | | | | | | 0.02 | | | 0.02 |
| f_6 | | | | | | | 0.08 | 0.07 | 0.01 | |
| f_7 | | 0.05 | | | 0.01 | 0.12 | 0.00 | | 0.38 | 0.02 |
| f_8 | 0.01 | | 0.05 | 0.02 | | 0.04 | 0.04 | | 0.72 | |
| f_9 | 0.01 | 0.01 | 0.01 | | | 0.03 | 0.09 | 0.75 | 0.00 | |
| f_{10} | | | 0.03 | | 0.05 | 0.02 | 0.05 | | 0.02 | |

When a student shifted from a function to another, there was a time lag in between because he needed a break or travelling time to move between physical spaces. A short break between two functions should be considered a usage pattern, meaning that students tended to participate in a function after participating in another. Long breaks between two functions were not considered to have a usage pattern. The *maximum break time* allowed in the experiment was 30 minutes. Thus, for example, a student leaving from a class and entering a laboratory 15 minutes later would construct a usage pattern between the class and the laboratory. A student leaving from a laboratory, having lunch, and entering a seminar room 45 minutes later would not construct a usage pattern.

5.3. Function assignment and movement simulation results

The FAMOS was run on a Pentium 3.40 GHz PC with 512 MB RAM, with the parameters *epoch_max* and *era_max* set to 5 and 4, respectively. Table 4 shows the different function-space assignments suggested by the architect (A_0), FAMOS (R_1), and administrator-adjusted versions (R_2 – R_5), and their corresponding performances. The performance data include the objective values obtained from the FAMOS optimization module and the movement distance and cumulative flow simulated by the FAMOS module.

The objective-function column shows that the FAMOS assignment has a 14.80% higher objective value than the architect’s assignment. The functions are also assigned to the space sizes most preferred by the administrator (e.g., f_1 is assigned to the large spaces of s_6 , and f_5 is assigned to the small spaces of s_9). Additionally, functions with a larger value of $R_{f_i f_j}$ are also placed at least at the same floor (e.g., functions f_8 and f_9 and functions f_7 and f_9 are on floor 4).

Although FAMOS produces the assignment with the highest objective value, the administrator attempted to adjust some specific function-space assignment to emphasize some sub-objectives such as movement distance. Alternatively, the administrator may consider other sub-objectives that were or could not be included in the objective function, such as congestion in corridors or the interference to the library because of congestion. The colored traced lines in Figure 3(a) show the approximate usage load of the corridors as occupants move by. Consequently, we set up several monitoring points for some locations near staircases or exits (Fig. 7, lines 1–6) and repeated the simulation to gather detailed flow density statistics.

Table 4 presents four adjustments (R_2 – R_5) in the order of their objective values. Compared to the architect’s version (A_0), the FAMOS version (R_1) reduced the movement distance by 1.89%. This result was expected because the movement distance is part of the objective function. The FAMOS version also reduced the cumulative flow density by 12.25%, despite the flow density

Table 4. Function-space assignments and movement simulation results

| Results | Function-space assignment | | | | | | | | | | Objective value | | Movement distance (m) | | Cumulative flow density (man-time) | |
|---------|---------------------------|-------|----------|-------|-------|----------|----------|-------|-------|----------|-----------------|---------------------------------|-----------------------|---------------------------------|------------------------------------|----------------------------------|
| | s_1 | s_2 | s_3 | s_4 | s_5 | s_6 | s_7 | s_8 | s_9 | s_{10} | Value | Improvement | Value | Improvement | Value | Improvement |
| A_0 | f_8 | f_9 | f_3 | f_7 | f_5 | f_2 | f_{10} | f_1 | f_4 | f_6 | 1.10471 | $\frac{(A_0-A_0)/A_0}{0.0\%}$ | 172.503 | $\frac{(A_0-A_0)/A_0}{0.0\%}$ | 2.392 | $\frac{(A_0-A_0)/A_0}{0.00\%}$ |
| R_1 | f_8 | f_9 | f_{10} | f_7 | f_4 | f_1 | f_6 | f_2 | f_5 | f_3 | 1.26821 | $\frac{(R_1-A_0)/A_0}{14.80\%}$ | 169.244 | $\frac{(R_1-A_0)/A_0}{-1.89\%}$ | 2.099 | $\frac{(R_1-A_0)/A_0}{-12.25\%}$ |
| R_2 | f_8 | f_9 | f_1 | f_7 | f_4 | f_{10} | f_6 | f_2 | f_5 | f_3 | 1.25849 | $\frac{(R_2-R_1)/R_1}{-0.77\%}$ | 169.856 | $\frac{(R_2-R_1)/R_1}{0.36\%}$ | 1.794 | $\frac{(R_2-R_1)/R_1}{-14.53\%}$ |
| R_3 | f_8 | f_9 | f_{10} | f_7 | f_5 | f_1 | f_6 | f_2 | f_4 | f_3 | 1.26784 | $\frac{(R_3-R_1)/R_1}{-0.03\%}$ | 169.371 | $\frac{(R_3-R_1)/R_1}{0.08\%}$ | 1.976 | $\frac{(R_3-R_1)/R_1}{-5.86\%}$ |
| R_4 | f_9 | f_8 | f_{10} | f_7 | f_4 | f_1 | f_6 | f_2 | f_5 | f_3 | 1.26739 | $\frac{(R_4-R_1)/R_1}{-0.06\%}$ | 169.816 | $\frac{(R_4-R_1)/R_1}{0.34\%}$ | 2.014 | $\frac{(R_4-R_1)/R_1}{-4.05\%}$ |
| R_5 | f_8 | f_9 | f_{10} | f_7 | f_4 | f_2 | f_6 | f_1 | f_5 | f_3 | 1.26729 | $\frac{(R_5-R_1)/R_1}{-0.07\%}$ | 172.130 | $\frac{(R_5-R_1)/R_1}{1.71\%}$ | 2.132 | $\frac{(R_5-R_1)/R_1}{1.57\%}$ |

not being part of the objective function. One possible explanation is that the FAMOS placement of the undergraduate classroom (f_2), which has a much higher frequency of use compared to the library (f_1) because of the required courses of undergraduate students, in the space (s_8) near the building entrance not only reduced the movement distance, but also reduced the flow density near the library.

Based on the FAMOS version, the administrator tried to swap some function assignments with the same space size requirement to further mitigate the interference of movement flow near the library. The R_2 version, shown in Table 4, is the best assignment among several attempts by the administrator. Compared to R_1 , R_2 swaps the assignments of the f_1 (library) and f_{10} (graduate classroom) and places the library in space s_3 , a space with a less populated neighborhood. The result is a significant reduction in flow density by 14.53% with only slightly worse performance on the objective value (-0.77%) and movement distance (+0.36%). This decrease in performance is the result of placing two strongly related functions (i.e., library (f_1) and undergraduate classroom (f_2)) separately in spaces on different floors.

As shown in Figure 7, the R_2 version needs four groups of swapping or moving compared to architect's function assignments (A_0), including (1) swapping the assignments of the f_1 (library) and f_2 (classroom); (2) swapping the assignments of the f_5 (meeting room) and f_9 (meeting room); (3) moving the assignments of the f_{10} (classroom), f_3 (seminar room) and f_6 (Administrator office); (4) swapping the assignments of the f_{10} (classroom) and f_1 (library). In addition to R_2 , we also tried to swap some function-space assignments (i.e., R_3 , R_4 , and R_5) with the same space size requirement to reduce movement distance, but were unable to find a better assignment. Besides, Figure 8 shows a comparison of these designs in 3D from the perspectives of objective function (x -axis), movement distance (y -axis), and cumulative

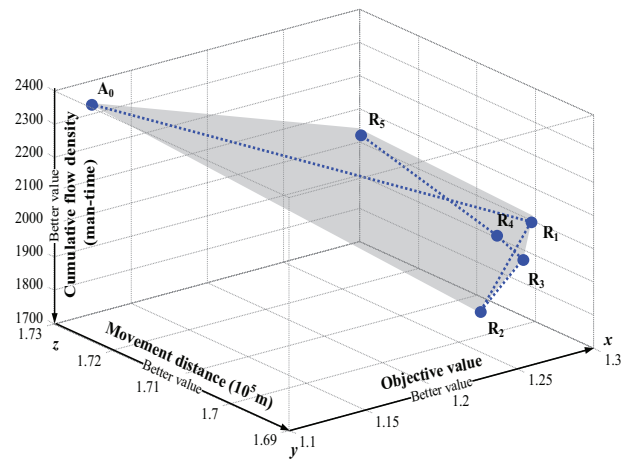


Fig. 8. Comparison of the performances in different function-space assignments

flow density (z -axis). At the end of the experiment, the administrator chose the R_2 version because he thought that its slightly worse performance on the objective value and movement distance could be tolerated to gain the benefit of less interference surrounding the library.

Therefore, the R_2 version is a best assignment and its effectiveness is described as follows: Compared to the architect's version (A_0), the R_2 version increased the objective-function by 13.92%, reduced the movement distance by 1.53%, and reduced the cumulative flow density by 25.00%.

Conclusions

This study presents the FAMOS model, which takes occupants' movement data and building floor plans as its input, and then searches for the optimal function-space assignment in a building. The FAMOS model has contributed to the following several aspects: First, integrating three techniques (that is, the RFID, fmGA, and movement simulation techniques) into the FAMOS mod-

el, which is a innovative model to systematically deal with real world the function-space assignment problems. Second, the FAMOS model can help architects or administrators to find the best assignment or improve their assignment according to their desired objectives, such as preferred space size, minimized movement distance, or removal of corridor congestion. Third, a real case study demonstrates the use of FAMOS and compares its assignments with those generated by a renovation architect/administrator. The objective function showed that FAMOS's assignment (R_1 version) had a 14.80% higher objective value than the architect's version. In addition, experimental results show that a human could not improve the optimal assignment proposed by the FAMOS. The computing efficiency of the FAMOS was also acceptable for solving the assignment problem with 10 spaces and 10 functions, taking only 16 s on a Pentium 3.40 GHz PC with 512 MB RAM. Our future research will focus on experiments with other types of buildings and objectives, and on extending the FAMOS.

References

- Armour, G. C.; Buffa, E. S. 1963. A heuristic algorithm and simulation approach to relative allocation of facilities, *Management Science* 9(2): 294–300. <http://dx.doi.org/10.1287/mnsc.9.2.294>
- Bazaraa, M. S.; Sherali, H. D. 1980. Benders' partitioning scheme applied to a new formulation of the quadratic assignment problem, *Naval Research Logistics Quarterly* 27: 29–41. <http://dx.doi.org/10.1002/nav.3800270104>
- Blue, V.; Adler, J. 2001. Cellular automata microsimulation for modeling bi-direction pedestrian walkways, *Transportation Research Part B* 35: 293–312. [http://dx.doi.org/10.1016/S0191-2615\(99\)00052-1](http://dx.doi.org/10.1016/S0191-2615(99)00052-1)
- Bourg, D. M.; Seemann, G. 2004. *AI for game developers*. Beijing: O'Reilly Media. 392 p.
- Brambilla, R.; Cattelani, L. 2009. Mobility analysis inside buildings using Distrimobs simulator: a case study, *Building and Environment* 44(3): 595–604. <http://dx.doi.org/10.1016/j.buildenv.2008.05.008>
- Burstedde, C.; Klauk, K.; Schadschneider, A.; Zittartz, J. 2001. Simulation of pedestrian dynamics using a two-dimensional cellular automaton, *Physica A: Statistical Mechanics and its Application* 295(3–4): 507–525.
- Cheung, S. O.; Tong, T. K.; Tam, C. M. 2002. Site pre-cast yard layout arrangement through genetic algorithms, *Automation in Construction* 11(1): 35–46. [http://dx.doi.org/10.1016/S0926-5805\(01\)00044-9](http://dx.doi.org/10.1016/S0926-5805(01)00044-9)
- Cheng, M. Y.; Wu, Y. W.; Wu, C. F. 2010. Project success prediction using an evolutionary support vector machine inference model, *Automation in Construction* 19(3): 619–629. <http://dx.doi.org/10.1016/j.autcon.2010.02.008>
- Choudhary, R.; Malkawi, A.; Papalambros, P. Y. 2005. Analytic target cascading in simulation-based building design, *Automation in Construction* 14(4): 551–568. <http://dx.doi.org/10.1016/j.autcon.2004.11.004>
- Christofides, N.; Benavent, E. 1989. An exact algorithm for the quadratic assignment problem, *Operations Research* 37(25): 760–768. <http://dx.doi.org/10.1287/opre.37.5.760>
- Dzeng, R. J.; Hsiao, F. Y.; Xie, Y. Q. 2012. A simulation and visualization model for occupants' movements in instructional buildings, *Computer-Aided Design and Application* 9(1): 95–100. <http://dx.doi.org/10.3722/cadaps.2012.95-110>
- Drezner, Z. 1987. A heuristic procedure for the layout of a large number of facilities, *International Journal of Management Science* 33(7): 907–915.
- Feng, C. W.; Wu, H. T. 2006. Integrating fmGA and CYCLONE to optimize the schedule of dispatching RMC trucks, *Automation in Construction* 15(2): 186–199. <http://dx.doi.org/10.1016/j.autcon.2005.04.001>
- Federal Highway Administration. 2012. *Designing sidewalks and trails for access*. Part II of II: Best practices design guide [online], [cited 21 June, 2012]. Available from Internet: http://www.fhwa.dot.gov/environment/bicycle_pedestrian/publications/sidewalk2/sidewalks208.cfm
- Goldberg, D. E.; Korb, B.; Deb, K. 1989. Messy genetic algorithms: motivation, analysis, and first results, *Complex Systems* 3(5): 493–530. <http://dx.doi.org/10.1007/978-1-4757-3643-4>
- Goldberg, D. E.; Deb, K.; Korb, B. 1991. Don't worry, be messy, in *Proc. of the Fourth International Conference on Genetic Algorithms*, 1991, La Jolla, CA, USA, 24–30.
- Goldberg, D. E.; Deb, K.; Kaegupta, H.; Harik, G. 1993. Rapid, accurate optimization of difficult problems using fast messy genetic algorithms, in *Proc. of the Fifth International Conference on Genetic Algorithms*, 1993, Urbana-Champaign, IL, USA, 56–64.
- Goldberg, D. E. 2002. *The design of innovation: lessons from and for competent genetic algorithms*. Boston: Kluwer Academic Pb. 248 p.
- Hahn, P. M.; Jrarup, J. 2000. A hospital facility layout problem finally solved, *Journal of Intelligent Manufacturing* 15(5/6): 1–11.
- Halhal, D.; Walters, G. A.; Savic, D. A.; Ouazar, D. 1999. Scheduling of water distribution system rehabilitation using structured messy genetic algorithms, *Evolutionary Computation* 7(3): 311–329. <http://dx.doi.org/10.1162/evco.1999.7.3.311>
- Hassan, M. M. D.; Hogg, G. L.; Smith, D. R. 1986. SHAPE: a construction algorithm for area placement evaluation, *International Journal of Production Research* 24(5): 1283–1295. <http://dx.doi.org/10.1080/00207548608919803>
- Hoffmann, F.; Pfister, G. 1996. Learning of a fuzzy control rule base using messy genetic algorithms, in F. Herrera (Ed.). *Genetic algorithms and soft computing*. Heidelberg: Physica-Verlag, 279–305.
- Holland, J. H. 1975. *Adaptation in natural and artificial system*. Ann Arbor: University of Michigan. 211 p.
- Hong, T.; Chou, S. K.; Bong, T. Y. 2000. Building simulation: an overview of developments and information sources, *Building and Environment* 35(4): 347–361. [http://dx.doi.org/10.1016/S0360-1323\(99\)00023-2](http://dx.doi.org/10.1016/S0360-1323(99)00023-2)
- Jang, H.; Lee, S.; Choi, S. 2007. Optimization of floor-level construction material layout using Genetic Algorithms, *Automation in Construction* 16(4): 531–545. <http://dx.doi.org/10.1016/j.autcon.2006.09.006>
- Jo, J. R.; Gero, J. S. 1998. Space layout planning using an evolutionary approach, *Artificial Intelligence in Engineering* 12(3): 149–162. [http://dx.doi.org/10.1016/S0954-1810\(97\)00037-X](http://dx.doi.org/10.1016/S0954-1810(97)00037-X)
- Kaklauskas, A.; Zavadskas, E. K.; Galiniene, B. 2008. A building's refurbishment knowledge-based decision support system, *International Journal of Environment and Pollution* 35(2–4): 237–249. <http://dx.doi.org/10.1504/IJEP.2008.021358>
- Kaklauskas, A.; Zavadskas, E. K.; Raslanas, S. 2005. Multivariate design and multiple criteria analysis of building refurbishments, *Energy and Buildings* 37(4): 361–372. <http://dx.doi.org/10.1016/j.enbuild.2004.07.005>
- Kalay, Y. E. 2004. *Architecture's new media: principles, theories, and methods of computer-aided design*. Cambridge: MIT Press. 536 p.

- Keßel, A.; Klüpfel, H.; Schreckenberg, M. 2002. Microscopic simulation of pedestrian crowd motion, in M. Schreckenberg, S. D. Sharma (Eds.). *Pedestrian and evacuation dynamics*. Berlin: Springer, 193–202.
- Khalil, T. M. 1973. Facilities relative allocation technique (FRAT), *International Journal of Productions Research* 11(2): 183–194.
<http://dx.doi.org/10.1080/00207547308929959>
- Koopmans, T. C.; Beckman, M. 1975. Assignment problems and the location of economic activities, *Econometrica* 25(1): 53–76. <http://dx.doi.org/10.2307/1907742>
- Lanzi, P. L. 1999. Extending the representation of classifier conditions Part I: From binary to messy coding, in *Proc. of the Genetic and Evolutionary Computation Conference (GECCO 1999)*, 1999, Orlando, Florida, USA, 337–344.
- Lanzi, P. L.; Perrucci, A. 1999. Extending the representation of classifier conditions Part II: From messy coding to S-expressions, in *Proc. of the Genetic and Evolutionary Computation Conference (GECCO 1999)*, 1999, Orlando, Florida, USA, 345–352.
- Lee, H. Y.; Yang, I. T.; Lin, Y. C. 2012. Laying out the occupant flows in public buildings for operating efficiency, *Building and Environment* 51: 231–242.
<http://dx.doi.org/10.1016/j.buildenv.2011.11.005>
- Lee, H. Y. 2012. Renovation scheduling to minimize user impact of a building that remains in operation, *Automation in Construction* 22: 398–405.
<http://dx.doi.org/10.1016/j.autcon.2011.09.018>
- Lee, R.; Moore, J. M. 1967. CORELAP-computerized relationship layout planning, *The Journal of Industrial Engineering* 18: 195–200.
- Liang, L. Y.; Chao, W. C. 2008. The strategies of tabu search technique for facility layout optimization, *Automation in Construction* 17(6): 657–669.
<http://dx.doi.org/10.1016/j.autcon.2008.01.001>
- Liggett, R. S.; Mitchell, W. J. 1981. Optimal space planning in practice, *Computer-Aided Design* 13(5): 277–288.
[http://dx.doi.org/10.1016/0010-4485\(81\)90317-1](http://dx.doi.org/10.1016/0010-4485(81)90317-1)
- Lionel, M. N.; Liu, Y. H.; Lau, Y. C. 2004. Patil AP. LAND-MARC: indoor location sensing using active RFID, *Wireless Networks* 10(6): 701–710.
<http://dx.doi.org/10.1023/B:WINE.0000044029.06344.dd>
- Loiola, E. M.; Abreu, N. M. M. D.; Boaventura-Netto, P. O.; Hahn, P.; Querido, T. 2007. A survey for the quadratic assignment problem, *European Journal of Operational Research* 176(2): 657–690.
<http://dx.doi.org/10.1016/j.ejor.2005.09.032>
- Mohan, C. K. 1993. A messy genetic algorithms for clustering, in C. H. Dagli, L. I. Burke, B. R. Feranandez, J. Ghosh (Eds.). *Intelligent engineering systems through artificial neural networks*. New York: ASME Press, 831–836.
- Rahul, N.; Abhinav, G.; Sean, C.; Ming, C. L. 2009. Aggregate dynamics for dense crowd simulation, *ACM Transactions on Graphics* 28(5): 122(1–8).
- Ramkumar, A. S.; Ponnambalam, S. G.; Jawahar, N. 2009. A new iterated fast local search heuristic for solving QAP formulation in facility layout design, *Robotics and Computer-Integrated Manufacturing* 25(3): 620–629.
<http://dx.doi.org/10.1016/j.rcim.2008.03.022>
- Seehof, J. M.; Evans, W. O. 1967. Automated layout design program, *The Journal of Industrial Engineering* 18: 690–695.
- Solimanpur, M.; Jafari, A. 2008. Optimal solution for the two-dimensional facility layout problem using a branch-and-bound algorithm, *Computers & Industrial Engineering* 55(3): 606–619.
<http://dx.doi.org/10.1016/j.cie.2008.01.018>
- Solimanpur, M.; Vrat, P.; Shankar, R. 2005. An ant algorithm for the single row layout problem in flexible manufacturing systems, *Computers and Operations Research* 32(3): 583–698. <http://dx.doi.org/10.1016/j.cor.2003.08.005>
- Skurikhin, A. N.; Surkan, A. J. 1996. Messy genetic algorithm learns a classifier to design multiplexers, in *Proc. of IEEE International Conference on Evolutionary Computation*, 1996, Nayoya University, Japan, 100–103.
<http://dx.doi.org/10.1109/ICEC.1996.542341>
- Tesoriero, R.; Tebar, R.; Gallud, J. A.; Lozano, M. D.; Penichet, V. M. R. 2010. Improving location awareness in indoor spaces using RFID technology, *Expert Systems with Applications* 37(1): 894–898.
<http://dx.doi.org/10.1016/j.eswa.2009.05.062>
- Tompkins, J. A.; Reed, J. R. 1967. An applied model for the facilities design problem, *International Journal of Production Research* 14(5): 583–595.
<http://dx.doi.org/10.1080/00207547608956377>
- Yang, L. Z.; Zhao, D. L.; Li, J.; Fang, T. Y. 2005. Simulation of the kin behavior in building occupant evacuation based on Cellular Automaton, *Building and Environment* 40(3): 411–415.
<http://dx.doi.org/10.1016/j.buildenv.2004.08.005>
- Yeh, I.-C. 2006. Architectural layout optimization using annealed neural network, *Automation in Construction* 15(4): 531–539. <http://dx.doi.org/10.1016/j.autcon.2005.07.002>
- Yue, H.; Hao, H.; Chen, X.; Shao, C. 2007. Simulation of pedestrian flow on square lattice based on cellular automata model, *Physica A: Statistical Mechanics and its Applications* 384(2): 567–588.
- Zavadskas, E. K.; Antucheviciene, J. 2007. Multiple criteria evaluation of rural building's regeneration alternatives, *Building and Environment* 42(1): 436–451.
<http://dx.doi.org/10.1016/j.buildenv.2005.08.001>
- Zavadskas, E. K.; Kaklauskas, A.; Turskis, Z.; Kalibatas, D. 2009. An approach to multi-attribute assessment of indoor environment before and after refurbishment of dwellings, *Journal of Environmental Engineering and Landscape Management* 17(1): 5–11.
<http://dx.doi.org/10.3846/1648-6897.2009.17.5-11>

Ren-Jye DZENG. Is a Professor at the Department of Civil Engineering, National Chiao-Tung University, Taiwan. He is also acting as the CEO of the Eco-City: Smart Living Technologies Integrated Center, Taiwan. He received his PhD from the Department of Civil & Environmental Engineering, University of Michigan, Ann Arbor, USA. His research interests include construction engineering & management, artificial intelligence, RFID, application of brain EEG and virtual reality for construction safety.

Wei-Chih WANG. Is a Professor in the Department of Civil Engineering at National Chiao-Tung University, Taiwan. He received his PhD degree from the University of California at Berkeley in 1996. He is a member of the Construction Management Association of Taiwan. His research interests include construction procurement, high-tech facility project management, and construction simulation.

Fan-Yi HSIAO. Is a PhD Candidate at the Department of Civil Engineering, National Chiao-Tung University, Taiwan. She received her Master degree from the Department of Civil Engineering, National Chiao-Tung University, Taiwan. Her research interests include construction engineering & management, artificial intelligence, and RFID application.