Wen J., Daher E. and Kubicki S. (2020). "Optimization of a user-involved floor layout recommendation system at the operation stage" In: *Proc. 37th CIB W78 Information Technology for Construction Conference* (CIB W78), São Paulo, Brazil, pp. 385-400. DOI: http://dx.doi.org/10.46421/2706-6568.37.2020.paper028

# OPTIMIZATION OF A USER-INVOLVED FLOOR LAYOUT RECOMMENDATION SYSTEM AT THE OPERATION STAGE

**Junhan Wen[1], Elie Daher[2], and Sylvain Kubicki[3]**

**Abstract:** The parametric modelling method, an algorithmic thinking process based on visual programming, is recently established as a supportive tool in the decision-making practice for architecture. For the time being, the method helps to inform, control and optimize the architectural design by expressing the input parameters and conditional rules that are generated according to the design objectives. This paper describes how a parametric-model-based recommendation system is developed for an interior floor layout optimization problem which supports user involvement in different aspects of the process. The system connects the design objectives and the user preferences to propose customized space layouts in the operation stage of buildings, and one of the feasible relaxations of this design study is to substantially generalize it as a multi-objective optimization problem. As such, the algorithm- part of the system contains three different functionalities: (a) a screening scheme to select the available spaces concerning given requirements, (b) a generation process to figure out different possibilities of interior plans, and followed by (c) an evaluation system to compare and recommend the best-matching solutions. Simultaneously, the user interface of the system builds up various interactions between the users and the parametric models throughout the process of design, so as to collect the criteria, preferences, and priorities of the design objectives. The recommendation system is also implemented in a real case study, the floor planning of the IAK-1 building of the European Investment Bank in Kirchberg, Luxembourg, to assess the algorithm performance and user experience. The results illustrate the applicability of this approach in real-life design, and the pros and cons of the generated plans are also analysed by comparing to traditional designs given by expert architects.

**Keywords:** Floor Planning, Multi-Objective Optimization, Parametric Modelling, Algorithmic Design.

## 1 INTRODUCTION

The design of a complex interior space-planning layout is always a complicated task for design teams and facility management teams. One of the major challenges is to propose solutions that satisfy the client requirements, respecting the building standards and the needed characteristics. In practice with the traditional design method, finding a solution that answers most of the constraints and input is a complicated task under various limitations. Nowadays, with the capabilities offered by new technologies for architectural

---

[1] Luxembourg Institute of Science and Technology, Esch-sur-Alzette, Luxembourg, junhan.wen@outlook.com

[2] Luxembourg Institute of Science and Technology, Esch-sur-Alzette, Luxembourg, elie.daher@list.lu

[3] Dr. Arch., Luxembourg Institute of Science and Technology, Esch-sur-Alzette, Luxembourg, sylvain.kubicki@list.lu

design process, there come possibilities to develop algorithms and programs to answer the design problems (Shea, Aish and Gourtovaia, 2005; Brown and Katz, 2011; Krish, 2011; Stavrić and Marina, 2011; Lee, Gu and Williams, 2014). Among the previous researches, one of the major limitations with the analytical optimization techniques is that the end users cannot always involve in the main design process.

Recently, the application of these computational methods has increasingly been thriving for interior space planning. Many researchers and practitioner architects are trying to investigate in generating space planning with the support of computational tools. The focus of this research is to enable the participation of users in the re-allocation of the interior floor plan with new functions, with employing programming-based systems: a series of algorithmic and parametric design tools combined with other participative devices such as tangible tabletops and graphical interfaces. Design decisions are recommended by an algorithm developed for this purpose with the help of parametric building data. Besides, the users are also given the ability to evaluate the design from the visual performance indicators running on the graphical and participative interfaces.

## 2 RELATED WORK

### 2.1 Space Planning and Generative Algorithm

Several academic and applied pieces of research were developed to investigate the generation and optimization of space planning with the use of computational architectural tools (Dutta and Sarthak, 2011). Boon et al. (2015) created an analytical system intended to enhance efficiency within the groups of the complexly-interconnected architectural program considering the adjacency between the spaces (Boon *et al.*, 2015). Another attempt developed by the architectural office Shepley Bulfinch aimed to determine the overall form and function of a building connected to a BIM tool, in order to provide a platform for programming and generating early geometrical form with physical and numerical parameters as input, i.e. size, shape, purpose and adjacency between functions (Nagy *et al.*, 2017). The Autodesk researchers also worked in developing a flexible generative workflow that creates a variety of space planning layouts including locating all necessary programs and people using a limited set of input parameters in two of their researches (Nagy *et al.*, 2017; Villaggi *et al.*, 2018). Bahrehmand (2017) developed a genetic algorithm for space planning with a multi-parental recombination method (Bahrehmand *et al.*, 2017). In another research, Das (2016) created scalable algorithms from computational geometry to deliver rational architectural space plans (Das, 2017).

In general, the proposed studies well implemented the generative algorithm into space planning, among which some systems were also able to generate satisfactory-feasible floor plans based on spatial quality metrics. Nevertheless, when the decision is generated based on the designer’s taste, limitations occur when the specific requirements from the end-users and the corresponding participatory approaches during the design progress need to be concerned. Since the design objectives and specifications change very often from the users, a system that integrates the first party in the process is necessary to be developed.

## 2.2 Decision-Making in Multi-Objective Question

In this specific case, since the separations and circulars are fixed at the operation stage of the interior layout, the problem of optimizing the office floor plan, could be essentially interpreted as a multi-objective allocation problem, of which the solutions might have the fitness from diverse aspects. For this master problem, two main directions of finding solutions are suggested, either classifying solutions from a best non-dominated level, or projecting the solutions to a created dimension so as to have a comparable objective function. Aligning to the former suggestion, Chaharsooghi et al. enhanced the ant colony optimization algorithm to figure out an allocation (Chaharsooghi, 2008), and Govil considered the cost of re-allocation as the objective to be optimized (Govil, 2011). Since the design problem in the operation stage limits the number of plan possibilities, it makes the constraints more stringent than common NP-hard problems. Therefore, it would be better to consider the problem as NP-complete in this specific situation, from which a searching scheme could be figured out to find the global optima with minimum iterations. Simultaneously, the use of a mathematical tree to reduce the search space for this type of problem (Garey, Graham and Johnson, 1976; Pan, Yu and Wang, 2003; Sun, 2004), inspires a direction to enhance the computing efficiency of defining the scope of searching.

In terms of deciding the objective function, fuzzy logic is taken into consideration because it is commonly employed as a control rule to decide the degree of output membership [16]. As illustrated in Figure 1: Illustration of the Fuzzy Logic.Figure 1, at least two control values are required to represent the critical range of 1 and 0, and the output will be calculated linearly when the input is between the thresholds. This rule is implemented in many optimization algorithms. For instance, Kumar et al. defined rules to decide the temperature-control parameters for a process (Kumar Singh, Singh and Gangwar, 2018). Chaouali et al. and Leonori et al. used fuzzy logic to supervise the energy management strategy (Leonori, De Santis, Rizzi and F. M. Frattale Mascioli, 2016; Leonori, De Santis, Rizzi and F. M.Frattale Mascioli, 2016; Chaouali *et al.*, 2018). These researches validate that fuzzy logic is a practical parameter configuration method as it could form a mono-objective function through convex combinations effortlessly.

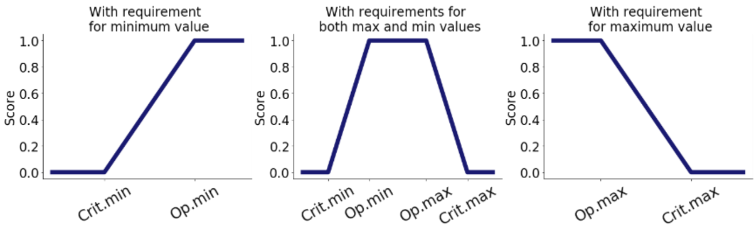

Figure 1: Illustration of the Fuzzy Logic.

Previous parametric modelling studies show that the time needed for computing and optimization is one of the substantial constraints and challenges of parametric modelling design in practice (Bittermann, Ciftcioglu and Sariyildiz, 2009; Dino, 2012; Zarei, 2012). Responding to this, fuzzy logic could be a choice to save the computing power, owing to its linear way of degree evaluation. Meanwhile, for this specific multi-objective optimization problem, some simulation results might give contradictory recommendations. Hence, instead of finding a feasible non-dominated dimension, the concept of creating combinations to form a new objective function could play as a better role, i.e. considering projecting to a self-defined dimension. In this case, the linear way of combination from fuzzy logic could also help in limiting the running time of functioning. Hence, fuzzy logic is decided as the major tool to evaluate the plan performances.

## 3 PROBLEM DESCRIPTION

### 3.1 Research Gap

Traditionally, space planning layout is designed by the architects, and the influences of noise, light, thermal comfort are considered empirically from a qualitative perspective. Nevertheless, the development of building simulation technologies gives more information and opportunities for the design process and design solutions. These solutions are generated in a top-down optimization approach, and the involvement of users is limited in the initial phase of the design process. This brings out a question: how to make appropriate deployment of the simulation data to optimize the working environment layout, which both meets the design constraints and allows a high level of users involvements throughout the process?

Concerning the research status of parametric design, a study is incubated to bridge the involvement of architects and users during the design phase. To answer the gap, a solution is proposed to create parametric interactions between the end-users and the allocation of functions of rooms in the layout of the office spaces. From a human-oriented aspect, it provokes another research question: how to facilitate this computational procedure to customize the plan in order to achieve an interactive and favourable user experience?

### 3.2 Research Objectives

Essentially, the study aims at developing a system that is capable to find the space planning layout of a building that fits the characteristics of the new functions so as to best re-allocate the floor plan according to the customer's definition of specific needs. The core algorithm aims at achieving the best coupling between the inputs of requirement and preferences from customers and the physical comfort of the building. Specifically, the objectives of this study are:

- To develop a scheme to quantify the space characteristics and user preferences.
- To form and optimize a planning algorithm in order to figure out the optimal solution from the two inputs in a qualitative and fast-functioning way.
- To visualize the result in clearly and concretely so that the users are able to compare and select their favoured plan(s).

## 4 METHODOLOGY

### 4.1 Assumptions

Some assumptions and constraints are made before developing the system:

- **Concerning the main task:** The system works on optimizing the floor plan at the operation stage, with all the separation walls and circulars constructed. Therefore, the new plan will be generated by re-allocating the newly-required functions of rooms (hereafter, called "functions" for short) into the existing interior spaces (hereafter, called "spaces" for short). Nevertheless, in case a larger area for a specific function is asked, there should be suggestions for removing the non-structural walls to form a larger space.

- **Concerning the space feature input:** The features of spaces consist of individual characteristics of each space, and the adjacency level between every two spaces. The individual features include the noises level from interior and exterior, the daylight and the radiation received during a year, as well as the views to the outside and to avoidable places like the restrooms. Due to the limitations of the model, such as available weather data and simplification during the simulation process, the space features cannot always present the absolute conditions. Nevertheless, since the algorithm considers the "best-fitting solution", which principally expects the comparative space conditions to be illustrated, the values of these features are still practical for the recommendation system.
- The connection between spaces, recorded in a symmetric matrix[4], are classified into three levels: (a) the highest level for horizontally adjacent spaces, (b) the medium level for spaces that could be reached vertically (close to a staircase or an elevator), and (c) the lowest for the rest cases.
- **Concerning the user-involving input:** On the starting stage, the customers are asked to define the priority class of every function. Then the requirements of each function could be given either quantitatively or qualitatively. The qualitative inputs, such as "the best 10%", will be quantified by an auxiliary module[5]. The favoured levels of features given by the customer are considered as the optimal range, and the system will also generate a larger value range, called the acceptable range, based on the priority classes. In addition, the customer is asked to provide the desired adjacency levels between specific room functions.
- **Concerning the scoring of fulfilment:** According to the application of fuzzy logic in decision making, when the space feature fulfils the optimal range, the score of the sufficiency will be 1. When the feature lies outside of the optima but within the acceptable scope, the score will be calculated linearly. When it is not in the acceptable range, the sufficiency will be 0 and the allocation of that function-space will be considered as invalid, i.e. the acceptable range also defines the critical values of the function requirements. These thresholds could also be altered manually if requested by the customers or building standards. Since the space adjacency is a descriptive feature, the evaluation will use a scoring dictionary instead of the fuzzy logic, ranging from 0 to 1, and no more acceptable ranges will be established for this requirement.

## 4.2 Conceptual Framework

The conceptual framework proposed is grounded in a wider theoretical paradigm developed and related to a Ph.D. study addressing the participation of users in the design as a new societal challenge essential to answer (Daher, Kubicki and Pak, 2018).

---

[4] The adjacency matrix uses names of functions as both the columns and rows, while each element shows the level of connection requirement of the two elements of its position. Hence, the result matrix will be symmetric, i.e. $A = A^T$.

[5] The auxiliary module means another python program that will run when the expert requires. The input will be the rank, shown as a number or a percent (e.g. best 5 spaces or best 10%), and the output will be the suggested value of feature requirement that could give to the system.

As illustrated in Figure 2**Erro! Fonte de referência não encontrada.**, the plan-optimization system starts with calculating the decision thresholds, and then screens the eligible spaces for each function. When there is no space that could meet all the requirements for certain functions, the system will suggest various possible approaches to deal with this circumstance. The bottom line of the optimization-computing procedure consists of three sections: plan generation, evaluation and recommendation. In the plan generation part, the system will allocate the functions to the interior spaces with respect to the specific screening results. The generated plans will then be ranked by scoring, which evaluates the overall sufficiency with weights on different features. After the assessment, the best *n* plans (using *n*=10 as a default value) will be recommended, and the final decision will be discussed with the customers, in case some cognitive perspectives are concerned. Parametric models of the saved plans will be fast rendered in Rhino, which could visualize the optimized layouts as 3D models to convene the decision of the customers.

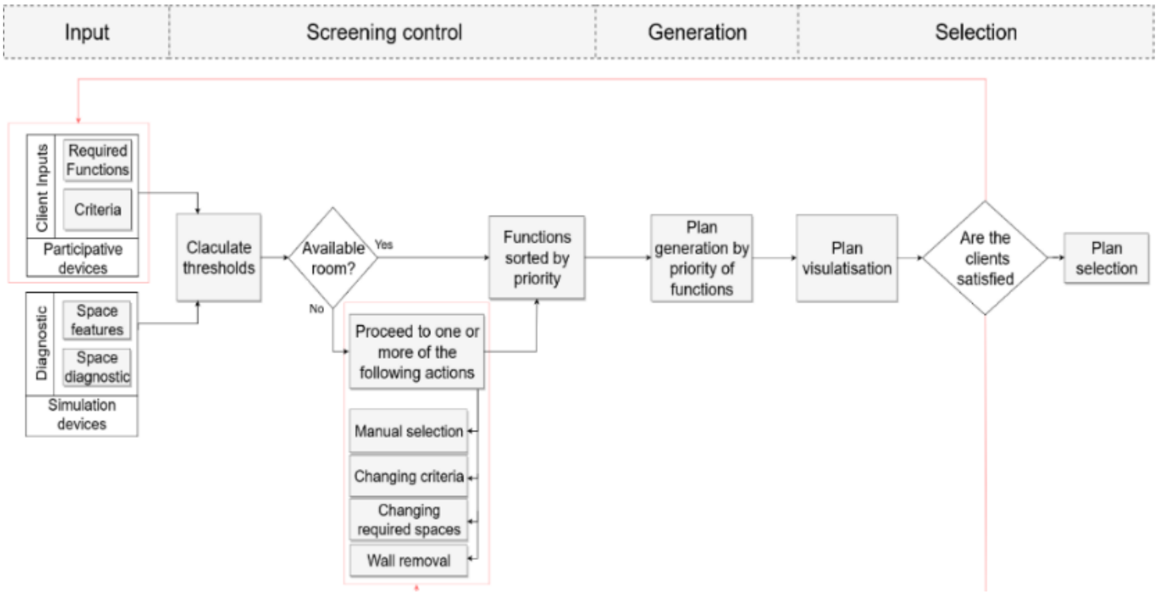

Figure 2: Illustration of Design Workflow.

## 4.3 Technical Framework

The algorithm is practised as a real-case prototype. The computation environment comprises Rhino[6] for the 3D modelling of the building structure, Grasshopper[7], the plugin for Rhino for building simulation and visualization, and Python[8] for the core algorithm development. In addition, a tangible tabletop is used as an auxiliary tool for user interface during the process and for connection with the database by PostgreSQL[9], which is also synchronized with the Python program.

---

[6] Rhino is a 3D modeler used to create, edit, analyse, document, render, animate, and translate Non-Uniform Rational B-Spline (NURBS) curves, surfaces, and solids, point clouds, and polygon meshes. The paper uses the latest version of Rhino 6.

[7] Grasshopper™ is a graphical algorithm editor tightly integrated with Rhino's modelling tools.

[8] Python is an open source programming language. The core algorithm uses Python 3.6, together with the packages NumPy, Pandas, Math, and Psycopg2 etc.

[9] PostgreSQL is an open source object-relational database system that uses and extends the SQL language. With Psycopg2, the program could take the input directly from the database.

### 4.4 Algorithm Development

Since the program covers different manipulations of information from multi-platforms, the development starts with a demo system with a small bunch of data, aiming at associating the data flow of the entire process. Then the algorithm is improved repetitively based on its performance shortcomings. This trial-and-error procedure is established as an iterative design approach to promote the final design to be reliable and practical.

## 5 ALGORITHM OPTIMIZATION

The workflow of the first demo system is to generate all the function-space allocations as the first step, and then to screen the allocations to ensure the allocated spaces meet the critical requirement of the corresponding functions. After that, the best-fitting plans are recommended. In short, it is called a G-S-O process (Generation-Selection-Optimization).

In the optimization part, the sufficiency of pairs of functions and spaces are calculated by fuzzy logic, which linearly calculates how much percent the space feature fits the optimal scope of the corresponding function. For assessing the descriptive adjacency requirements, the system uses "Boolean comparison", i.e. "*true (=1)*" for having an equal or better connection as required, and "*false (=0)*" for not holding.

The major problem arises from the first test is that a considerable long time should be spent if all the permutations are generated. With the increasing number of spaces and functions, these allocations will holding a complexity of $O(n!)$, i.e. grow in a factorial way. For instance, when allocating 5 functions into 10 spaces, the possibilities will reach a number of $A_{10}^{5} \approx 3 \times 10^{4}$. Whilst allocating 10 functions into 20 spaces, the possibilities grow to $A_{20}^{10} \approx 6 \times 10^{11}$, and require over 20 GB for storing afterwards.

To reduce the computation time, two methods are considered to improve the performance. First of all, the "priority" is added as a feature to classify the functions. Instead of finishing the entire process in one turn, the system iterates the G-S-O procedures within every priority class. For the previous example, those $6 \times 10^{11}$ plans could be reduced to $2 \times 3 \times 10^{4} = 6 \times 10^{4}$ plans. In addition, the customer could also interact with the system and select plans for the next iteration at the end of each loop.

The major problem that is causing the large computation power is that all of the functions have the chance to be placed into every space. Nevertheless, due to various requirement of every function, a large percentage of the plans could not have all the function-space allocations fulfil the critical needs. Essentially, the stricter the requirements are, the larger the combination storage wastes. Thus, another improvement is to utilize the selection result in the generation procedure, i.e. the algorithm considers the screening control before starting the design permutations.

With iterative research and experiments, a new combination algorithm is figured out as generating the function-space allocations as a tree structure. It starts with deciding the sequence of planning according to the number of possibilities. In the first step, the function with the least available spaces are allocated, and each pair of function-space forms one parent node. Then, the function with the second least available spaces forms the children leaves of that node, with removing the spaces "occupied" in the corresponding parent node. In the end, every branch in this tree forms an eligible allocation possibility without duplication.

An example of the plan generation method is shown below. Assume that 3 functions (F1, F2, F3) need to be allocated into 6 spaces (S1, S2, ..., S6), where F1 could be allocated into S1, S2 and S3, F2 has only S1 eligible, and F3 could be put in S1, S2, S3 and S4. For the enumeration method, a large proportion of allocations will not be available, including all of the plans which consider S5 or S6. As the new method illustrated in Figure 3, the only 4 capable plans are generated efficiently as the leaves of a tree.

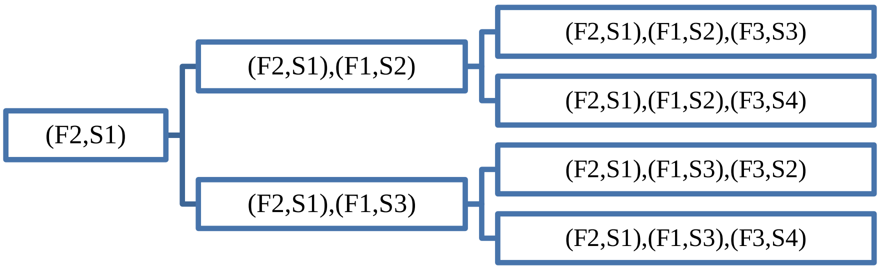

Figure 3: Example of the Tree Combination Method.

Meanwhile, there could also be the circumstance when not all of the branches consist of exactly the same functions. In this situation, the system will firstly select the longest branches of the tree, i.e. comprising the most of functions. When the branches have the same lengths yet their contents of functions do not, the system will construct additional data frames for the additional selections of functions so as to make reasonable comparisons among plans.

Some small improvements are also made in the optimization process. For the scoring of individual features, fuzzy logic is seen as an efficient and practical solution. For the evaluation of adjacencies, a renewed score dictionary is developed to make situation-wise punishments, as shown in Table 1. The full score of the adjacency is scaled to be equal to that of other features in the branch.

Table 1: Scoring Dictionary of Adjacency.

| **Obtain**<br>**Request** | 0 | 1 | 2 |
|---|---|---|---|
| 0 | - | - | - |
| 1 | 0 | 1 | 0.7 |
| 2 | 0 | 0.3 | 1 |

At the end of the algorithm, the weighted sum of the sufficiency scores will be the decider for ranking and for choosing the optimal solution. Thereafter, to limit the number of branches, only the best 10 plans will be selected as the recommendations in that priority. Before going to the next iteration, the customers could check the visualized plan and modify or select some plans to better meet their underlying principles.

# 6 CASE STUDY

## 6.1 Implementation

The system was tested in IAK building at the European Investment Bank, situated in the Kirchberg quartier in Luxembourg, and as depicted in Figure 4. The case study was to allocate 12 functions of rooms to a target area that includes 19 spaces distributed in the first two floors, which are 8-meter and 4-meter-high respectively.

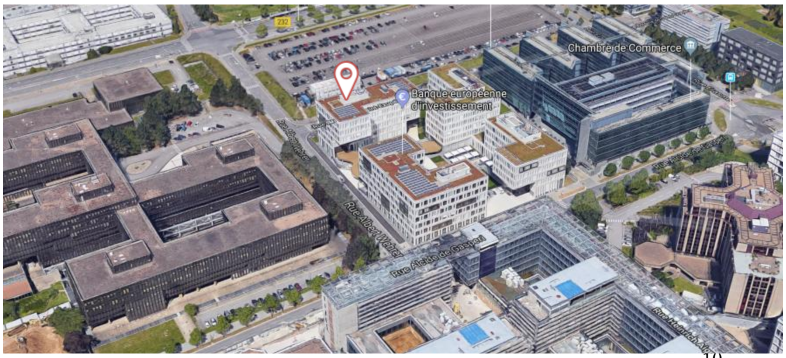

Figure 4: The neighbourhood of the IAK-1 Building[10].

The case study starts with modelling the building structure and its neighbourhood, from which the comfort simulation can be done. As previously mentioned, these simulation results compose individual space features. Selected results are depicted in Figure 5.

---

[10] Source: https://www.google.ch/maps/@49.6206851,6.1472378,230a,35y,357.89h,56.1t/data=!3m1!1e3

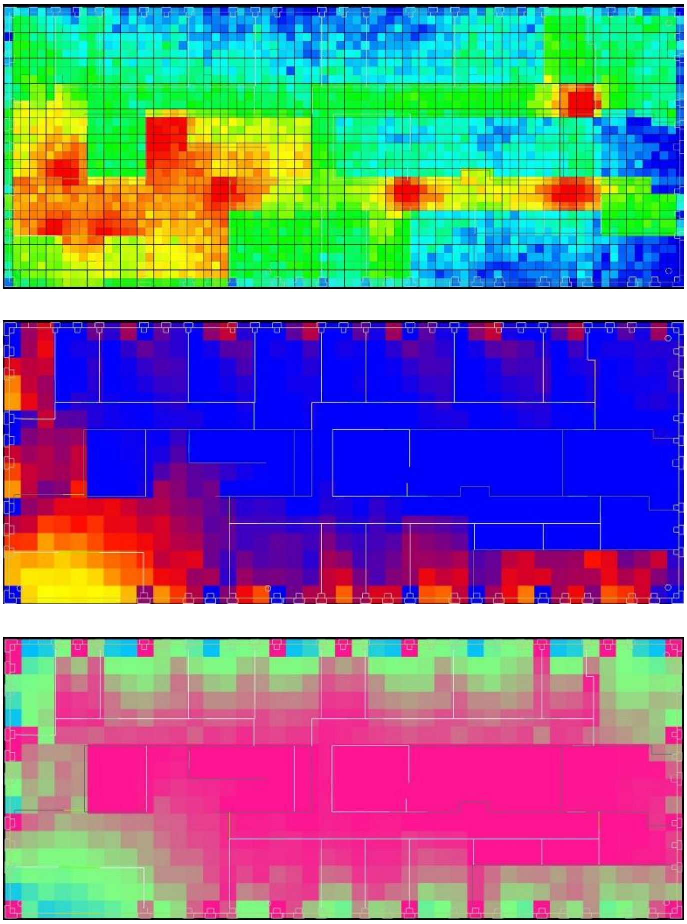

Figure 5: Simulation Results of the Internal Noise Level (top), Daylight Hours (middle), and Skyview from Windows (bottom) of the first floor from Grasshopper.

In the study, the functions are divided into 4 functions in Priority A and 8 in Priority B. With the latest plan generation method, the number of possible plans for the A priority is reduced from $A_{19}^{4} \approx 9 \times 10^{4}$ to 60. Then 10 best plans/nodes of the A-priority are retained, from which the branches of the B-priority start to grow. At the end of the Priority B, 404 plans with different numbers of functions are generated. According to the algorithm, only the longest branches are kept. Hence, 20 best plans with two combinations of 10 selected functions are kept as the recommendation, whereas 2 functions are not allocated. In this situation, the system will ask whether the customers want to consider the wall removal suggestions or to manually give their arrangement, referring to the sufficiency score of the specific function-space allocation. All of the decisions will be made with the help of a tangible table and a series of graphical interfaces. In Figure 6, three visualizations of the plan recommendations after Priority B are illustrated.

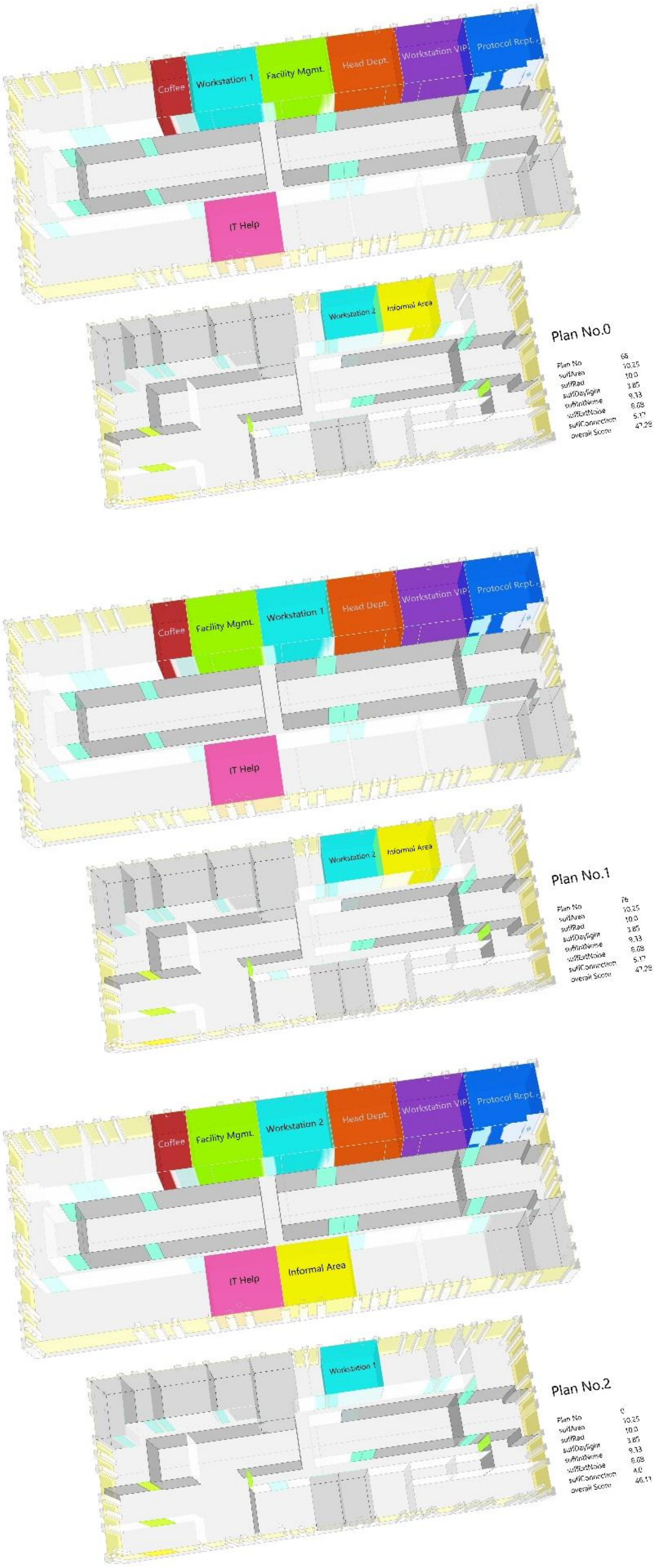

Figure 6: Visualization of the Top Three Plans Recommended by the Algorithm.

## 6.2 Results

Table 2 presents the scores of five recommended plans after the final iteration of the optimization system. The final score is the weighted summation from the six sufficiency scores. For simplification, the score in Table 2 is evenly weighted. The average score is the sum score divided by the number of comprising functions.

As is depicted, the two best plans have the same degree of sufficiency according to the proposed requirements. Since in total 10 functions are allocated in the recommendations, the results indicate that the selected plans have excellent sufficiency in terms of area and radiation requirements and good performance of environmental noise. The scores of direct daylighting imply a relatively lower performance compared to other criteria, which might due to the insufficient lighting condition of the whole building, as could be referred to in the middle of Figure 5. Since the degree of connection is a group feature of specific combinations, it might be limited by the plans that are generated by only screening the individual space features. Thus, the adjacency score is not horizontally comparable with other features. Nevertheless, this criterion is, to some extent, the decider of the final recommendations under this specific circumstance.

Table 2. Sufficiency Table of Generated Plans Sorted by Overall Score

| Plan No. | Use of Functions | Suff-Area | Suff-Rad | Suff-Daylight | Suff-IntNoise | Suff-ExtNoise | Suff-Adjacency | Sum Score | Avg Score |
|---|---|---|---|---|---|---|---|---|---|
| 68 | 10 | 10.25 | 10 | 3.85 | 9.33 | 8.68 | 5.17 | 47.28 | 4.73 |
| 76 | 10 | 10.25 | 10 | 3.85 | 9.33 | 8.68 | 5.17 | 47.28 | 4.73 |
| 0 | 10 | 10.25 | 10 | 3.85 | 9.33 | 8.68 | 4 | 46.11 | 4.61 |
| 92 | 10 | 10.25 | 10 | 3.85 | 9.33 | 8.68 | 4 | 46.11 | 4.61 |
| 60 | 10 | 10.25 | 10 | 3.85 | 9.33 | 8.68 | 4 | 46.11 | 4.61 |

## 6.3 Design Evaluation

The recommendations from the algorithm are also compared with the plans designed by experienced architects under the same qualitative constraints[11]. Three senior architects and one building engineer were involved in this experiment. The evaluation and screening results are summarized in Table 3 and Table 4 respectively.

The assessment of “overall available allocations” is calculated by accumulating the combinations of functions-spaces that meet all of the critical requirement of the six criteria of individual features. Compared to other single criteria, the rather low values in the overall availability demonstrate that one of the most challenging objectives for human experts when figuring out the solutions is to ensure the decision performance meets the thresholds from every perspective simultaneously. Since the fuzzy logic does not give negative scores even when the object value is far out of acceptable range, it is indicated that there might be potential accomplishments in the traditional design to be overestimated. On the other way, it is demonstrated that one of the major competencies of the algorithm is to ensure the operation performances of the generated plans are all within the constraints of critical requirements.

---

[11] All the constraints considered simulation results are given qualitatively, e.g. “the best daylight” or “medium noise level”. The optimal areas of functions are given the same as to the program.

Table 3. Sufficiency Table of Architect-Designed Plans.

| Plan No. | Use of Functions | Suff-Area | Suff-Rad | Suff-Daylight | Suff-IntNoise | Suff-ExtNoise | Suff-Adjacency | Sum Score | Avg Score |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 12 | 10.97 | 9 | 8.09 | 11.33 | 8.56 | 3.15 | 51.10 | 4.26 |
| 2 | 12 | 11.87 | 10 | 7.71 | 10.33 | 10.88 | 6.15 | 56.94 | 4.74 |
| 3 | 11 | 10.92 | 7 | 8.38 | 9.66 | 9.88 | 5.64 | 51.48 | 4.68 |
| 4 | 12 | 9.35 | 11 | 9.17 | 11 | 9.41 | 2.55 | 52.48 | 4.37 |

Table 4. Eligible Function-Space Allocations from Architect-Designed Plans.

| Plan No. | Use of Functions | Avail-Area | Avail-Rad | Avail-Daylight | Avail-IntNoise | Avail-ExtNoise | Overall Available Allocations | Time [min] |
|---|---|---|---|---|---|---|---|---|
| 1 | 12 | 11 | 9 | 10 | 12 | 10 | 5 | 30 |
| 2 | 12 | 12 | 10 | 9 | 12 | 11 | 6 | 20 |
| 3 | 11 | 11 | 7 | 9 | 11 | 10 | 5 | 25 |
| 4 | 12 | 10 | 11 | 11 | 12 | 10 | 7 | 30 |

# 7 DISCUSSION

## 7.1 Conclusion

The study demonstrates how parametric data could be used in floor plan generation and optimization, plus how customer interaction could be integrated into this design process.

The plan generation algorithm considers the possibility of allocating diverse functions of rooms into separated spaces in the interior floor layout. In order to increase the efficiency in practice, the algorithm generates a mathematical tree store all the eligible plans, and uses pre-defined priority categories to limit the computation power in each run. The optimization procedure recommends the plans that best suffice the optimal user requirements comprehensively, of which the assessment is made based on fuzzy logic for quantitative features and situation-wise scoring dictionaries for descriptive items.

During the entire procedure, the customers have several opportunities to interact with the design. The most important information is the optimal value ranges for each criterion and the priority category of each function, which are given through a tangible tabletop connected with a database in the cloud. Besides, elective propositions from customers are also established in the interface for further requisites. For instance, the customer could check the suggestion of change the floor layout by moving non-structural walls, or they could have manual placement of particular functions. When making the final plan decision, they could also give their preferences for specific types of space criteria, by giving weights of the sufficiency scores to influence the final ranking. At the end of each iteration, the customers also have the opportunity to select the preferred plans directly, or apply some additional manipulation. Altogether, all of these interfaces ensure the algorithm to customize the plans as decisive as possible from the end user's request, which build up the core concept of the program.

When comparing the algorithm outcomes with the outcomes from experienced architects, it is shown that the plans from the architects would be able to locate more functions, yet some spaces' comfort condition is overestimated. Hence, the two advantages of the plan optimization algorithm could be indicated. For one thing, the algorithm shows better decisions while working on cognitive parameters by utilizing the simulation results. For another, the comprehensive plan generation scheme, which considers all the ranges of acceptance, could better guarantee the user experience in the long-term operation.

### 7.2 Future Study

Still, there are some limitations in the study. For instance, it is only implemented in the operating stage on the current stage, with all the separations and circulars pre-defined. Based on such constraints, the solution algorithm narrows down the sorts of cases that it could be further applied. Accordingly, it is suggested that some optimization methods for NP-hard problems might be further directions to broaden the scope of application. More importantly, more interactions in the early design phase could benefit more for the core idea of achieving high user involvement.

During the plan generation process, the algorithm emits re-simulation to save the reacting time. Nonetheless, the change of floor plans might alter the room characteristics such as noise sources, internal viewing, etc. Therefore, figuring out a reliable approximating process of update features could also help to improve the feasibility.

Last but not least, some accesses of user interactions are proposed in this algorithm, i.e. removing non-structural walls, manual allocation of function-space, and searching the spaces out of the original definition etc. These interactions have mock-ups of codes now but are not yet integrated into the program and the tangible user interface. Therefore, figuring out better suggestion methods and integration of the interactions should better support the achievement of the algorithm by enhancing user involvements in design.

## 8 ACKNOWLEDGEMENT

The authors want to thank the European Investment Bank for the funding of a grant for this research and the opportunity of this interesting case. Thanks also to the colleagues of LIST who participated in the comparative study.

## 9 REFERENCES

Bahrehmand, A. et al. (2017) 'Optimizing layout using spatial quality metrics and user preferences', Graphical Models, 93, pp. 25–38. doi: 10.1016/j.gmod.2017.08.003.

Bittermann, M. S., Ciftcioglu, Ö. and Sariyildiz, I. S. (2009) 'A cognitive system based on fuzzy information processing and multi-objective evolutionary algorithm', in 2009 IEEE Congress on Evolutionary Computation. Trondheim, pp. 1271–1280. doi: 10.1109/CEC.2009.4983091.

Boon, C. et al. (2015) 'Optimizing spatial adjacencies using evolutionary parametric tools: using grasshopper and galapagos to analyze, visualize, and improve complex architectural programming', Perkins + Will Research Journal, 7(2), pp. 25–37. Available at: https://pdfs.semanticscholar.org/ad0d/320d58c6375aaf572390fed24be06be195dd.pdf.

Brown, T. and Katz, B. (2011) 'Change by design', Journal of Product Innovation Management, 28(3), pp. 381–383. doi: 10.1111/j.1540-5885.2011.00806.x.

Chaharsooghi, S. K. A. H. M. K. (2008) 'An effective ant colony optimization algorithm (ACO) for multi-objective resource allocation problem (MORAP)', Applied Mathematics and Computation, 200(1), pp. 167–177. doi: 10.1016/j.amc.2007.09.070.

Chaouali, H. et al. (2018) 'Fuzzy logic optimization of a centralized energy management strategy for a hybrid PV/PEMFC system feeding a water pumping station', International Journal of Renewable Energy Research, 8(4), pp. 2190–2198. doi: 10.14569/IJACSA.2017.080839.

Daher, E., Kubicki, S. and Pak, B. (2018) 'Participation-based parametric design in early stages: A participative design process for spatial planning in office building', in the 36th eCAADe Conference. Lodz, pp. 429–438. Available at: https://lirias.kuleuven.be/retrieve/517143.

Das, S. (2017) 'Interactive artificial life based systems, augmenting design generation and evaluation by embedding expert a human machine dialogue for form finding', in the 35th eCAADe Conference, pp. 85–94. Available at: http://cumincad.scix.net/data/works/att/ecaade2018_438.pdf.

Dino, İ. G. (2012) 'Creative design exploration by parametric generative systems in architecture', Metu Journal of the Faculty of Architecture, 29(1), pp. 207–224. doi: 10.4305/METU.JFA.2012.1.12.

Dutta, K. and Sarthak, S. (2011) 'Architectural space planning using evolutionary computing approaches : a review', Artificial Intelligence Review, 36(4), pp. 311–321. doi: 10.1007/s10462-011-9217-y.

Garey, M. R., Graham, R. L. and Johnson, D. S. (1976) 'Some NP-complete geometric problems', in Proceedings of the eighth annual ACM symposium on Theory of computing, pp. 10–22. Available at: doi.org/10.1145/800113.803626.

Govil, K. (2011) 'A smart algorithm for dynamic task allocation for distributed processing environment', International Journal of Computer Applications, 28(2), pp. 13–19. doi: 10.5120/3362-4641.

Krish, S. (2011) 'A practical generative design method', Computer-Aided Design. Elsevier Ltd, 43(1), pp. 88–100. Available at: http://dx.doi.org/10.1016/j.cad.2010.09.009.

Kumar Singh, R., Singh, D. K. and Gangwar, S. (2018) 'Fuzzy logic optimization of process parameters affecting surface temperature in MAF', in IOP Conference Series: Materials Science and Engineering. Lucknow, pp. 12–17. doi: 10.1088/1757-899X/404/1/012017.

Lee, J., Gu, N. and Williams, A. P. (2014) 'Parametric design strategies for the generation of creative designs', International Journal of Architectural Computing, 12(3), pp. 263–282. doi: 10.1260/1478-0771.12.3.263.

Leonori, S., De Santis, E., Rizzi, A. and Mascioli, F. M. Frattale (2016) 'Multi objective optimization of a fuzzy logic controller for energy management in microgrids', in 2016 IEEE Congress on Evolutionary Computation (CEC). IEEE, pp. 319–326. doi: 10.1109/CEC.2016.7743811.

Leonori, S., De Santis, E., Rizzi, A. and Mascioli, F. M.Frattale (2016) 'Optimization of a microgrid energy management system based on a fuzzy logic controller', in IECON 2016 - 42nd Annual Conference of the IEEE Industrial Electronics Society. Florence, pp. 6615–6620. doi: 10.1109/IECON.2016.7793965.

Nagy, D. et al. (2017) ‘Project discover: An application of generative design for architectural space planning’, in Proceedings of the Symposium on Simulation for Architecture and Urban Design, pp. 1–8. doi: 10.22360/simaud.2017.simaud.007.

Pan, Y., Yu, Z. and Wang, L. (2003) ‘A genetic algorithm for the overlay multicast routing problem’, in Proceedings of the 2003 International Conference on Computer Networks and Mobile Computing. Shanghai, pp. 261–265. doi: 10.1109/ICCNMC.2003.1243054.

Shea, K., Aish, R. and Gourtovaia, M. (2005) ‘Towards integrated performance-driven generative design tools’, Automation in Construction, 14(2), pp. 253–264. doi: 10.1016/j.autcon.2004.07.002.

Stavrić, M. and Marina, O. (2011) ‘Parametric modeling for advanced architecture’, International Journal of Applied Mathematics and Informatics, 5(1), pp. 9–16. Available at: http://www.universitypress.org.uk/journals/ami/19-794.pdf.

Sun, T. (2004) A constrained steiner tree approach for reconstructions of multicast trees. The Chinese University of Hong Kong. Available at: https://core.ac.uk/download/pdf/48536443.pdf.

Villaggi, L. et al. (2018) Survey-based simulation of user satisfaction for generative design in architecture, Humanizing Digital Reality. Springer. doi: 10.1007/978-981-10-6611-5_36.

Zarei, Y. (2012) The Challenges of Parametric Design in Architecture Today: Mapping the Design Practice. The University of Manchester. Available at: https://search.proquest.com/docview/2066683436?accountid=27229.