Research Article

International Journal of Architectural Computing
1–17

DOI: 10.1177/1478077120943532
journals.sagepub.com/home/jac
SAGE

Using genetic algorithm to automate the generation of an open-plan office layout

Chen Chen[1], Ricardo Jose Chacón Vega[2] and Tiong Lee Kong[3]

Abstract

Today, the concept of open plan is more and more widely accepted that many companies have switched to open-plan offices. Their design is an issue in the scope of space layout planning. Although there are many professional architectural layout design software in the market, in the real life, office designers seldom use these tools because their license fees are usually expensive and using them to solve an open-plan office design is like using an overly powerful and expensive tool to fix a minor problem. Therefore, manual drafting through a trial and error process is most often used. This article attempts to propose a lightweight tool to automate open-plan office layout generation using a nested genetic algorithm optimization with two layers, where the inner layer algorithm is embedded in the outer one. The result is enhanced by a local search. The main objective is to maximize space utilization by maximizing the size of the open workspace. This approach is different from its precedents, in that the location search is conducted on a grid map rather than several pre-selected candidate locations. Consequently, the generated layout design presents a less rigid workstation arrangement, inviting a casual and unrestrictive work environment. The real potential of the approach is reflected in the productivity of test fits. Automating and simplifying the generation of layouts for test fits can tremendously decrease the amount of time and resources required to generate them. The experimental case study shows that the developed approach is powerful and effective, making it a totally automated process.

Keywords

Automated process, office design, genetic algorithm, open-plan office, space layout planning

Introduction

In recent years, the open-plan office design has been promoted and has become popular worldwide. Many Silicon Valley firms such as Facebook and Google favor it because it encourages collaboration, facilitates teamwork, and increases productivity. Compared to the traditional cellular or private office configurations,

[1]SJ-NTU Corporate Lab, Nanyang Technological University, Singapore
[2]CBRE Pte. Ltd (Singapore), Singapore
[3]School of Civil and Environmental Engineering, Nanyang Technological University, Singapore

Corresponding author:
Chen Chen, SJ-NTU Corporate Lab, Nanyang Technological University, ABN-B3a-02, 639798, Singapore.
Email: chenchen@ntu.edu.sg

open-plan offices are more advantageous for the employers because of lower building expenses due to reduced required partitions, lower rental cost due to increased worker density, better adjustability, and better access of daylight.[1] On the contrary, studies show that poorly designed open-plan offices lead to decreased productivity and issues with health due to terrible acoustic environments.[2] Fortunately, researchers found it is possible to ameliorate the negative environmental effects by offering the employees opportunities for quiet, concentrated work and privacy.[3,4]

Open-plan office design is considered within the scope of space layout planning. However, the space layout design is a classic (and difficult) problem that has two issues that need to be considered; one is the topological assignment of space elements and the other is the sizing of those space elements. Both encounter large numbers of combinations and permutations. The problem thus requires extensive computations. Over the past decades, several computer models have been developed by researchers in order to boost designers' productivity and ease this time-consuming task. Because space layout planning is difficult to be formulated and solved algorithmically, evolving approaches, especially genetic algorithms (GAs), have been popularly used for finding the optimal over the huge solution space.[5–9] GA is a heuristic search inspired by Charles Darwin's theory of natural evolution. It reflects the process of natural selection where the fittest individuals are selected for reproduction in order to produce offspring for the next generation. GAs are far more powerful and efficient than random search and exhaustive search algorithms, and thus, they have been widely used in many practical optimization problems.

This article attempts to propose a tool to automate open-plan office layout generation based on GA optimization. This tool is especially useful for the real estate and design consultants or corporate real estate teams to generate a test fit. Test fits are one of the most common, but also important, tasks done by real estate and design consultants as well as corporate real estate teams when choosing office spaces before any detailed design is done. A test fit is a floor plan used to confirm that the stated needs and requirements can be accommodated within a specific space. In an open-plan office test fit, the number and size in terms of dimensions and capacity of workstations, meeting rooms, training rooms, board rooms, open collaboration spaces, as well as support spaces (e.g. for a toilet, kitchen, and storage) will be determined. In practice, it probably takes a designer several hours or days to work out the best plan through a trial and error process.

Since the 1960s, researchers have devoted efforts to developing computer programs for the automated solution of spatial layout problems.[10] There are a lot of different kinds of planning approaches.[11] However, using tools\ developed for an architectural layout design that have a rather complicated interactive process for open-plan office design is like using an overly powerful and expensive tool to fix a minor problem. Therefore, this article aims to design a more efficient automated approach that can quickly generate a favorable layout design for open-plan offices with a less complicated, costly, and troublesome process. This study concentrates on office space configuration. This is the first step toward a more comprehensive method to totally automate the office design process including furniture selection. This article can contribute to the state of the practice by providing a lightweight tool for designers to complete the layout design task, particularly the test fit, in a shorter time.

Some researchers from Japan[8,12] proposed an office layout support system for polygonal space using GA. They generated room arrangement plans first, and then based on one selected plan, they generated the detailed layout. Meanwhile, some researchers from Australia[6,13] formulated the problem into a quadratic assignment problem (QAP) and solved it using GA. The problem has been finding a one-to-one mapping of the discrete set with m elements (set of activities, e.g., office facilities) onto another discrete set of n elements (set of locations, e.g., floors of the building where these facilities should be placed). However, the models of the Japanese researchers presumed that the rooms were exclusively along the walls and the models of the Australian researchers requested several pre-given room location candidates.

Different from the early approaches, this study finds the optimal room arrangement through a comprehensive search on a grid map. The grid size complies with the size of a single standard workstation. The biggest advantage of using a grid map is that the room locations are less constrained. Rooms, in theory, can be placed anywhere; consequently, the generated layout design presents a less rigid workstation arrangement, inviting a casual and unrestrictive work environment. It is believed that a more relaxed work environment would be more effective to increase inspiration, motivation, and ultimately productivity of the modern worker.

The scope of this study is limited to workspace arrangement. The following desk arrangement within this workspace can be made using the existing algorithms such as the procedural algorithms proposed by Anderson et al.[14] The main objective of the study is to maximize space utilization because space capacity is the most desirable question to inquire in a test fit. The objective can be achieved by maximizing the size of open workspace through assigning rooms to locations where the circulation area is minimized. The proposed solving method is a nested GA, where the inner layer algorithm is embedded in the outer one. It is hybridized with a local search to improve its performance. In short, this approach is a totally automated process. The inputs are the room size requirements and office space outline. The output is the office space layout plan associated with a large open workspace.

The rest of this article is organized as follows: after this introduction, the existing space layout planning approaches are described in Section "Space layout planning". Section "The problem formulation" elaborates on the problem model, and Section "The approach" describes the solving approach. Section "Design experiment" shows the experimental result. Finally, a conclusion is drawn in Section "Conclusion".

Space layout planning

When working on space planning problems, there are usually multiple aspects to consider, such as geometry of space, availability and location of emergency exits, lifts, stairs, and toilets, natural light sources, as well as the design requirements. In addition, because space can be represented in different ways, for example, discrete objects, area, and shapes, space planning problems have been solved by different approaches. In this section, the factors affecting the floor-plan layout design for buildings and the existing typical planning approaches are reviewed.

Considerations in space layout planning

The procedure to manually derive a space layout plan usually consists of two steps. First, starting from specification constraints, some sketches are drawn which represent space zoning or planning principles or topological feasible solutions with no precise geometrical dimensions. These provide an initial thinking of where the main working, collaboration, social, and support areas will lay. Next, geometrical dimensions are decided upon objective requirements. The objective requirements are expressed by constraints including dimensional constraints (e.g. constraints on the surface, length or width, and space orientation) and topological constraints (e.g. constraints on adjacency, non-adjacency, and proximity).[15]

Practitioners have summarized the factors affecting the floorplan layout design for buildings. These factors help the architects to refine the layout design. Lobos and Donath[11] recommended solar, views, accessibility, related functions, minimum distance, efficiency, size, geometric composition, golden ratio, shape, sustainability, and others as criteria and variables for office design. Gilbert[16] suggested using various relationship diagrams and the space requirements for people and tasks. Brooks[17] proposed the application of ergonomics to design office layouts, focusing on collecting the requirements, views, and feelings of all the stakeholders including managers, office staff, unions, and maintenance/cleaning personnel. Furthermore, Asefi et al.[18] introduced space syntax which is used in the field of spatial analysis for the evaluation of office layout plans.

Computer-aided layout planning systems

Researchers started to develop computerized systems for building space layout planning around half a century ago. The objectives and scale of the problems have varied widely, from the assignment of rooms on multiple floors of a building to the location of a desk in a single room. They can be either interactive or automated systems. Mitchell and Dillon[19] were the first researchers who brought the layout planning problem to the architectural design field. Currently, the research has fallen into two categories: the first is based on facility layout problems in operations research and industrial engineering applications, and the second is based on graph theory.[20] From a historical and architectural point of view, there are four main approaches: the expert systems, shape grammar, generative, and constraint-based.[11]

Automated space planning systems require methods that find optimal solutions among a large number of possible solutions and methods that allow designers to continually refine the problem definitions and solutions. The topological solution and geometrical solution are usually generated in a hierarchical way. Such hierarchical steps limit the combinatorial explosion of geometrical solutions and represent the architect's habits. Some approaches are generative in nature that they seek to produce as many design alternatives as possible, such as evolutionary approaches[7,21] and shape grammars.[22] An evolutionary approach is based on Darwin's theory to explore the search space with the help of the entire population of individuals. A shape grammar consists of shape rules and a generation engine that selects and processes rules. It is often assisted by space syntax measures which represent, quantify, and interpret spatial configuration.[23] Some approaches construct the problem into optimization models such as nonlinear programming.[15,24,25] Because constructing an optimization model requires lots of expertise and mathematical background, some researchers try to improve the initial layout by interchanging the space elements pair-wisely using a so-called "CRAFT" (computerized relative allocation of facilities technique) technique.[26] Others represent dynamic motion and changes in geometry by modeling space elements as mechanical elements that behave according to the laws of physics.[27,28]

The research gaps

The literature review shows that there are various kinds of approaches available to assist in architectural layout design. However, these approaches usually involve a rather complicated interactive process and using them for open-plan office design is like using an overly powerful and expensive tool to fix a minor problem. As in an open-plan office, the work seats are shared, and it is not necessary to define the topological relationship among rooms. As a result, the problem is simpler. Therefore, this study aims to design a lightweight tool compared to the existing tools, considering the special features of open-plan offices, and is an automated process. Solving the problem using an optimization method rather than an interactive method could greatly save time and energy, as there are vast numbers of combinations and permutations. It would save the designers troublesome trial-and-error processes and would allow them to focus on more valuable activities.

The problem formulation

The problem hereby presented is to design the open workspace configuration for a single floor office. The objective is to maximize its size by assigning rooms to locations where the circulation area is minimized. This section elaborates the problem formulation. Figure 1 illustrates the model plan with model constraints and structure. The approach consists of three steps. In the first step, given the office boundary and the grid size, the whole office area is tessellated. In the second step, given the room geometry information and the location of office exits, a GA is developed to find the best combination of room locations. Then, each combination of room locations is assessed in the third step, which involves another GA to minimize the circulation area. Finally, the plan with the minimum circulation area is selected as the final solution.

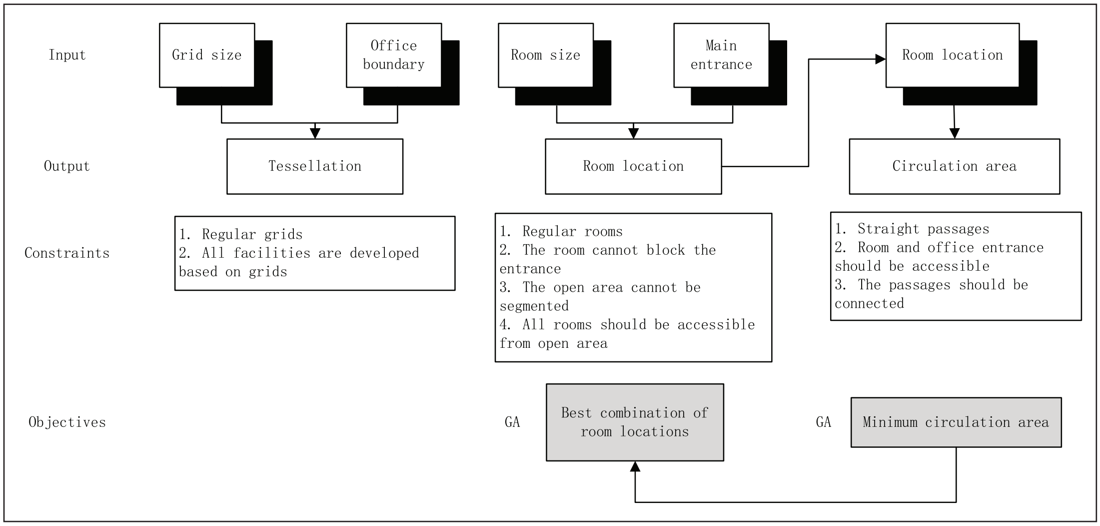

Figure 1. The problem model.

The main assumptions of the problem model are as follows:

1. The rooms are assumed to be in rectangular shapes.
2. Room geometries are given.
3. All the office facilities can be developed from rectangular grids.

The other constraints are as follows:

1. The exits cannot be blocked.
2. The open area cannot be segmented into isolated regions.
3. All rooms should be accessible from the open area.
4. Passageways are straight and they should be connected.

Grid map

Because the proposed approach is based on a grid map, the office space should first be discretized. The grid size must be properly chosen so that all office facilities can be developed from them. On the one hand, the computational cost is associated with the grid size, and using too small grids would greatly increase the computational cost. On the other hand, office furniture is usually modular and follows standardized dimensions. Therefore, it is recommended to choose the grid size based on the size of one workstation that will be used in this office. Note that grids can always be pre-reserved for specific usage in this approach, which implies their exclusions in the next room and main passageway construction.

Fast access to exits

Workplace safety and health guidelines require easy access to exits so that the exit access should not be blocked. In this regard, a constraint is applied that the grids facing the doors cannot be used for room construction. As such, the whole office space is segmented into several regions. People in each region can get fast access to exits in an emergency.

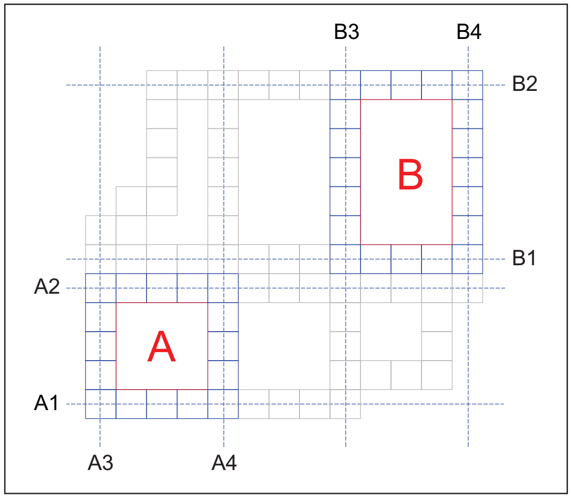

Figure 2. Passageway generation.

The heuristic for main passageway generation

Based on the grid map, the room arrangement problem is formulated into a many-to-one assignment problem. Then, for each combination of room locations, its corresponding circulation area is evaluated. The best room layout plan returns the smallest circulation area. Hence, the focus of the problem is placed on studying the circulation area. In an office environment, straight passageways are more desirable than bending ones. As their main purpose is to link the rooms, the main passageways should be along with one of the room sides (each room has four sides). It is possible to reach a passageway network by joining the trajectories of these room sides. As shown in Figure 2, rooms A and B can be connected by joining their trajectories, which are non-parallel, for example, {A1, B3}, {A3, B1}, {A1, B4}, {A2, B4}, {A3, B2}, {A4, B2}, or {A2, B1}. Among them, the shortest passageway is finally selected; in this case, it is {A2, B3}.

In addition to rooms, the passageway network must link to office exits, which means they should join with the exit accesses too. Furthermore, the passageway network should connect to the open workspace and the passageway network itself should be a simply connected space. If not, this passageway network is considered invalid.

The approach

In this section, the proposed approach is elaborated. The flowchart of the nested GA is shown in Figure 3, in which the inner layer GA is embedded in the outer one to provide the fitness values for the latter. The general procedure of a GA can be described as follows. To begin with, the parameters, including the population size, the maximum iteration number, the crossover and mutation probabilities, and the elitist ratio, are selected. The performance of a GA will be largely determined by these parameters. There is usually no universal parameter setting that can be drawn for all the problems; the best parameter combination is often found through computational experiments. Then, the initial population is randomly generated. In GAs, potential solutions to a problem are represented as a population of chromosomes. Each chromosome is evaluated and given a fitness value, according to which, the chromosomes are ranked. From the initial population, the GA carries out an evolutionary process through fitness-based selection, crossover operation, and mutation operation. The new population for the successive generation is produced by combining the elitists in the previous

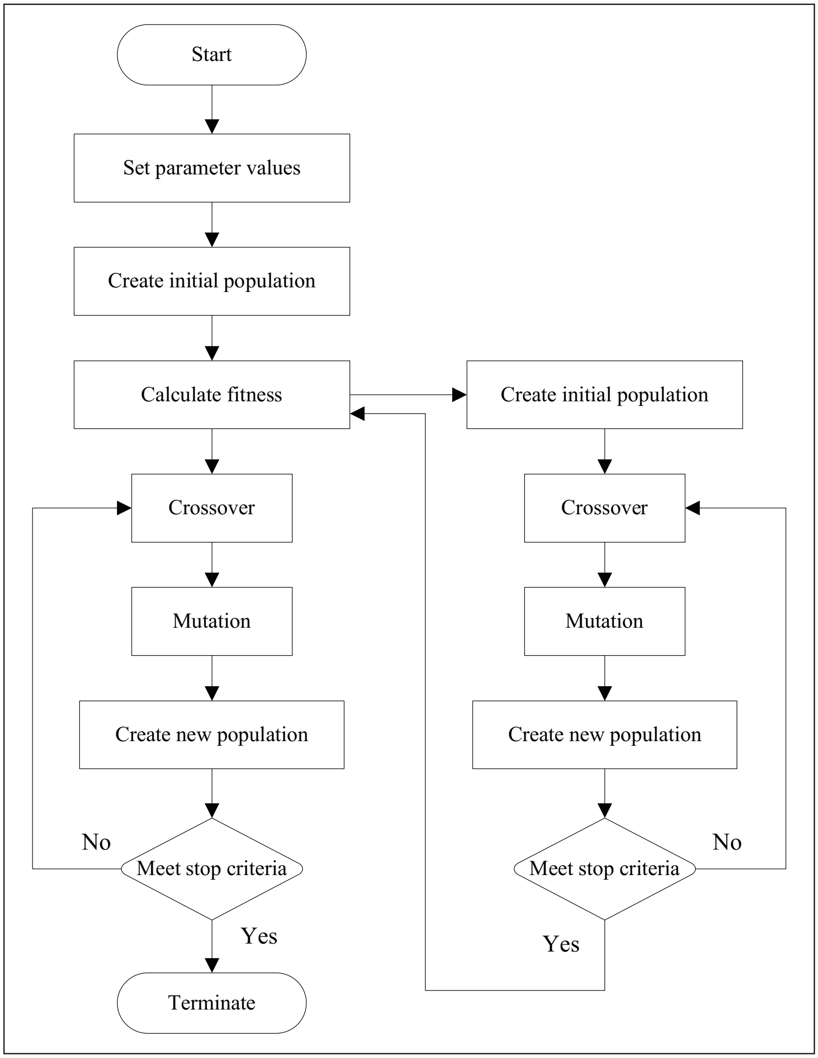

Figure 3. Flowchart of the nested genetic algorithm.

population and the children after the evolutionary process. The above iterative process will not cease until the stopping criterion is reached. GA can quickly locate high-performance regions of vast and complex search spaces, but it is not well-suited for fine-tuning solutions, which are very close to optimal ones. Hence, a local search is continued following GA search procedure to discover the optimal ones.

Outer layer GA

The outer layer GA is for room location arrangement. The objective is to find the minimum fitness value. Its details are explained below.

Chromosome representation. First, all feasible locations of a room are discovered and kept in a list. The chromosome is designed in an integer form. Then, the position of a digit in the chromosome refers to a room and each digit refers to an index in the list pointing to a feasible location of that room. A sample chromosome is given below:

Chromosome	3	5	6	3	4	7	9	1	10
Room	1	2	3	4	5	6	7	8	9

Because one grid cannot be assigned to more than one room (otherwise, the rooms will be overlapped), we will apart those overlapped rooms by selecting other locations for them.

Selection. A tournament selection operator is used for parent selection. If the tournament size is larger, weak individuals have a smaller chance to be selected because if a weak individual is selected to be in a tournament, there is a higher probability that a stronger individual is also in that tournament. Hence, the tournament size is selected to be 2, which is a very popular selection.

Crossover. The single-point crossover method is used, and it happens with a given probability $P_{crossover}$. The digits of two parent chromosomes are exchanged from a randomly chosen crossover point to generate two child chromosomes.

Mutation. The mutation operator performs on the child solutions after crossover. It alters one digit in the chromosome. Similarly, a mutation happens according to a given probability $P_{mutation}$.

Creation of the next generation. A small portion of the fittest candidates are copied into the next generation. Consequently, the new population is contributed by the children after crossover and mutation and these elitists.

Fitness calculation. The fitness value is obtained from the return of the inner layer GA.

Termination criteria. The search terminates after reaching a certain number of iterations.

Inner layer GA

The inner layer GA is for main passageway arrangement based on the room arrangement. If the chromosome of the outer layer GA is found infeasible, the inner one will automatically abort. The procedure of the inner one is similar to that of the outer one. The same strategies are applied for selection, crossover, mutation, creation of the next generation, and termination. The objective is to find the minimum fitness value.

Chromosome representation. Using the heuristic described in Figure 2, the passageway network can be easily created by joining the trajectories of room sides and the trajectories of exits. Each room has four sides: down, up, left, and right. We use digits 1 to 4 to denote them. Accordingly, an integer chromosome is designed. The position of a digit in the chromosome points to a room, and each digit refers to a room side trajectory. When there is no grid along a trajectory, the trajectory is considered infeasible and removed from the pool for gene selection. A sample chromosome is shown below:

Chromosome	1	3	4	2	3	4	2	3	1	1
Room	1	2	3	4	5	6	7	8	9	10
Trajectory	down	left	right	up	left	right	up	left	Down	down

Fitness calculation. The fitness value is simply the total number of grids that participate in the construction of the main passageways. Consequently, it becomes very easy to calculate the total workspace size, which is the total number of grids subtracting the grids that are reserved for rooms and main passageways. The workspace size implies the office capacity, which is the most desirable question of inquiry in a test fit.

Strategies to deal with infeasible solutions

In the outer layer GA, the chromosome should comply with the following constraints: (1) one grid cannot be assigned to more than one room, and (2) the rooms cannot segment the open workspace into isolated regions. In the inner layer GA, the constraints are as follows: (1) the passageway network should connect to the open workspace, and (2) the passageway network itself should be a simply connected space. If any of these constraints are violated during evolution, the chromosomes become infeasible. Because the main passageway design depends on the room locations, in case the chromosome of the outer layer GA is infeasible, there is no need to continue to proceed to the inner one. As the randomly generated chromosomes and the child chromosomes after crossover and mutation do not always satisfy the constraints, we make extra effort to correct them, but they are limited to those suffering from the overlapping problem for the sake of computational time reduction. We developed two strategies to deal with the remaining infeasible ones.

Strategy I—Giving the infeasible ones very large fitness values to reduce their possibilities of being selected as solutions.

Strategy II—Dropping off the infeasible ones and replacing them with another randomly generated feasible ones.

Local search for further improvement

Local search maintains the current solution and explores the search space by steps within its neighborhood to seek for a better one. In this approach, a better solution is found by perturbing the room locations and rotating the rooms. A solution is considered better if it can either reduce the main passageways or create a larger continuous workspace area.

Design experiment

This section illustrates an experimental case study to demonstrate the performance of the developed open-plan office design algorithm. The algorithm was coded in C# language and tested on a laptop computer with a processor of Intel (R) Core (TM) i7-8750 H CPU@2.20 GHz and 32.0 GB installed memory (RAM).

Case information

Figure 4 shows the existing plan for an open-plan office, and its grid map is as illustrated in Figure 5. The grid size is $1900\,\text{mm} \times 2000\,\text{mm}$, which is approximately the dimension of a workstation in this plan. The room dimension requirements are given in Table 1. Because the office exits are in Grids 108, 156, 195, 211, and 84, Grids {101, 102, 103, 104, 105, 106, 107, 108}, {149, 150, 151, 152, 153, 154, 155, 156}, {188, 189, 190, 191, 192, 193, 194, 195}, {211, 227, 243, 259, 281, 303, 325, 347, 369}, and {71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84} were excluded from room construction.

Results

The performance of a GA depends on the setting of its parameters. Usually, there is no universal parameter setting that fits all problems. The appropriate parameter setting can be found only from a trial-and-error

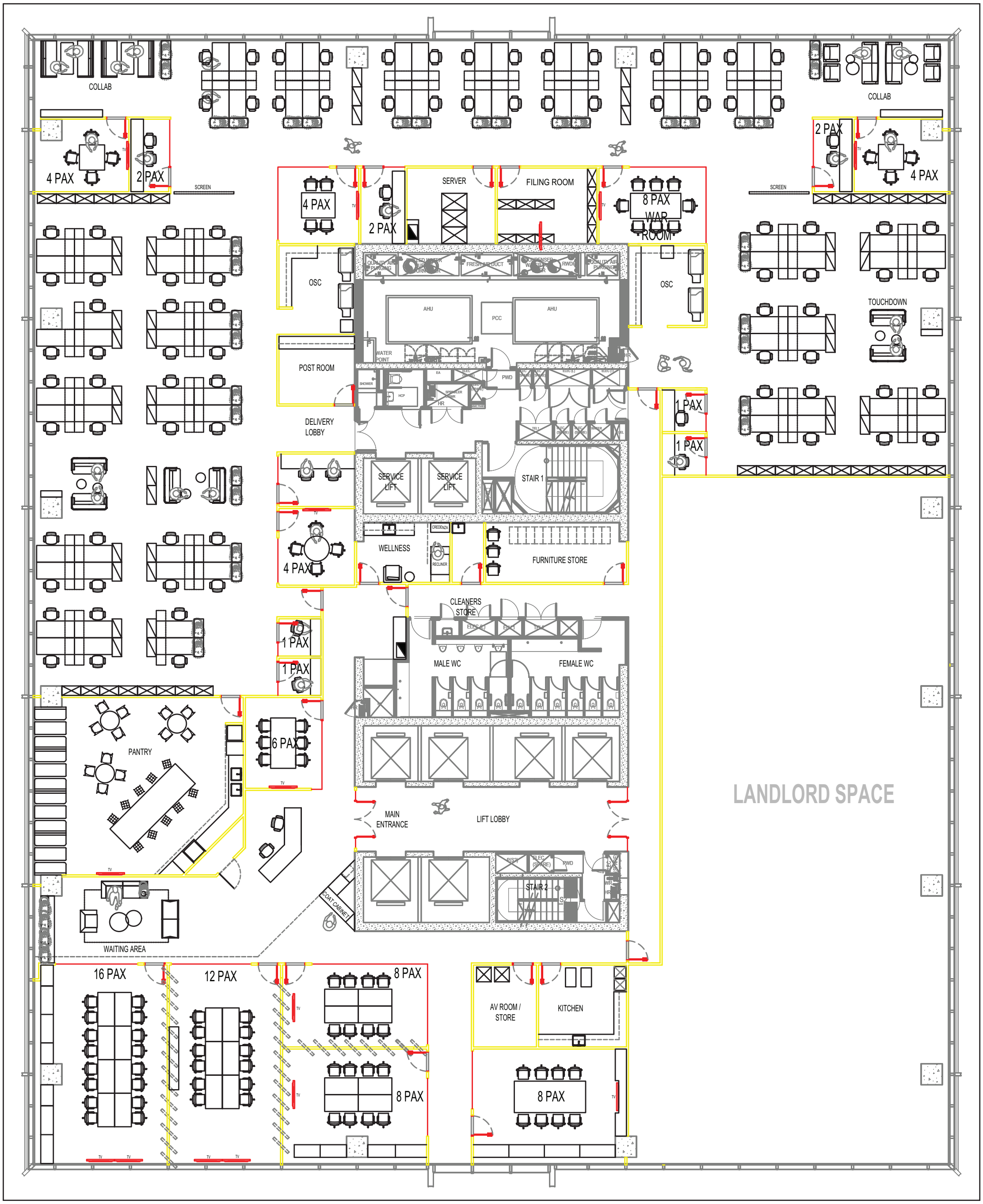

Figure 4. The existing plan.

method. A previous study[29] suggests that crossover has a high probability, typically 0.8–0.95, while mutation has a low probability, typically 0.05–0.1. In this regard, this study chose the crossover rate as 0.8 and the mutation rate as 0.1. In addition, the elitist ratio is chosen as 0.4, which is a random choice to start with.

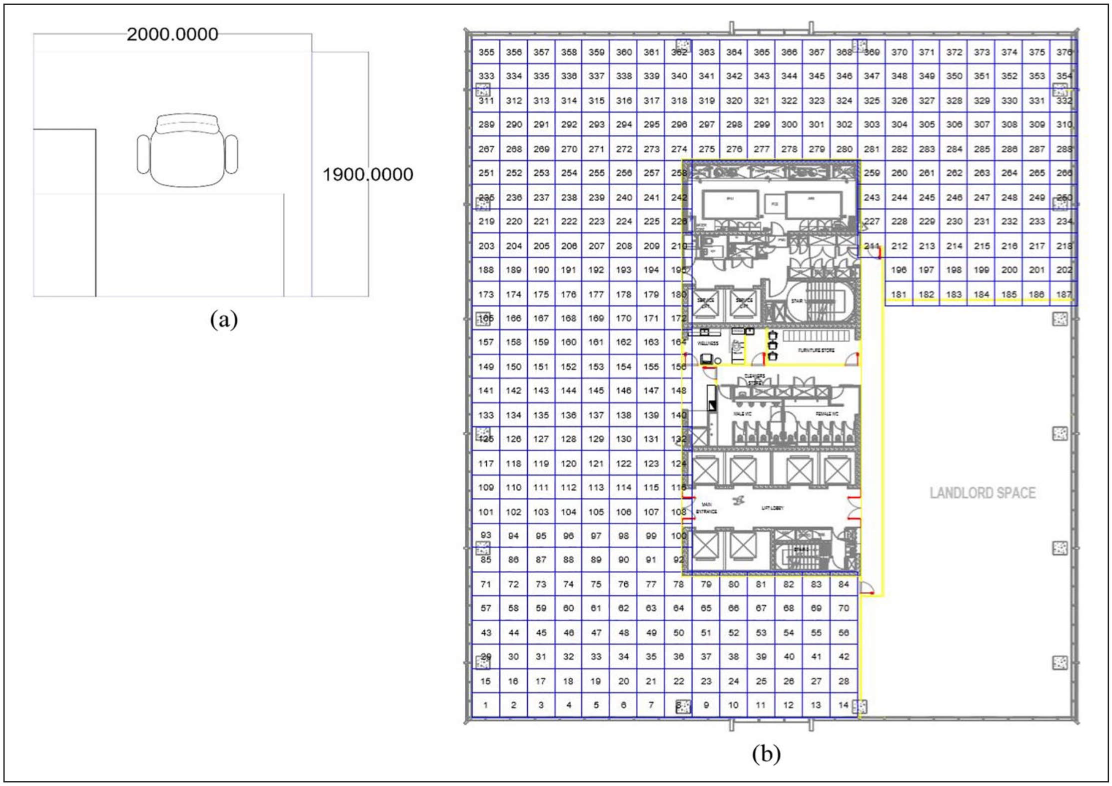

Figure 5. Creation of the grid map: (a) grid design and (b) tessellation.

Table 1. Room requirement.

Room	Quantity	Dimension requirement (grids × grids)
16-PAX meeting room	1	3×5 or 5×3
12-PAX meeting room	1	3×5 or 5×3
8-PAX meeting room	2	3×3
8-PAX meeting room	2	3×2 or 2×3
6-PAX meeting room	1	2×3 or 3×2
4-PAX meeting room	4	2×2
2-PAX meeting room	3	2×1 or 1×2
1-PAX meeting room	4	1×1
Others	3	2×1 or 1×2

The population size and iteration times have the biggest effects on the solution quality and computational cost. Generally, larger population sizes and longer iterations result in better solutions but suffer from higher computational costs (see Table 2). There is a tradeoff between solution quality and computational cost. Table 3 shows that using Strategy I and using Strategy II have no big difference. Figure 6 presents the corresponding convergence line, which shows the developed nested GA approach is very efficient at the beginning. Figure 7 gives an example to illustrate the effect of the local search. It shows the local search can substantially improve

Table 2. Comparison of computation time and result quality using different population size and iterations.

Case	Outer layer GA		Inner layer GA		Strategy I	
	Population size	Iteration	Population size	Iteration	Computation time (h)	Best fitness
1	50	50	50	50	2.4	64
2	50	100	50	50	4.9	64
3	50	200	50	50	9.4	56
4	100	50	50	50	9.3	63
5	100	100	50	50	25.3	62
6	100	200	50	50	47.2	50
7	200	50	50	50	30.2	55
8	50	50	50	100	6.4	59
9	50	100	50	100	9.4	57
10	100	50	50	100	18.2	62
11	100	100	50	100	42.5	57
12	50	50	100	50	3.6	63
13	50	100	100	50	3.8	58
14	100	50	100	50	9.9	54
15	100	100	100	50	14.2	53
16	100	200	100	50	56.2	56
17	50	50	100	100	4.7	65
18	50	100	100	100	14.6	61
19	100	50	100	100	22.6	55
20	100	100	100	100	40.8	55

GA: genetic algorithm.

Table 3. Results of 10 tries using population = 50 and iteration times = 50.

Run	Strategy I		Strategy II	
	Computation time (h)	Best fitness	Computation time (h)	Best fitness
1st	2.7	63	2.5	67
2nd	3.4	61	3.4	61
3rd	3.1	61	2.5	56
4th	2.3	71	2.7	62
5th	2.6	63	3.1	62
6th	2.1	59	2.8	68
7th	2.8	56	3.4	61
8th	2.2	62	2.3	69
9th	2.3	68	3.7	74
10th	2.4	64	3.8	59
Average	2.59	62.8	3.02	63.9

the GA result. As the computational cost is quite high for this embedded GA, hybridization with the local search can significantly reduce the computational cost. Figure 8 provides the generated optimal layout benchmarking against the original plan.

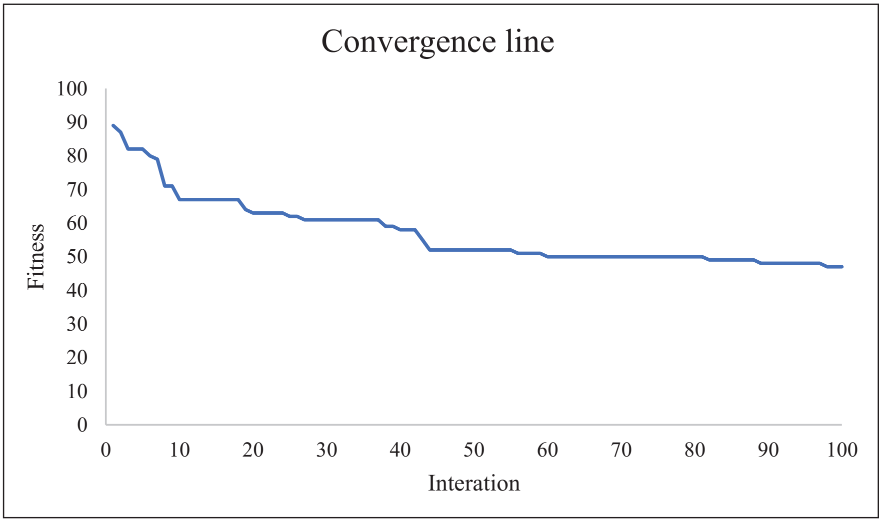

Figure 6. Convergence line.

Discussions

Since the optimization objective is to search for the smallest main passageways, the rooms tend to locate near the exit accesses. In this case, the elevator lobby is in the middle, so the rooms are attracted to the central lobby area, leaving the corner area to the workspace. In modern buildings, corner windows make for the most panoramic views. Users seating in these corners enjoy not only better external views but also more natural sunlight. Besides, the passageway network is dense in the GA result, which implies high space utilization. Moreover, we purposely leave space in front of an exit unobstructed by rooms enabling fast retreat for users in an emergency. Generally, there is no best solution in architecture design. All designs are the result of a compromise of multiple factors. Therefore, once the computation is done, it still gives the designers opportunities to modify the result and exchange areas used for workstations for alternative activities, such as pantries, reception, open collaboration, and IT infrastructure support.

GA is simple to implement. Nevertheless, GA has some drawbacks. The biggest problem of GA is its premature problem. It has a tendency to converge toward local optimum rather than the global optimum of the problem. This problem may be alleviated by maintaining a diverse population of solutions. In the experimental case, we investigated two strategies to treat infeasible solutions. One strategy keeps the infeasible solutions, whereas the other drops them. However, we did not discover the benefit of maintaining a diverse population by continuously adding new random chromosomes during evolution. There is no big difference between the results from using the two different strategies. The reason might be that some infeasible solutions contain good genes and keeping them creates opportunity for good results. Also, we found it is more effective to hybridize the GA with a local search. Although GA explores the search area effectively and converges very fast at the beginning, it is slow to locate the exact optimum at the later stage. Hence, a local search enhances the GA and the hybrid GA is much better than the pure GA, especially when it has an embedded iterative process.

This research gives an optimization approach for designers to create an open-plan initial layout. However, in architectural design, there are many known, partially known, successively revealed, and loose constraints, leading to the conditions of wicked problems. Algorithms have difficulty even to understand all these constraints, but only human beings have the capability to synthesize them. For algorithm development, there is

Figure 7. Improvement by local search: (a) Case I $f = 56$ and $f = 49$ (b) Case II $f = 45$ and $f = 45$. (* refers to the adjusted rooms).

still a long way to go. As the initial settings all depend on designers, this research hopefully can help designers at least to reduce some repetitive work.

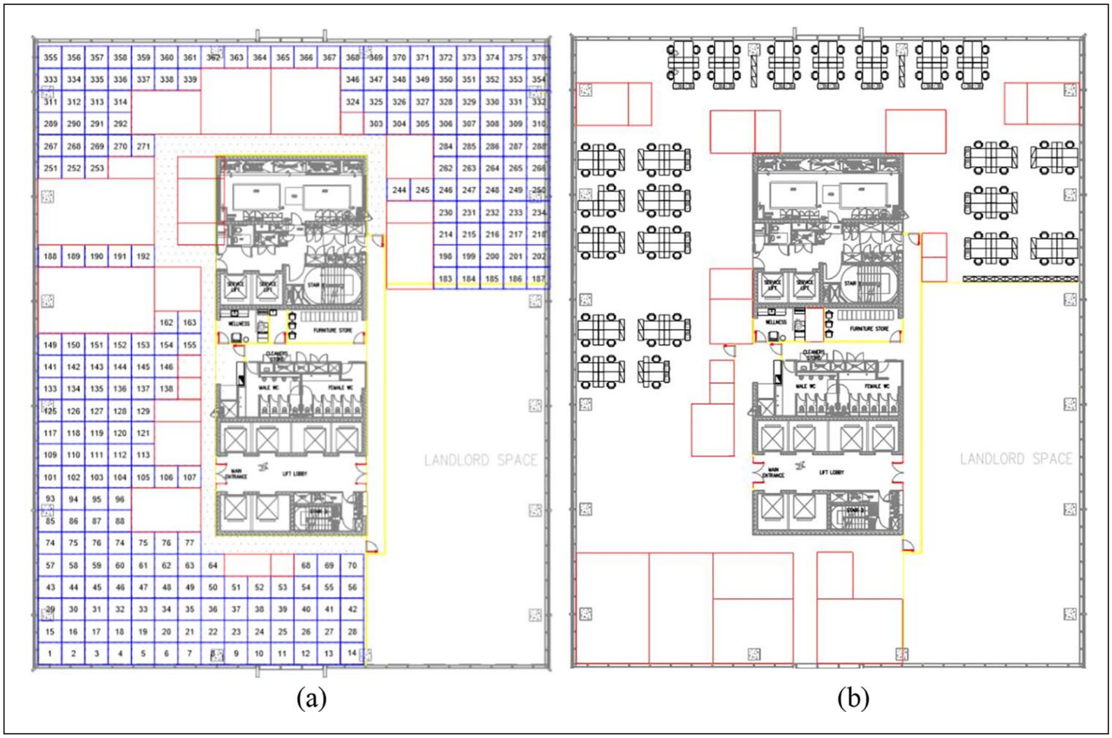

Figure 8. Comparison of the genetic algorithm (GA) result and the original plan: (a) GA result and (b) original plan.

Conclusion

This article studied the space configuration for a single-floor open-plan office using an approach based on grids. A nested GA with two layers is developed to solve the problem, where the inner layer algorithm is embedded in the outer one. The objective is to maximize space utilization by maximizing the size of the open workspace. The size of the open workspace is maximized by minimizing the passageway network.

The problem of office design involves a lot of objectives and constraints. Solving such a problem is not an easy task in view of a large number of combinations and permutations. The existing approaches often reduce the problem size by pre-selecting several room location regions. The current approach is different from them, in that it conducted an exhaustive room location search in the whole office area by establishing a grid map model. The result is more optimal and less regimented.

GA is used to solve optimization problems. As previously mentioned, GA has been popularly used by researchers to solve the space design problem. It is an efficient approach that can figure out the best one in an explosively large number of alternatives in a short time. Compared to the other approaches used in architectural design, this is a lightweight tool, less complicated, costly, and troublesome.

It offers an automated process. The real potential of this approach is especially reflected in the productivity of test fits that traditionally the designers have to do "by hand." Automating and simplifying the generation of layouts for test fits can tremendously decrease the amount of time and resources required to generate them, and moreover, also contribute to determining, mathematically, the optimal solution, given certain requirements.

In the future, further studies will be conducted to refine the designs considering multiple work settings and multiple activity-based spaces. We aim to provide the designers with a comprehensive automated tool for layout design processes as well as a comprehensive evaluation tool to compare multiple design alternatives.

Acknowledgements

The authors thank the editor and the anonymous reviewers for their careful reading of our manuscript and their many insightful comments and suggestions.

Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) received no financial support for the research, authorship, and/or publication of this article.

ORCID iD

Chen Chen https://orcid.org/0000-0003-4114-9074

References

1. Shafaghat A, Keyvanfar A, Ferwati MS, et al. Enhancing staff's satisfaction with comfort toward productivity by sustainable open plan office design. *Sustain Cities Soc* 2015; 19: 151–164.
2. Smith-Jackson T, Middlebrooks R, Francis J, et al. Open plan offices as sociotechnical systems: what matters and to whom? *Works* 2016; 54(4): 807–823.
3. Haapakangas A, Hongito V, Varjo J, et al. Benefits of quiet workspaces in open—plan offices—Evidence from two office relocations. *J Environ Psychol* 2018; 56: 63–75.
4. Morrison RL and Smollan RK. Open plan office space? If you'are going to do it, do it right: a fourteen-month longitudinal case study. *Appl Ergonom* 2020; 82: 102933.
5. Jo JH and Gero JS. A genetic search approach to space layout planning. *Arch Sci Rev* 1995; 38(1): 37–46.
6. Gero JS and Kazakov VA. Learning and re-using information in space layout planning problems using genetic engineering. *Artif Intell Eng* 1997; 11(3): 329–334.
7. Elezkurtaj T and Franck G. *Algorithmic support of creative architectural design*. Weimar, Germany: Umbau 19, 2002, pp. 129–137.
8. Araki Y and Osana Y. Office layout support system for polygonal space using interactive genetic algorithm. In: *Proceedings of the 2012 IEEE international conference on systems, man, and cybernetics (SMC)*, Seoul, Korea, 14–17 October 2012.
9. Dino IG and Űçoluk G. Multiobjective design optimization of building space layout, energy, and daylighting performance. *J Comput Civil Eng* 2017; 31(5): 04017025.
10. Liggett RS. Automated facilities layout: past, present and future. *Automat Constr* 2000; 9(2): 197–215
11. Lobos D and Donath D. The problem of space layout in architecture: a survey and reflections. *Arquitetura Revista* 2010; 6(2): 136–161.
12. Tachikawa R and Osana Y. Office layout support system using genetic algorithm—generation of room arrangement plans for polygonal space. In: *Proceedings of the 2010 second world congress on nature and biologically inspired computing (NaBIC)*, Fukuoka, Japan, 15–17 December 2010.
13. Jagielski R and Gero JS. A genetic programming approach to the space layout planning problem. *CAAD Futures* 1997; 1997: 875–884.
14. Anderson C, Beiley C, Heumann A, et al. Augmented space planning: using procedural generation to automate desk layouts. *Int J Arch Comput* 2018; 16(2): 164–177.
15. Medjdoub B and Yannou B. Separating topology and geometry in space planning. *Comput-aided Design* 2000; 32(1): 39–61.
16. Gilbert JP. Construction office design with systematic layout planning. In: *Proceedings of the 2nd world conference on POM 15th annual POM conference*, Cancun, Mexico, 30 April–3 May 2004.
17. Brooks A. Ergonomic approaches to office layout and space planning. *Facilities* 1998; 16(3/4): 73–78.

18. Asefi M, Haghparast F and Sharifi E. Comparative study of the factors affecting the generativity of office spaces. *Front Arch Res* 2019; 8(1): 106–119.
19. Mitchell WJ and Dillon R. A polynomial assembly procedure for architectural floor planning. In: *Proceedings of the 3rd environmental design research association conference*, Los Angeles, CA, 24–27 January 1972.
20. Nassar K. New advances in the automated architectural space plan layout problem. In: *Proceedings of the international conference in computing in civil and building engineering*, Nottingham, UK, 2010.
21. Calixto V and Celani G. A literature review for space planning optimization using an evolutionary algorithm approach: 1992-2014. *XIX Congresso Da Sociedade Ibero-americana De Gráfica Digital* 2015; 2(3): 662–671.
22. Steadman P. Generative design methods and the exploration of worlds of formal possibility. *Arch Design* 2014; 84(5): 24–31.
23. Hillier B, Hanson J and Graham H. Ideas are in things: an application of the space syntax method to discovering house genotypes. *Env Plann B Plann Design* 1987; 14(4): 363–385.
24. Li S-P, Frazer JH and Tang M-X. A constraint based generative system for floor layouts. In: *Proceedings of the Fifth Conference on Computer Aided Architectural Design Research in Asia (CAADRIA 2000)*, Singapore, 18–19 May 2000, pp. 441–450. Singapore: CuminCAD.
25. Michalek JJ, Choudhary R and Papalambros PY. Architectural layout design optmization. *Eng Opt* 2002; 34(5): 461–484.
26. Buffa ES, Armour GC and Vollmann TE. Allocating facilities with CRAFT. *Havard Bus Rev* 1964; 42(2): 136–158.
27. Arvin SA and House DH. Making designs come alive: using physically based modeling techniques in space layout planning. In: Augenbroe G and Eastman C (eds) *Computers in building*. Boston, MA: Springer, 1999, pp. 245–262.
28. Arvin SA and House DH. Modeling architectural design objectives in physically based space planning. *Automat Const* 2002; 11(2): 213–225.
29. Grefenstette JJ. Optimization of control parameters for genetic algorithms. *IEEE Trans Syst Man Cybern* 1986; 16(1): 122–128.