A Heuristic Approach for the Automated Generation of Furniture Layout Schemes in Residential Spaces

Sherif Abdelmohsen, Ayman Assem, Sherif Tarabishy and Ahmed Ibrahim

Abstract A variety of heuristic methods and algorithms have been developed for space layout planning problems. Recent efforts to generate furniture layout schemes in existing spatial configurations have mostly relied on exhaustive search and are likely to produce dysfunctional or counter-intuitive solutions. In this paper, we propose a heuristic approach for the automated generation of furniture layout schemes, with specific focus on residential spaces. First, we present an operational definition for furniture entities, space configurations, and space entities. Then we introduce a heuristic algorithm for generating furniture layout schemes based on a set of space subdivision rules, object-object relations, and object-space relations. Using Grasshopper, we generate a group of possible schemes for a sample residential living space. A discussion follows, outlining current limitations, expanding the context of the study, and possibilities for development.

Introduction

As a subset of space layout planning, furniture layout design involves a continuous process of divergence and convergence of solution space in order to achieve maximum diversity of alternatives along with informed decision making and near optimum solutions. Benchmarking the quality of layout schemes and design alternatives requires a balance between exhausting a plethora of possibilities, and achieving rational optimality.

Furniture layout organization as a task is traditionally similar to the problem of placing an object (with complex geometry) in space (implying topological

S. Abdelmohsen (✉) · S. Tarabishy
The American University in Cairo, Cairo, Egypt
e-mail: sherifmorad@aucegypt.edu

S. Abdelmohsen · A. Assem
Ain Shams University, Cairo, Egypt

S. Abdelmohsen · A. Assem · S. Tarabishy · A. Ibrahim
UNii Engineering Consultancy, Cairo, Egypt

J.S. Gero (ed.), *Design Computing and Cognition '16*,
DOI 10.1007/978-3-319-44989-0_25

relations). This problem, according to Flemming et al. (1988), implies an approach that aims at systematically enumerating alternative solutions while simultaneously considering a wide spectrum of criteria; both beyond human cognition. Some of the main challenges in this problem involve the need for efficient search strategies while dealing with the infinite solution space resulting from the inherent growing geometrical complexity (Flemming et al. 1988), and the need to associate with the natural design process of an architect while attempting to address basic topological solutions (Medjdoub and Yannou 2000).

Research addressing the automation of furniture layout schemes in existing spatial configurations stems from two lines of work; (1) facility layout and space layout planning, and (2) furniture layout optimization based on interior design guidelines. In essence, we approach the furniture layout problem as a space planning problem, where space boundary is perceived as a container for subzones that hold furniture arrangements.

A variety of heuristic methods have been developed to address space layout planning problems, including greedy algorithms (Boswell 1992; Ahuja et al. 2000), branch and bound methods (Kim and Kim 1999; Xie and Sahinidis 2008), dynamic programming (Rosenblatt 1986), and single-solution metaheuristic methods such as tabu search (Abdinnour-Helm and Hadley 2000; Chiang and Kouvelis 1996). Some recent efforts to generate furniture layout schemes in existing configurations (Kjølaas 2000; Akazawa et al. 2005; Germer and Schwarz 2009; Larive et al. 2004; Sanchez et al. 2003; Merrell et al. 2011; Yu et al. 2011) have adopted an exhaustive search process that is likely to produce illogical or uninhabitable arrangements. Others focused on merely ergonomic factors. Some others relied on object-object relations without much attention to the analysis of space boundaries, or required manual user intervention.

In this paper, we propose a heuristic approach for the automated generation of furniture layout schemes that involve a thorough analysis of spatial configurations and furniture objects, object-object relations, and space-object relations, to produce a habitable layout scheme. We identify a set of rules for the logical and intuitive placement and arrangement of furniture objects in a given space based on these relations. We claim that the resulting range of possible furniture layout schemes satisfy an intuitive furniture layout process, without the need for an exhaustive search through all possible—and perhaps likely dysfunctional—solutions. Our goal is to emulate how architects would perceive a given spatial configuration, analyze it and propose basic habitable furniture layout schemes that are diverse enough and at the same time meet the existing spatial constraints and conditions.

First, we present an operational definition for furniture entities, and space configurations and entities. Then we introduce a heuristic algorithm for generating furniture layout schemes based on space subdivision rules, object-object relations, and object-space relations. We use Grasshopper to generate furniture layout schemes for a sample residential living space.

Related Work

A variety of approaches have been proposed for space layout planning that apply for our context of study. The graph theory approach is one example, where form emerges from the arrangement of spaces in a planar graph. Another approach involves quadratic assignment problems, where entities are assigned to cells in a matrix representation (Balakrishnan and Cheng 2000; El-Baz 2004). Contrary to space emergence approaches, we focus on methods that employ the generation of spaces or partial spaces within a fixed boundary. Developed originally within the facility planning realm, the slicing tree approach is one of the early methods that attempt to continually subdivide a given space through horizontal and vertical lines to obtain new subzones (Tam 1998; Azadivar and Wang 2000; Al-Hakim 2000; Wu and Appleton 2002; Shayan and Chittilappilly 2004; Honiden 2004; Aiello et al. 2006; Banerjee et al. 2008; Aiello et al. 2012). In this approach, terminal nodes represent spaces and internal nodes represent the vertical and horizontal subdivision of the children nodes.

Following the slicing tree approach which was limited in its configuration geometry, other approaches emerged such as combining the space filling curves (SFC) technique with genetic algorithms, where a continuous path is defined that generates space on a matrix (Buscher et al. 2014; Hu et al. 2007; Islier 1998; Kochhar et al. 1998; Wang et al. 2005). Others used evolutionary techniques to solve the planning of departmental spaces (Dunker et al. 2003), where fixed or flexible spatial blocks are allocated within a certain boundary. Other factors were considered in later approaches like hierarchical organization of layout elements (Koenig and Schneider 2012), calculation of distances between departments through aisles using graph algorithms and genetic algorithms (Lee et al. 2005), and multi-level space allocation using hybrid evolutionary techniques (Rodrigues et al. 2013).

An important factor to consider is how architects attempt to visualize and subdivide spaces to establish possible layout configurations. According to Indraprastha and Shinozaki (2011), space boundaries and inter-relationships between different architectural elements and planned activities constitute an essential component of space composition. In their method, architectural space is composed of subdivided enclosed territorial spaces defined by internal circulation paths, and each of these spaces has a set of distinct physical properties that are impacted by different elements.

Early attempts to generate furniture layouts include the work by Kjølaas (2000), where functional space is represented as a nested hierarchy of rectangular templates that are swapped by eight predetermined mutation functions, while free space is represented using empty boxes in front of doors and windows. This approach is limited however to strictly rectangular configurations. A later approach introduced parent-child relationships between furniture objects and used a semantic database to store these relations but with manual control on inter-object relations and constraints (Akazawa et al. 2005). Germer and Schwarz (2009) introduced an agent-based approach, where each furniture object was viewed as an agent seeking to attach to a parent object. Parent-child relationships of each object had to be manually defined.

Merrell et al. (2011) introduced a furniture layout system that generates proposed arrangements interactively according to a set of developed interior design guidelines, such as balance, alignment, emphasis and conversation. Yu et al. (2011) introduced an automated furniture layout synthesis approach using realistic indoor scenes, whereby they considered human factors including visibility, constraints, accessibility, and pathways. While these approaches focused on ergonomics and studied further aspects of visibility and accessibility, they tend to conduct an exhaustive search for all possible schemes regardless of spatial analysis or space-object relations, and generate solutions that may be counter-intuitive to how architects perceive space and its closely coupled relation with furniture layouts to produce habitable space. Our approach attempts to address this gap, and adopts a heuristic approach to automatically generate basic furniture layout schemes in a given spatial configuration, taking into consideration an in-depth analysis of space entities, furniture entities, and space-object relations.

Approach

In order to study furniture arrangement within a given space, we first identified a number of possible configurations and space boundaries within which furniture groups are to be positioned. Below are some possible families of space boundaries and their potentials and constraints (Fig. 1).

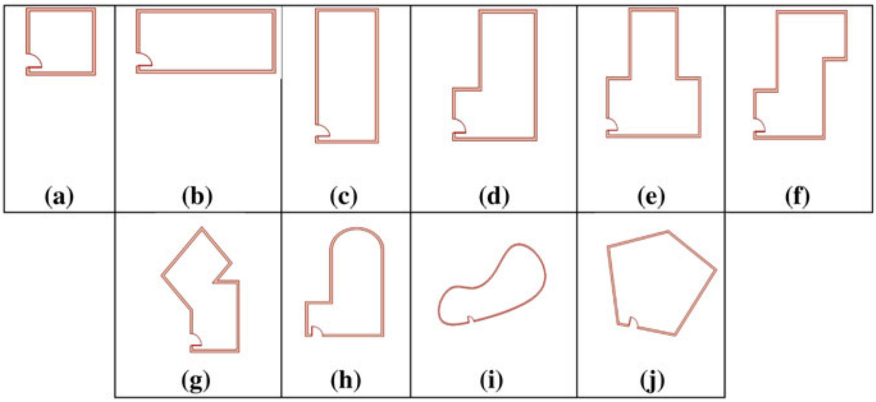


Fig. 1 Possible space boundaries for furniture arrangement: **a**–**c** single rectangular boundary, **d**–**f** clustered rectangular boundary, **g** multi-grid boundary, **h** circular boundary, **i** organic boundary, **j** angled boundary

1. Regular Space Boundary
   (a) Single rectangular boundary: Our assumption is that the space boundary in this configuration is the basic envelope for the bounding box of a single furniture group (e.g. sofa and two armchairs, dining table with chairs, etc.), or multiple groups, depending on space proportion. For example, a 4 m × 4 m configuration is assumed to host a single furniture group, while a 6 m × 8 m configuration is assumed to host two or more furniture groups.
   (b) Clustered rectangular boundary: The space boundary in this configuration is assumed to host multiple furniture groups due to the inherent space division into subzones. For example, an L-shaped or a Z-shaped configuration can host two or more groups depending on space proportion.
   (c) Multi-grid boundary: The space boundary in this configuration is assumed to host multiple furniture groups with different alignments on its walls that take different angles, provided clear paths for circulation in between the furniture groups.
   (d) Circular boundary: The space boundary in this configuration allows for furniture groups with different alignment situations, where circular walls constrain the placement of furniture objects.
2. Irregular Space Boundary
   (a) Organic boundary: In this configuration, the space boundary is more constraining in terms of furniture layout possibilities, but defines an implicit subdivision of space based on organic virtual walls that define possible subzones for furniture placement.
   (b) Angled Boundary: The space boundary in this configuration, which is characterized by its sharp edges and angled walls, is assumed to host multiple furniture groups with constrained possibilities related to alignment and spatial relationships.

In these configurations, two main factors affect the possibilities of furniture arrangement: the feasibility of hosting single or grouped furniture elements in space based on physical dimensions, and the divergent/convergent approach exercised by the architect to generate rational arrangement alternatives within the given spatial conditions.

Our basic assumption for these spatial configurations involves single-level spaces with one or two entry points, windows and columns that are either embedded inside the walls or protruding within the space so as to implicitly define subzones. However, there exist other configurations with special elements that have direct implications for furniture arrangement, such as spaces with free-standing columns, multi-level spaces, and filleted or chamfered spaces, as illustrated in Fig. 2.

In these configurations, the column, the difference in level, and the chamfered wall imply virtual walls that either introduce subzones as in the case of the column and multi-level space, or introduce a virtual corner point for the space boundary. These have consequences on the logic of furniture arrangement within the newly

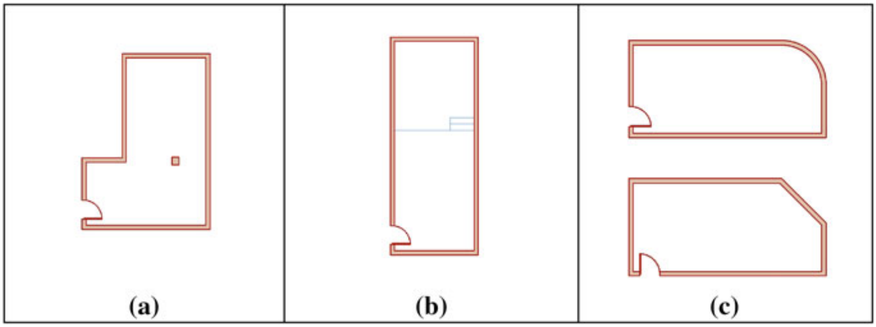


Fig. 2 Spatial configurations with implications for furniture arrangement: **a** space with free-standing column. **b** multi-level space, **c** chamfered space

perceived space. For the scope of this paper, we focus on single and clustered rectangular boundaries. In order to study possible furniture arrangements within a given space, we first provide operational definitions for three main components of the furniture-space setting: (1) furniture entities, (2) space entities, and (3) object-space relations. Then we define rules for each component that constitute the main algorithm for the proposed automated furniture layout system.

Furniture Entities

As shown in Table 1, we describe some of the main furniture entities. We classify elements of furniture in a given space into *Furniture Objects*, such as TV, chair, sofa, table, etc, and *Furniture Groups*, such as seating groups (including sofa, two armchairs, and table), or dining groups (including dining table and six chairs). Some of these objects and groups necessarily require clear orientation logic (*Front* and *Back*) for wall alignment purposes, such as sofas and chairs, while others do not, such as tables. We explicitly embed this logic for the purpose of our system, as this affects how furniture is placed within a given space and in relation to other elements.

We identify a *Furniture Base Point* by which the furniture object or group is spatially allocated. For each, we define a *Furniture Internal Boundary* and *Furniture External Boundary*. The internal boundary specifies the exact bounding box enclosing the object or group, and the external boundary includes circulation and use space (Fig. 3). This becomes significant in addressing circulation issues within a given space. We also define an *Entry Point* for furniture groups, as this defines a likely access point and has implications on circulation patterns within the space. For each internal boundary, we identify a *Corner* and *Center* for the purpose of alignment of furniture with a given wall or virtual wall.

Table 1 Furniture entity definitions

Definition	Description
Furniture object	A single furniture element (e.g. chair, table, TV, etc.)
Furniture group	An arrangement that includes a number of furniture objects (e.g. dining table with chairs, sofa with table and armchairs, etc.)
Furniture base point	A reference point for spatial allocation of the furniture object or group
Furniture internal boundary	A bounding box that encloses a furniture object or group
Furniture external boundary	A bounding box that includes a furniture object or group, and their corresponding circulation and use space
Center of furniture internal boundary	Center of gravity of bounding box of furniture object or group
Corner of furniture internal boundary	Corner of internal bounding box of furniture object or group
Corner of furniture external boundary	Corner of external bounding box of furniture object or group
Entry point of furniture internal boundary	A point that defines access from circulation space to a furniture object or group
Front of furniture external boundary	A line segment on the external bounding box that defines a non wall-aligning face of a furniture object or group, i.e. that faces another object or group in the space
Back of furniture external boundary	A line segment on the external bounding box that defines a wall-aligning or virtual wall-aligning face of a furniture object or group

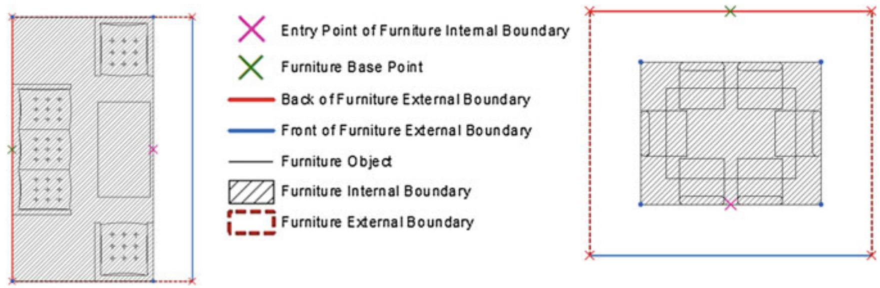


Fig. 3 Furniture group with distinguishable front and back external boundaries (*left*) and non-distinguishable front and back external boundaries (*right*)

After defining the basic furniture entities, we implement three categories of rules for furniture entities: (1) object-object rules, (2) object-group rules, and (3) group-group rules. For the object-object rules, we implement rules for distance constraints based on interior design standards data. For example, we define a range of 2–3 m as an allowed viewing distance between a sofa and a TV, measured from the *Center of Furniture Internal Boundary* for both objects.

For an object-group relation, such as seating group and TV, we assign the distance constraints to the median of the distances between each object of the seating group and the TV. The priority of sequencing of furniture objects and groups is significant in this case, as the seating group or sofa follows the TV object space allocation.

For group-group relations, we define distance/viewing constraints in addition to alignment and attachment constraints. For example, in a seating-dining group relation, we define alignment conditions (which can allow for alignment from Corner, Center, Front or Back of each furniture external boundary to allow for through circulation).

These rules define interrelationships between furniture objects and groups, yet not in context. In the next section, we analyze the space boundary and enclosure, and the logic of furniture allocation within that boundary.

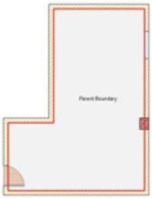
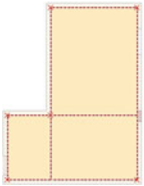
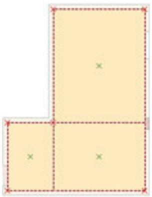
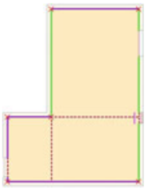
Space Entities

For our space analysis, we use the clustered rectangular configuration (an L-shaped space organization) as an example. To implement the logic of furniture allocation, we differentiate between two main entities: space boundary (or *Parent Boundary*) and *Wall Segment*. For an architect attempting to allocate furniture within space, we assume that two simultaneous processes take place; (1) *space decomposition*: a process of perceiving space and its boundaries and subdividing it visually into potential subzones for single or multiple furniture arrangements (depending on space proportion, area and specific layout configuration), and (2) *wall decomposition*: a process of aligning furniture objects or groups onto wall segments (rather than just full walls), which are physically divided by columns and openings, in addition to virtual walls, through basic geometric subdivisions.

Rather than an exhaustive approach of arbitrarily placing furniture objects in all possible points in a given area or aligned on all possible points on the wall, we constrain the furniture allocation possibilities to logical attempts by architects during the process of perceiving any configuration.

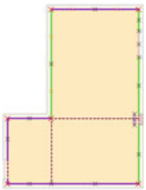
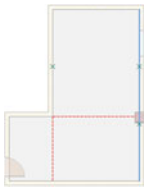
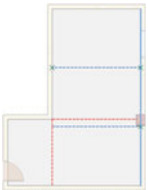
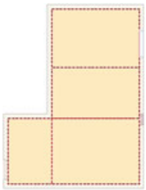
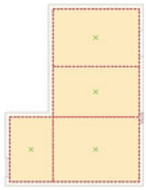
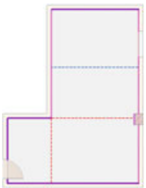
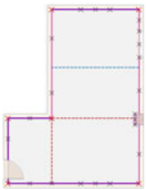
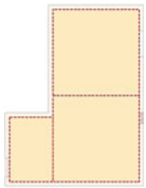
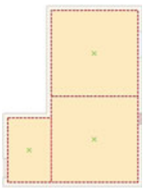
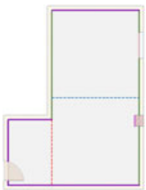
The logic of subdividing and decomposing space depends typically on two factors: the area/proportion of the space, and furniture size. As mentioned earlier, we take into consideration two main furniture components; objects and groups. For furniture groups, we assume that the minimum acceptable area of a group is $2.5 \text{ m} \times 2.5 \text{ m}$, measured from its external boundary to account for circulation and use. This becomes the basic unit upon which the space subdivision logic is based. To carry out the subdivision, we use virtual walls from key points on the parent space boundary (such as corner points, midpoints, one-third, and so on). The number of subdivisions, *child boundaries* and potential furniture allocation points is directly proportional to space area, where the larger the area, the more the virtual walls, and the higher the division level (DL), where DL1 is a subdivision of the first level with one virtual wall and two subzones, DL2 is a subdivision of the second level with two virtual walls and three subzones, and so on.

Table 2 Space entity definitions

Parent boundary The full boundary of a given space, excluding any openings or protruding columns	**Parent boundary corner points** Corner points on the space parent boundary	**Virtual wall from corner** Virtual wall from parent boundary points that subdivides a given space into subzones
		
Parent boundary segment for virtual walls Line segment on parent boundary resulting from space subdivision by a virtual wall	**Child boundary DL1** The full boundary of a subzone resulting from space subdivision by a virtual wall of first division level	**Child boundary points DL1** Points on the space child boundary resulting from space subdivision by a virtual wall of first division level
		
Virtual Wall from parent boundary points DL1 Virtual wall that subdivides space into two subzones	**Child boundary center DL1** The center of gravity of a child boundary subzone	**Wall segments DL1** Wall segments resulting from space subdivision by a virtual wall of first division level, in addition to subdivision by a column or window or door opening
		
Wall segment points DL1 Points on wall resulting from subdivision level 1 (dividing wall into two wall segments)	**Parent boundary segments DL2** Line segment on parent boundary resulting from space subdivision level 2	**Child boundary points DL2** Points on the space child boundary resulting from space subdivision by a virtual wall of second division level

(continued)

Table 2 (continued)

		
Virtual wall from parent boundary points DL2 Virtual wall that subdivides space into three subzones	**Child boundary DL2_1** The full boundary of a subzone resulting from space subdivision by virtual wall of second division level (alternative 1)	**Child boundary center DL2_1** The center of gravity of a child boundary subzone resulting from space subdivision by a virtual wall of second division level (alternative 1)
		
Wall segments DL2_1 Wall segments resulting from space subdivision by a virtual wall of second division level, and subdivision by a column or window or door opening (alternative 1)	**Wall segment points DL2_1** Points on wall resulting from subdivision level 2 (alternative 1)	**Child boundary DL2_2** The full boundary polyline of a subzone resulting from space subdivision by a virtual wall of second division level (alternative 2)
		
Child boundary center DL2_2 The center of gravity of a child boundary subzone resulting from space subdivision by a virtual wall of second division level (alternative 2)	**Wall segments DL2_2** Wall segments resulting from space subdivision by a virtual wall of second division level, in addition to subdivision by column, window or door opening (alternative 2)	**Wall segment points DL2_2** Points on wall resulting from subdivision level 2 (alternative 2)
		

We define below our space decomposition rules:

- If the length of the largest parent boundary line (the largest segment in space to determine subdivision) is greater than 5 m (to accommodate two furniture groups of minimum dimensions), draw virtual wall from the midpoint of the parent boundary line. If length is greater than 8 m, divide into three subzones with two virtual walls. If length is less than 5 m, no subdivision should be done (available space is sufficient for just one furniture group).
- In this layout configuration, virtual walls can also be drawn from parent boundary corners. Subdivision from corners is only allowed if the resulting subzones allow for a furniture group minimum dimension. Therefore, only if the vertical length of the shorter parent boundary line is greater than 2.5 m, a horizontal virtual wall can be drawn to introduce a new subzone, and vice versa.
- If DL1 or DL2 virtual walls are within near proximity (50 cm) to virtual walls from a corner, the virtual wall from corner precedes. The corner tends to define space and subzones physically more than visual subdivisions from the space boundary. If the distance however is from 50 cm to 2.50 m, both alternatives are considered.

Upon subdivision and the definition of child boundaries and new points on these boundaries, the virtual walls establish a new set of wall segments. The following rules are then applied on these segments:

- For wall segments larger than 0.25 m and less than 8 m, divide segment by 2. For segments between 3 m and 10 m, divide by 3, and for segments between 4 m and 12 m, divide by 4. A 6 m wall segment for example would contain furniture allocation placeholders at 3 points resulting from subdividing it into four segments.
- For columns and window openings, their midpoints and endpoints should be added to the list of potential points for allocation.
- If the virtual wall happens to hit the middle of the window, the system should allow for a subdivision after or before the window and at the midpoint as well to accommodate for visual placement of furniture groups or objects in relation to the window segment.

Table 2 describes the different space entities and the subdivision process of boundaries and walls to identify potential points for furniture allocation.

Object-Space Relations

We describe in this section rules for two basic relations between the space and furniture objects or groups: (1) alignment and (2) circulation. Regarding alignment, we distinguish between three types of furniture arrangements: wall-aligned furniture, free-standing furniture, or furniture that accepts both arrangements. The nature of how these furniture objects or groups align with the resulting child space boundaries differs according to these arrangements. If the furniture object or groups is free-standing, it can align with the child boundary in one of four possible

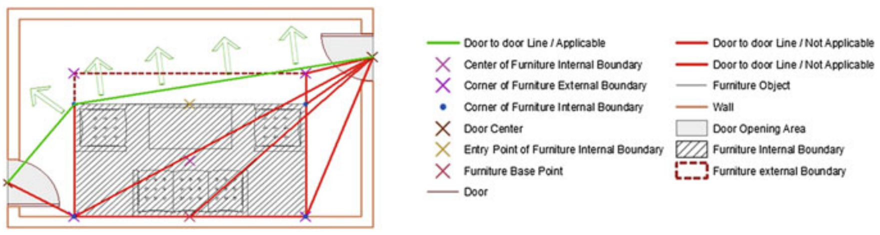


Fig. 4 Circulation rule for an object-space relationship

scenarios: (a) center of gravity of furniture to center of gravity of child boundary, (b) furniture boundary corner to child boundary corner, (c) furniture boundary base point to child boundary point, and (d) furniture boundary base point to nearest wall segment point DL1, DL2, DL3, etc. Wall-aligned furniture objects and groups can only align based on conditions (c) and (d).

Regarding circulation, we introduce this layer of analysis to narrow down possibilities of furniture allocation, considering how an architect would approach the problem. We put forward that circulation within a habitable space containing furniture objects is considered by the architect in two instances; (1) as a pre-checking mechanism, where basic walkability is determined through circulation area percentages, and (2) as a detailed study of possible circulation paths around furniture arrangements and their entry points. For the second more detailed circulation study, we propose the following rules for a given space with two doors or access points (Fig. 4):

- Identify *Furniture Boundary Internal* edges for furniture objects
- Connect from midpoint of first *Door Opening* to the nearest *Furniture Internal Boundary Corner* of each group.
- For each group or object, if there is direct access from nearest corner to second access point in space, without intersecting any internal boundary within that group or object, then draw a valid circulation edge. Else, search for next corner of the same group or object, and apply again until valid circulation edge is identified.
- The identified circulation edge represents one boundary of the circulation path. To identify a valid circulation path (with at least 80 cm), search for wall segment, column, or other edge (that defines the other boundary of the circulation path). If the distance between both boundary lines is equal to or more than 80 cm, add this path as a valid circulation path. Else, eliminate the alternative that contains this furniture arrangement from the possible solutions.

Case Study

We selected a 39 m^2 living area L-shaped configuration with two doors, one window, and a protruding column, as shown in Fig. 5. We introduced two furniture groups: a seating group (sofa, two armchairs, table), a dining group, including a dining table and six chairs, and a TV object.

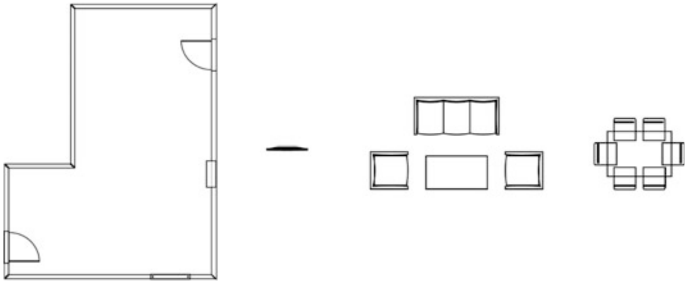


Fig. 5 Case study spatial configuration and introduced furniture groups and objects

We used Grasshopper to define the algorithm for generating possible furniture arrangements, as shown in Fig. 6.

For the algorithm, as illustrated in Fig. 7, we extend lines and draw virtual walls from Parent Boundary Corner Points.

We identify new space subdivisions and child boundaries based on the virtual walls. We draw *Virtual Wall from Parent Boundary Points DL1*, and identify the resulting *Child Boundary DL1*, *Child Boundary Center DL1*, *Wall Segments DL1*, and *Wall Segment Points DL1*. After applying *Virtual Wall from Parent Boundary Points DL2*, we identify the resulting *Child Boundary DL2_1*, *Child Boundary DL2_2, Child Boundary Center DL2_1*, *Child Boundary Center DL2_2, Wall Segments DL2_1*, *Wall Segments DL2_2, Wall Segment Points DL2_1,* and *Wall Segment Points DL2_2*.

After getting all possible furniture allocation points, we apply bounding boxes for *Door Opening Area* entities to exclude them from any possible furniture

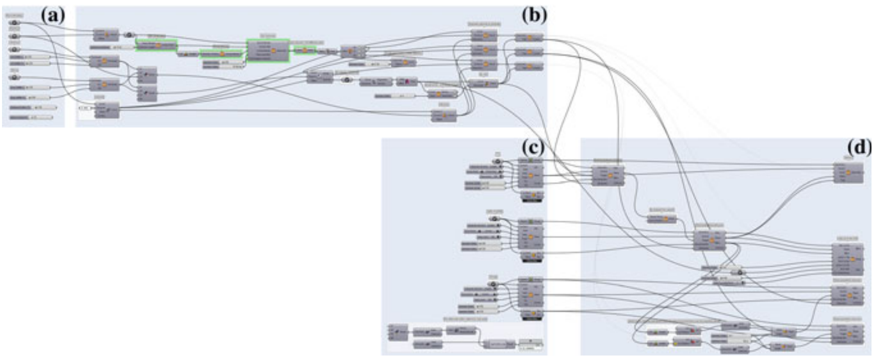


Fig. 6 The Grasshopper definition used to implement the algorithm for generating alternatives for furniture arrangement: **a** defining curves for room boundary, openings buffer, and other distance parameters controlling the placement of furniture groups, **b** extracting virtual walls, creating all possible combinations of subzones, **c** defining the furniture groups that will be used and setting their priority of placement, **d** placing furniture groups in all possible subzones while testing for collisions with room boundary and openings buffer

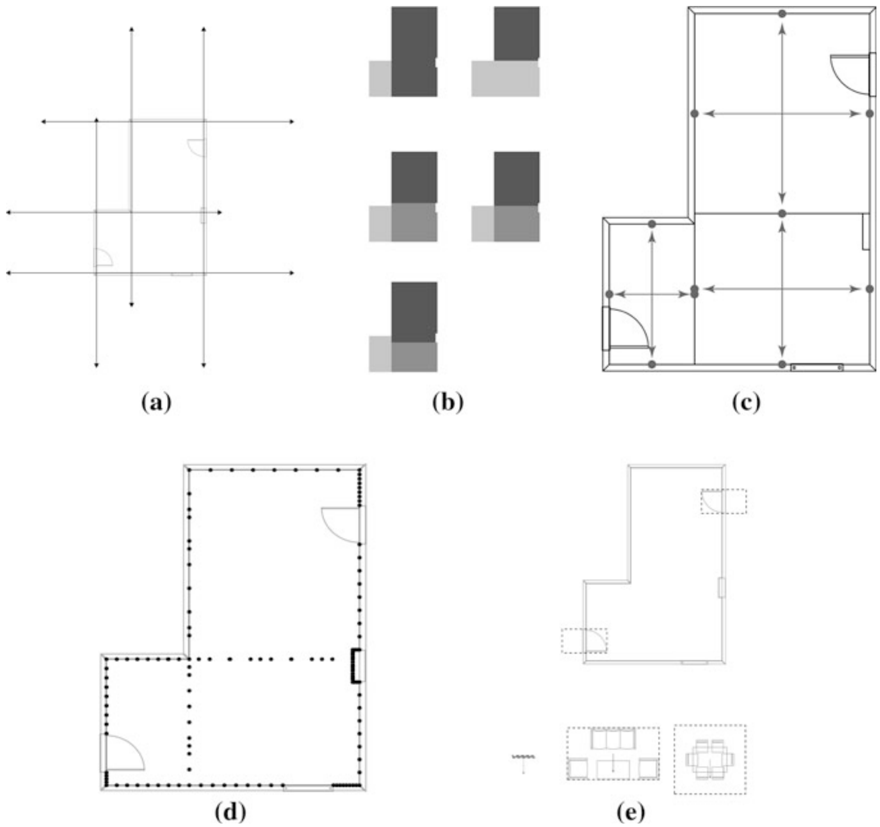


Fig. 7 Algorithm for generating possible furniture arrangements: **a** draw virtual walls from parent boundary corner points, **b** identify child boundaries, **c** identify center points of child boundary and child boundary points DL1, **d** identify resulting child boundaries DL2 and DL3, wall segments DL2 and DL3, and wall segment points DL2, DL3; **e** identify door opening bounding box, apply circulation rules and apply algorithm to dining group, seating group and TV

Fig. 8 Set of possible furniture arrangements generated by the algorithm

allocation. Then we apply the circulation rules to check for valid paths. We run the algorithm, given the object-group relations between the TV and seating group, and the group-group relations between the dining group and seating group, as defined earlier. For this case, 10 possible alternatives for furniture arrangement were generated, as shown in Fig. 8.

Discussion and Future Work

Our research addressed the automated generation of furniture arrangements within a given space, taking a residential living area as a sample with an L-shaped spatial configuration. The logic presented in this paper, which involves space decomposition and wall decomposition, is assumed to respond to the divergent/convergent conversation conducted by architects while attempting to analyze a given space, its boundaries, possible subdivisions and virtual subzones, in addition to walkability scenarios, while taking into consideration basic object-space and object-object relations. This comes in contrast to other previous approaches that computationally exhaust all possible solutions but might fall short of the logical and geometrical iterations that an architect experiences during a space planning or furniture layout process.

Our algorithm applies to mostly regular space configurations, with the exception of circular boundaries, which require—along with irregular space configurations—further analysis of space perception, subdivision logic, and object-space relations. We attempt to address different typologies of space boundaries, as well as different furniture and equipment typologies. As they currently stand, our furniture entities exhibit only a subset of all possible furniture classes (furniture with a fixed position in space such as sofas and dining tables, movable furniture such as chairs, etc.). Other possible classes that we have not addressed include multi-level furniture, dynamic and extendable furniture, furniture with dynamic use, and transformable furniture, only to mention a few. This also applies to equipment and accessories in space. Further investigation of these entities and their object-object and object-space relations will allow for an all-encompassing logic for a comprehensive allocation of any object typologies in space.

We addressed space subdivision, circulation and walkability in space and around furniture objects, according to design guidelines and standards, and based on how architects would simply attempt to perceive a given configuration. More evidence is needed to corroborate this basic assumption. Our ongoing research involves assigning a group of configurations and furniture requirements to experienced architects in the form of a graphical survey, where they are required to allocate furniture objects in those configurations based on intuition and experience. Results from these surveys—in the form of object and space coordinates—will be used to infer statistically the guidelines and rules for our automated furniture generation system.

Another venue to explore is deducing rules and guidelines from patterns of use and behavior in different spatial configurations, perhaps extending the context of study into other settings other than residential units, where the complexity of behavior and space dynamics are expected to highly inform the automated space planning and furniture layout generation.

Our approach in general relied on a consistent dataset resulting from geometric subdivision of spaces and allocating furniture objects based on key points on a spatial configuration. Our next step involves expanding the scope of our input parameters to include complex and inconsistent datasets, including volumetric spatial configurations, visibility, access, behavior patterns, environmental aspects, etc. These dimensions cannot be incorporated into our system as it stands, especially when the objective does not only involve generation of configurations and furniture layout schemes, but also an informed evaluation and optimization of those schemes. We assume that the integration of fuzzy logic would contribute to achieving a more true representation of spatial configurations and complex use scenarios rather than just geometrical relations informed by non context-specific design guidelines.

Acknowledgements We would like to thank UNii Engineering Consultancy for sponsoring and supporting this research.

References

Abdinnour-Helm S, Hadley SW (2000) Tabu search based heuristics for multi-floor facility layout. Int J Prod Res 38:365–383

Ahuja RK, Orlin JB, Tiwari A (2000) A greedy genetic algorithm for the quadratic assignment problem. Comput Oper Res 27:917–934

Aiello G, Enea M, Galante G (2006) A multi-objective approach to facility layout problem by genetic search algorithm and Electre method. Robot Comput-Integr Manuf 22:447–455

Aiello G, La Scalia G, Enea M (2012) A multi objective genetic algorithm for the facility layout problem based upon slicing structure encoding. Expert Syst Appl 39:10352–10358

Akazawa Y, Okada Y, Niijima K (2005) Automatic 3D scene generation based on contact constraints. In: Proceedings of conference on computer graphics and artificial intelligence, pp 593–598

Al-Hakim L (2000) On solving facility layout problems using genetic algorithms. Int J Prod Res 38:2573–2582

Azadivar F, Wang J (2000) Facility layout optimization using simulation and genetic algorithms. Int J Prod Res 38:4369–4383

Balakrishnan J, Cheng CH (2000) Genetic search and the dynamic layout problem. Comput Oper Res 27:587–593

Banerjee A, Quiroz JC, Louis SJ (2008) A model of creative design using collaborative interactive genetic algorithms. In: Design computing and cognition'08. Springer, Berlin, pp 397–416

Boswell SG (1992) TESSA: a new greedy heuristic for facilities layout planning. Int J Prod Res 30:1957–1968

Buscher U, Mayer B, Ehrig T (2014) A genetic algorithm for the unequal area facility layout problem. In: Operations research proceedings 2012. Springer, Berlin, pp 109–114

Chiang W-C, Kouvelis P (1996) An improved tabu search heuristic for solving facility layout design problems. Int J Prod Res 34:2565–2585

Dunker T, Radons G, Westkämper E (2003) A coevolutionary algorithm for a facility layout problem. Int J Prod Res 41:3479–3500

El-Baz MA (2004) A genetic algorithm for facility layout 706 problems of different manufacturing environments. Comput Ind Eng 47:233–246

Flemming U, Coyne R, Glavin T, Rychener M (1988) A generative expert system for the design of building layouts—version 2. Technical Report. Carnegie Mellon University, Engineering Design Research Center (EDRC-48-08-88)

Germer T, Schwarz M (2009) Procedural arrangement of furniture for real-time walkthroughs. Comput Graph Forum 28(8):2068–2078

Honiden T (2004) Tree structure modeling and genetic algorithm-based approach to unequal-area facility layout problem. Ind Eng Manag Syst 3:123–128

Hu MH, Ku M-Y, Chen C-C (2007) A genetic algorithm for optimizing facility layout in a wafer fab. In: IMECS, pp 2026–2031

Indraprastha A, Shinozaki M (2011) Elaboration model for mapping architectural space. J Asian Archit Build Eng 10(2):351–358

Islier A (1998) A genetic algorithm approach for multiple criteria facility layout design. Int J Prod Res 36:1549–1569

Kim J, Kim Y (1999) A branch and bound algorithm for locating input and output points of departments on the block layout. J Oper Res Soc 50:517–525

Kjølaas KAH (2000) Automatic furniture population of large architectural models. Master's thesis, Massachusetts Institute of Technology, Department of Electrical Engineering and Computer Science, Cambridge, MA

Kochhar JS, Foster BT, Heragu SS (1998) HOPE: a genetic algorithm for the unequal area facility layout problem. Comput Oper Res 25:583–594

Koenig R, Schneider S (2012) Hierarchical structuring of layout problems in an interactive evolutionary layout system. Artif Intell Eng Des Anal Manuf 26:129–142

Larive M, Roux OL, Gaildrat V (2004) Using meta-heuristics for constraint-based 3D objects layout. In: Proceedings of conference on computer graphics and artificial intelligence, pp 11–23

Lee K-Y, Roh M-I, Jeong H-S (2005) An improved genetic algorithm for multi-floor facility layout problems having inner structure walls and passages. Comput Oper Res 32:879–899

Medjdoub B, Yannou B (2000) Separating topology and geometry in space planning. Comput Aided Des 32(1):39–61

Merrell P, Schkufza E, Li Z, Agrawala M, Koltun V (2011) Interactive furniture layout using interior design guidelines. ACM Trans Graph 30(4): 87:1–87:9

Rodrigues E, Gaspar AR, Gomes Á (2013) An approach to the multi-level space allocation problem in architecture using a hybrid evolutionary technique. Autom Constr 35:482–498

Rosenblatt MJ (1986) The dynamics of plant layout. Manag Sci 32:76–86

Sanchez S, Roux O, Luga H, Gaildrat V (2003) Constraint-based 3D-object layout using a genetic algorithm. In: Proceedings of conference on computer graphics and artificial intelligence

Shayan E, Chittilappilly A (2004) Genetic algorithm for facilities layout problems based on slicing tree structure. Int J Prod Res 42:4055–4067

Tam K (1998) Solving facility layout problems with geometric constraints using parallel genetic algorithms: experimentation and findings. Int J Prod Res 36:3253–3272

Wang M-J, Hu MH, Ku M-Y (2005) A solution to the unequal area facilities layout problem by genetic algorithm. Comput Ind 56:207–220

Wu Y, Appleton E (2002) The optimisation of block layout and aisle structure by a genetic algorithm. Comput Ind Eng 41:371–387

Xie W, Sahinidis NV (2008) A branch-and-bound algorithm for the continuous facility layout problem. Comput Chem Eng 32:1016–1028

Yu L-F, Yeung S-K, Tang C-K, Terzopoulos D, Chan T, Osher S (2011) Make it home: automatic optimization of furniture arrangement. ACM Trans Graph 30(4): 86:1–86:11