DeciGenArch

A generative design methodology for participatory architectural configuration via multi-criteria decision analysis

Aditya Soman[1], Shervin Azadi[2], Pirouz Nourian[3]
[1] Ramboll UK [2] Eindhoven University of Technology [3]Delft University of Technology
[1] aditya.soman@ramboll.co.uk [2]s.azadi@tue@nl [3]p.nourian@tudelft.nl

Our approach to Generative Design converts the problems of design from the geometrical drawing of shapes in a continuous setting to topological decision making about spatial configurations in a discrete setting. The paper presents a comprehensive formulation of the zoning problem as a sub-problem of architectural 3D layout configurations. This formulation focuses on the problem of zoning as a location-allocation problem in the context of Operations Research. Specifically, we propose a methodology for solving this problem by combining a well-known Multi-Criteria Decision-Analysis (MCDA) method called 'Technique for Order of Preference by Similarity to Ideal Solution' (TOPSIS) with a Multi-Agent System (MAS) operating in a discrete design space.

Keywords: *3D Layout problem, Participatory Design, Multi-Agent System, Multi-Criteria Decision Analysis, TOPSIS.*

INTRODUCTION

The problem of architectural 3D layout configuration, when regarded in its full sophistication, could be regarded as a classical example of a 'wicked problem' (Rittel and Webber, 1973),(Lobos and Donath, 2010) due to the numerous human and physical complexities involved (Zawidzki, 2016). Breaking this problem down into a set of subsequent problems through a methodical approach facilitates understanding and addressing the underlying complexities like multi-dimensional, multi-criteria, multi-actor, multi-value complexities holistically (Azadi and Nourian, 2021).

In this paper, the zoning problem, which is one of the most critical parts of the 3D layout problem, is studied in detail, and a methodology for resolving it is proposed. The zoning problem is arguably one of the most essential parts of the 3D layout problem. Generating and locating the zones appropriately in the building mass has a massive impact on the building's functionality, efficiency, and utility for its occupants and their activities. Improper zoning can result in a poorly configured building, leading to many social, environmental, economic problems, which casts a shadow on the sustainability of the building in its life cycle.

The zoning problem in a continuous setting would be concerned with the shape of the spatial zones designated to the distinct activity spaces (also known as departments or facilities) as foreseen in a programme of requirements (colloquially referred to as the so-called programmatic functions). In a discrete spatial setting, however, the design/decision space is already subdivided into volumetric units (voxels). This transforms the problem of zoning into a location-allocation problem, as it involves the allocation of and locating various zones in a discretized building mass corresponding to different programmatic functions. Simply put, such a 'location allocation' problem is concerned with the difference made in terms of multiple quality criteria by different assignments/allocations of such zones to the existing cellular spaces. If the voxelated design/decision space has a non-trivial shape determined a priori, we refer to it as the mass or the envelope interchangeably; referring to the colloquial convention referring to the rough design of a building as 'massing'. A typical goal of the zoning problem is to achieve the required space and spatial qualities for all the zones as stated in a so-called Programme of Requirements (PoR), which is defined by the design choices made by the actors/stakeholders while maintaining the closeness requirements of the zones as prescribed in a so-called Activity Relationship Chart (ARC, a.k.a. RELChart).

Our methodology proposes a Multi-Agent System combined with a MCDA whose computational agents act as proxies for the departments listed in the PoR seeking to meet the quality criteria. Practically, the agents can pursue an adaptable agenda of human actors and grow a configuration for the zoning problem collaboratively in participatory settings. Determining the appropriate location of zones highly depends on the spatial quality preferences set by the actors.

The level of control over the geometrical and topological resolution is also an important part of the problem. Forming the best possible configuration according to the design intention while maintaining topological connectivity for the zones is essential and challenging to manage when following the desirability results regarding the location of the zones. In this approach, the main actor (architect) in this particular level of the problem needs to define the rules/policies for the growth of the zones, which will generate the intended spatial quality of the zones to ensure the validity of the design solution.

An illustrative problem for the design of a mixed use residential/commercial development in North Amsterdam is considered to test the method proposed in the paper. The ready availability of the data of the site and the future plans for the site drafted by the municipality of Amsterdam gives a PoR, and complex multiple quality criteria to simulate a real-life scenario. An arbitrary massing model is considered as a base for the zoning problem since the generation of the massing model is a complex sub-problem in the 3D layout problem itself and lies out of the scope of this paper (Azadi and Nourian, 2021). Once this base information is collected, a zoning problem will be defined where the number and size of the required zones, and the specific quality criteria (and their weights of importance given by the actors) are given. Once the problem is defined as such the MAS runs to find a satisfactory solution. It must be noted that going beyond a single quality criterion means that there cannot be a globally optimal solution and thus we reformulate this problem as a matter of adaptation instead of optimization, aiming at devising a Spatial Decision Support System (SDSS) for finding the most satisfactory solution for a given PoR specified by potential stakeholders. In other words, the proposed methodology is meant to facilitate exploration of such decision spaces for mass-customization of housing as a matter of adaptation to different environmental contexts as well as the wishes of various actor groups.

BACKGROUND

A simple combinatorial analysis will show that in the discrete setting posed for the problem, a naive search in the space of all possible configurations will be rendered infeasible due to the astronomical size of such a set of possible permutations, e.g. for n voxels and m colours, then we will have m^n possible zoning alternatives (in this case with ~1000 voxels and 7 colours, this will be beyond the capacity of most calculators). Therefore, historically, the problem of zoning with comparable formulations has been approached in single-objective and multi-objective optimization settings as a matter of heuristic optimization. Recently, attention has been drawn to the formulation of this problem as a game for multiple agents learning how to achieve the goals and solve the constraints at the same time. For a recent review of these approaches see (Veloso and Krishnamurti, 2020). A good example of this can be seen in the work of (Guo and Li, 2017) where they combined a 3D bubble diagram with a grid-based system to allocate activities for architectural layouts. The agents of the MAS designed in this method have the capability to interact with each other and the environment using actions like swap attraction, repulsion and compression. In the simulation, the agents are placed according to adjacency requirements initially and an evolutionary solver is used for optimization where at each iteration, a child solution is generated by mutation. In the interactions the room identities are exchanged, and room shapes are changed by moving connected faces. The parent and child are evaluated by their room shapes, dimensions, aspect ratios and building shape.

Another interesting example of a multi-agent approach can be seen in the work of (Dino, 2016) where the Precedence-Based Layout Configuration Heuristics (P-LCH) was introduced, presenting a tool called Evolutionary Architectural Space layout Explorer (EASE) for early architectural design that aims to facilitate better-informed decision-making under spatial constraints using evolutionary optimization. In this approach the starting point is discretization of the space into a voxel grid resulting in the conversion of the problem into a location allocation problem, where each space with *n* units needs to be matched with *n* voxels. An objective statement is developed which is an aggregation of spatial qualities like compactness, closeness, accessibility, spatial aspect ratios and an evolutionary solver is used to develop iterations and pick the best one. The level of control and the element of participation by the human actors is missing in most of these approaches which is a key differentiator with our approach.

FORMULATION OF THE ZONING PROBLEM

The zones are to be created according to the higher-order classification of the functions of the space program as seen in the Table 1. There is no explicit closeness requirement between the zones in the mixed-use program due to the nature of the spatial functions involved. However, if the zoning scale is changed to a higher resolution, then the closeness requirements can be derived for units inside the individual zones. So the problem statement is twofold: firstly, the higher-order zoning problem where there is no closeness requirement, and secondly, there is a closeness requirement for zoning of units inside the zones obtained in the first step. We dub the zoning problem at a higher resolution (smaller scale) the Unit Assignment problem.

The ultimate goal in resolving the zoning problem is to assign voxels to zones in a discretized massing model, which will achieve a state of equilibrium, characterized with the aggregate qualities being in balance such that they have a minimum distance from the ideal possible state and the maximum distance from the worst possible combination of decisions. The tokens of decision analysis will be the values of voxels in terms of allocating the most desirable voxels for each zone while keeping all other zones in balance. In a nutshell, the problem of zoning can be defined as follows:

Given a discretized design/decision space consisted of voxels, a set of environmental quality criteria defined as scalar fields with known values over the whole range of the discrete elements barring the elements which are reserved for circulation, and a set of arbitrary weights of importance for these criteria, it is desired to find an assembly of 'coloured' voxels, i.e. configuration consisted of voxels assigned to a set of colours/programmatic labels as a decision matrix, such that the whole configuration will have the highest attainable aggregate quality for all zones (colours).

Note that the notion of 'highest attainable quality' needs to be further specified in different ways. As mentioned before, in a multi-criteria & multi-actor setting with various weights of importance set for the criteria there does not exist a universally ideal solution to be found, hence a variety of MCDA methods exist for decision analysis in such complex settings (Wątróbski *et al.*, 2019).

CASE STUDY

We use the case study of a mixed-use commercial/residential building in the district of Buiksloterham in North Amsterdam as described before.

Site features. The site is a rectilinear shaped 23,000 m^2 in size. It is surrounded by a canal joining the river IJ on the East-side, and the main road is accessible from the west. There is a new mixed-use development on the south side which can potentially block the access to sunlight to the site.

Spatial Program. The space allocation program proposed for this location is based on the existing proposal of the municipality of Amsterdam. The housing part of the program is divided into three categories: Privately-owned housing, Social-sector rental, Free-sector rental housing; and the commercial part consists of office spaces, restaurants, and retail spaces. There is also a need for parking spaces for bikes and limited spots for car parking. The distribution of these programs can be seen in Table 1.

Zones	Total Volume(m^3)	Number of Voxels
P_Owned_Volume	89640	415
S-Rental_Volume	65016	301
FS-Rental_Volume	87048	403
Restaurant_and_Cafe_Volume	4104	19
Retail_stores_Volume	15768	73
Office_Volume	20088	93
Parking_Bikes_Volume	11016	51

Table 1
Table showing the PoR for the case study

Massing Model. The volumetric design (massing) depends on the urban level impact on the surroundings, the architectural vision, design concept, socio-cultural context, etc. which lie out of the scope of this paper. This paper is part of the larger project done by the author where solutions for all the stages of the 3D layout problem are discussed in detail, with the same problem case (Soman, 2021). The solution developed for the massing problem in that project is considered as a base for the zoning problem formulation in this paper. The massing model is discretized into 3D voxels (Azadi and Nourian, 2021b) whose size is decided by the desired resolution as seen in figure 1. This enables discrete decision-making in the form of location-allocation problem, which we will elaborate on in the methodology section .

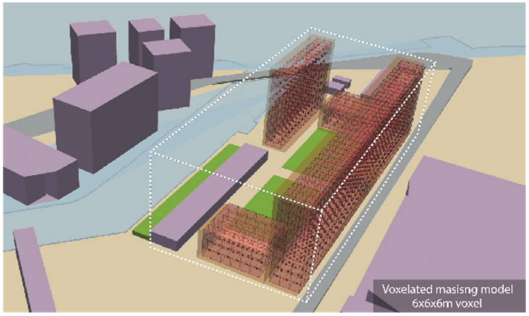

Figure 1
Diagram showing the discretised massing model

METHODOLOGY

The proposed methodology consists of four major steps:

1. Evaluating of absolute environmental quality criteria using ex-ante assessment methods for all voxels in the discretized volume, resulting in the so-called 'Environmental Quality Lattices'.
2. Aggregation of quality criteria per voxels for each zone considering the different weights of quality criteria for that zone using an MCDA method (e.g. TOPSIS), resulting in the fields of desirability, dubbed as 'Desirability Lattices'.
3. Defining space allocation agents as proxies for growing zones in a Multi-Agent System (MAS).
4. Setting the assignment approach and running the MAS until they realize their volumetric goals.

Environmental quality lattices

Environmental quality lattices represent the result of the ex-ante assessment of spatial quality criteria in the envelope. The absolute environmental quality criteria are specific to each design case and can be modified according to the project's context. These indicators are classified into two types based on the nature of their evaluation process: firstly, distance-based indicators representing spatial qualities that depend on the euclidean or geodesic distance of each voxel to a target concerning such things as closeness, noise, etc.; and secondly, sightline-based indicators which describe spatial qualities that depend on the existence of a clear sightline between each voxel and a target such as daylight, visibility, sky view factor, etc.

Distance-based. The two specific distance-based indicators that we utilize here are closeness indicators and quietness indicators. Closeness indicators represent the spatial quality of being close to roof, ground, and facades, See figure 2. The computation of closeness indicators is based on the distance between the reference voxels and the outermost voxels in the direction of interest.

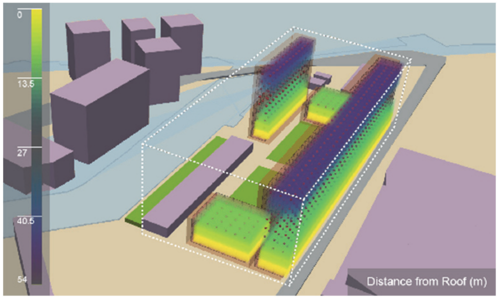

Figure 2
Diagram Showing the distance from roof scalar field

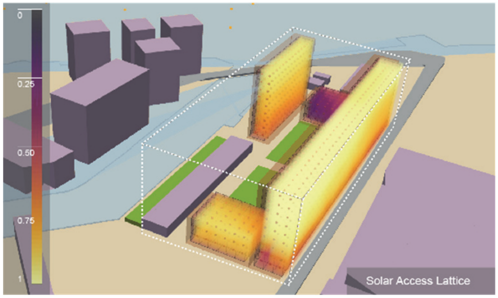

Figure 3
Diagram Showing the Sun-access Scalar field

The second type of distance-based indicator, the quietness lattice estimates the complement of relative noise potential received at each voxel. The precise evaluation of noise is complicated and requires information about the used materials, insulation, exact sound pressure at the sound sources, etc. However, for decision-making purposes, simple indicators like the aggregate distance from the noise source could function as a comparatively accurate enough proxy that indicates how quiet space would be based on a noise source's location.

Sight-line-based. The first type of sightline-based indicator introduced here is the daylight access indicator. Access to daylight is critical for a healthy indoor environment and has shown links to productivity and well-being. Daylight analysis is conventionally performed in a later stage of the design process where the internal layout of spaces, the geometry of the openings, and material property of the facade elements are defined. However, for decision-making purposes, we can simplify the evaluation process to compute the number of hours during the year that each voxel receives daylight. This can be calculated by the vertical sky component (VSC), representing the fraction of sky visible from a voxel reference considering the obstructions in the surroundings in terms of solid angle. That area of the visible sky is expressed as a percentage of an unobstructed hemisphere of sky and, therefore, represents the amount of skylight available for that particular window. It does not consider the size of the room or the number of windows; it only calculates the access to the sky.

Sun vectors at hourly intervals were selected as the target, and rays were shot from all voxels in the reverse direction of sun vectors. The number of obstructed rays was computed, and the percentage of accessible solar rays for each voxel was determined. This metric can indicate the solar access for each voxel in the massing, see figure 3,4.

Finally, the facade visibility indicator is essential to ensure the compatibility of a building with its

surroundings. The visibility of a building from different viewpoints like main streets, public areas, or other buildings becomes critical to the location of certain functions which thrive on better visibility like retail shops and restaurants. The degree of privacy can also be determined based on the direct visibility of a space from a point.
The evaluation method is similar to the solar-access as described before, the only difference being that instead of sun vectors, homogeneously scattered points on the surface of the visibility target are used as the sources of rays. The visibility towards the river IJ and the visibility to the building mass from the road was considered for the case study.

Input	Data Type	Input Name: Notes
Array of Voxels	$\mathbf{V} := [v_i]_{n\times1}$	Array of all the discrete volumetric elements of the mesh representing the building mass
Sun Vectors	$\mathbf{S_p}$	Selected sun-positions for the HOY's.
Array of Rays	$\mathbf{r} := [r_k]_{m\times1}$	Array of all the rays that are supposed to be shot from sun toward the voxel centroids
Context mesh	$\mathcal{M}_c$	The surface mesh representing the context of the building
Output	**Data Type**	**Output Name: Notes**
Array of Scalar values	$\mathbf{S_a} := [S_i]_{n\times1}$	Array of scalar values having the same structure as the voxel array indicating the sun-access in percentage

Problem: generate a scalar array indicating the access to sun for each voxel in the discretised model represented in percentage.

Algorithm 3: Sun-access lattice Algorithm

```
Generate HOY input (W):
    I ← day intervals for the HOY = 30
    foreach day in range(365) do
        if d % I == 0 then
            foreach h in range(24) do
                i ← d * 24 + h S_p = sun-positions[i]
    return S_p
r= rays originating from S_p towards voxel centroids Sun-access
  Lattice construction (r, Mc, V):
    V' ← initiate an empty lattice with same structure as V
    foreach voxel v in V do
        foreach ray r in R do
            I ← check intersection of Mc and ray (r, v) // a
               ray with the source of v and direction
               of r I is 1 when true and 0 when false
            val ← percentage of intersections for each voxel for
              all the rays.
            val ← append the val in V'
    return V' which equals S_a
```

Figure 4
Framework and Pseudocode for Algorithm 1 : Sun access Lattice generation

Desirability lattices

To find the most suitable location for each zone in the envelope, we aggregate the environment lattices based on the weights that human actors have specified for the importance of each indicator for each zone and form the desirability lattices that describe how desirable each voxel is for each zone. These lattices are the basis for the decision-making process of the computational agents in generating the zoning model.
To aggregate the incomparable values of different spatial quality indicators, we use Multi-Criteria Decision Analysis (MCDA) methods. The selection of an appropriate MCDA method is crucial for the generation of an appropriate solutions. In his paper (Wątróbski *et al.*, 2019) has described a formal guideline for MCDA method selection, which is independent of the problem domain. We utilized his tool (Watrobski, 2018) to choose an appropriate MCDA method. The following details were added to find the MCDA method (has weights, weights type(quantity), scale(quantity), has uncertainty(no uncertainty), topic(ranking and choice)). The analysis of the zoning problem indicates that TOPSIS is an appropriate method for solving this MCDA problem.

TOPSIS (Technique for Order of Preference by Similarity to Ideal Solution) is a multi-criteria decision-making technique used to rank a finite set of alternatives based on the minimization of distance from an ideal point and the maximization of distance from an anti-ideal point (Assari, Maheshand and Assari, 2012). The normalization step in the method can help compare values with different ranges, scales, and units, which is especially beneficial when comparing different environmental simulation values. One of the results of the decision making for the zones can be seen in figure 5

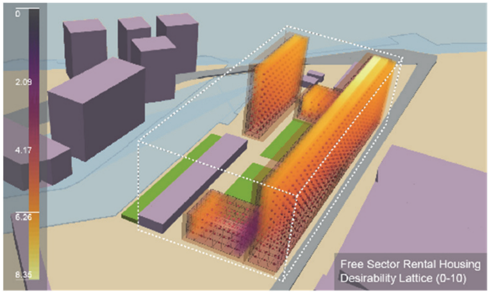

Figure 5
Diagram Showing the desirability lattice for Free sector rental housing zone

Agent behaviours

Agent behaviors are the actions that the computational agents representing the zones can take to achieve the goals set for that zone. The primary behavior of each agent is the occupy behavior. The occupy behavior calls for parsing through the voxels in the discretized massing model for the desirability values and deciding if the voxel is occupied or left empty. Each zone has a specific number of voxels that they have to occupy, depending on the size of each zone. The agents can also target a voxel to achieve a particular shape if required. Here, two occupation behaviors are presented: the 3D Cuboidal behaviour, and the 2D Rectilinear behaviour.

3D Cuboidal Behaviour. In this Behaviour, the agent searches for neighbours using a 26-neighbourhood stencil as seen in figure 6,7. A neighborhood stencil is a topological object from the topoGenesis python library (Azadi and Nourian, 2020) which describes a neighbourhood in a discrete 3D space. The neighbour with the maximum desirability value is selected and occupied, and in the next iteration, the neighbours of the same occupied agent will be used as candidates. To achieve the cuboidal form, candidate voxels that are neighbours of more than one occupied voxel are given a boost in their desirability value, which increases their chance of being selected as the next voxel for occupation and inclines the growth behaviour toward cubic shapes

2D Rectilinear behaviour. The 2D rectilinear behaviour also works similar to 3D Cuboidal, but the search stencil is an 8-neighborhood one in 2D.

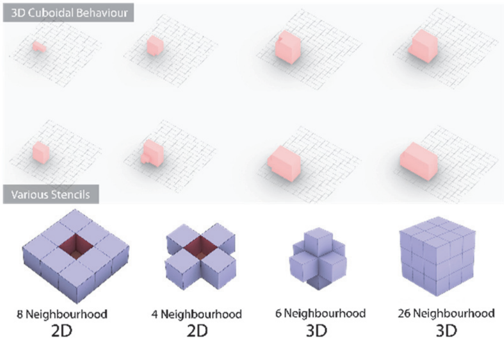

Figure 7
3D cuboidal behaviour steps and various stencils used

Figure 6
Framework and Pseudocode for Algorithm 2 : 3D Cuboidal Occupy behaviour

Input	Data Type	Input Name: Notes
Array of Voxels	$\mathbf{V} := [v_i]_{n\times1}$	Array of all the discrete volumetric elements of the mesh representing the building mass
Array of Environ-ment values	$\mathbf{E} := [E_i]_{n\times1}$	Array of scalar values having the same shape and size as the voxel array representing a results for decision making about the spatial quality of the zone.
Target growth size	$\mathbf{T}$	List of numbers representing the voxel count which each zone has to grow into.
Search Stencil	$\mathbf{s}_{st} := [S_st]_{n\times1}$	Array defining the neighbourhood definition of the search space.
Output	**Data Type**	**Output Name: Notes**
Array of Voxels	$\mathbf{V} := [v_i]_{n\times1}$	Array of coloured voxels or occupied voxels indicating the location of the zones grown in the voxelated array

Problem: grow a decision using computational agents to form zones of same colour in the voxelated array.

Define Agent Class $(\mathbf{Origin}, \mathbf{Stencil}, \mathbf{id})$:
  $O \leftarrow$ Agent origin which is a voxel from V'
  $id \leftarrow$ Agent id which is a number for identification
  $S_{st} \leftarrow$ Neighbourhood definition for searching
  $N \leftarrow$ Retrieved neighbours for agent at location O according to S_{st}
  Define Agent Agent behaviour *(***self***)*:
    foreach *voxel* v′ *in* **N do**
      $A \leftarrow$ check availability of voxel for occupancy `// A value above or below 1 indicates voxel is occupied`
      $val \leftarrow$ retrieve value from E corresponding to the neighbour id's
      $N_o \leftarrow$ The argmax of val is the new agent origin
      update the value in V' with id at the location of N_o
    return V' with agent at N_o and voxel at O coloured with the agents colour.

Define Environment Class $(\mathbf{E}, \mathbf{V'}, \mathbf{A}, \mathbf{B}, \mathbf{T})$:
  $E \leftarrow$ Input Environment lattices
  $V' \leftarrow$ Copy of input Voxel array
  $A \leftarrow$ List of Agents
  $B \leftarrow$ List of Agent Behaviors
  return E_c Environment

Run Simulation $(\mathbf{E_c})$:
  foreach *env* v′ *in* $\mathbf{E_c}$ **do**
    foreach *agent* v *in* **env do**
      $B \leftarrow$ perform the behavior B from E till the target T is met
      update the value in V' with id's of the agents in the environments E_c of the base lattice
    return V' coloured voxelated array indicating zones inside the voxelated discretized massing model.

Spatial Agent assignment process

In our approach the zoning problem is considered as an example of an assignment problem researched extensively in Operations Research. The problem was first described by (Koopmans and Beckmann, 1957) as a relatively simple problem in the allocation of indivisible resources is that of matching two sets of an equal number *n* of objects by making up pairs of objects consisting of one object from each set (n voxels and m colours). Objects belonging to the same set are similar but not identical. For each of the m^n possible pairs, a score or value is given. The problem is to find a matching (or assignment to each other) of objects for which the sum of the scores of pairs matched is as high as possible. Two methods were taken to fill the nxm permutation matrix. The Linear (concurrent) assignment method which prioratises the rows and the Sequential assignment one which prioratises the columns of the permutation matrix.

Linear Assignment.

The objective of the linear assignment approach is to maximize the assignment benefit of the agent origins at a set of locations in the discretized massing model. In simple terms, the problem deals with assigning the agents at locations in the voxel grid so that each agent occupies the most desirable voxels for their zones, thus generating a zoning option that has the highest overall desirability.

The steps followed in this process were as follows:

1. Several agents were created for each zone in the problem depending on the size of the zone. The intention is to create an equal occupancy target for each agent to minimize the conflicts between agents.
2. Environmental lattices are aggregated based on the weights of each zone, and the desirability lattices were generated for all the agents.
3. Considering the number of voxels that each agent in the process has to occupy, a radius is determined. Considering this radius, the voxels in the discretized models are parsed for possible agent origin locations.
4. The occupy behaviour is simulated for agents from each zone at all the possible agent origin positions considering their respective desirability lattices. The total aggregated desirability value is recorded as a benefit for that position for that agent. This process is repeated for all the agents to create a benefit matrix.
5. The CP-SAT solver from the Google OR tools (Perron and Furnon) is used to solve the linear assignment problem with respect to the benefits matrix, and the permutation matrix is generated.
6. The MAS simulation is executed to generate the solution for the zoning problem. See figure 8.

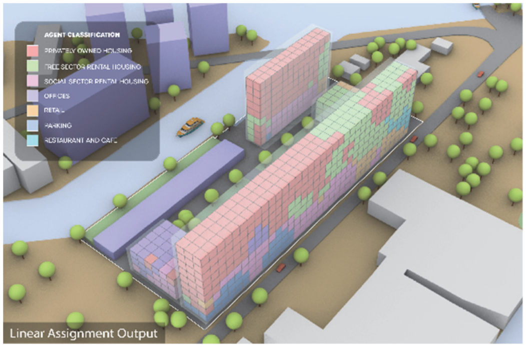

Figure 8
Diagram Showing the final zoning output of the Linear Assignment process

Sequential Assignment. The second approach considered for growing the decision of the actors was the Sequential assignment approach. In this approach, the zones are sequentially configured one after the other till all the zones are configured. This approach is similar to the design process that is followed conventionally where the zones maintain their topological structure and form.

The steps followed in this process were as follows:

1. The zones are ranked according to their level of impact on the overall configuration. The level of impact is determined by the zone size and the requirement of unrestricted access to the voxel grid. In the case study, privately-owned housing is the largest zone by size; however, it is critical to access the ground-level voxels for the parking zone. Hence both are given the first priority in the ranking list.
2. Similar to the first approach, the agent origin locations will be ranked based on the potential benefit that the agent can attain from that location.

3. All the agents of a particular zone will be deployed together considering their rank from the first step, and zones will be configured in the order of their ranking. See figure 9.

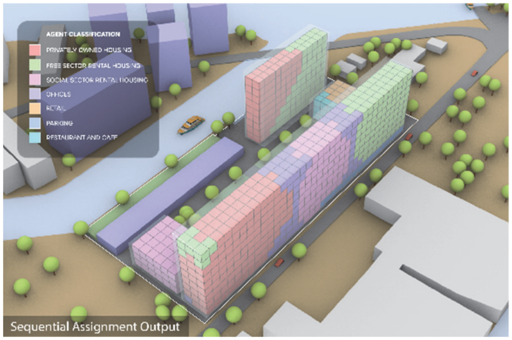

Figure 9
Diagram Showing the final zoning output of the Sequential Assignment process

RESULTS

Since the massing model is discretized into voxels, it is possible to make discrete decision-making, and the problem of zoning can be looked at in the form of a location-allocation problem. Due to this conversion of the abstract architectural problem into a mathematical problem, research from other engineering disciplines can be referenced.

Both of the approaches mentioned in the previous section successfully generated zoning options following the preferences specified for each zone. However, if both approaches are compared, then specific strengths and weaknesses can be observed.

In the Linear Assignment (concurrent growth) approach, the total benefit generated for all the agents combined is higher compared to the sequential assignment method mainly because of better access to the desirability lattice (fields) for the agents due to their limited size and efficient assignment. Thus, the LA approach can deal with conflicting contradictory requirements of multiple zones in an efficient manner. However, the topological structure of the zones generated by LA can be problematic due to staggered growth and incomplete disconnected zones. The primary reason for this is the increased number of interested agents for each zone and their virtually simultaneous (competitive) act of occupation. This is not a problem if there is no explicit requirement of zones. The best use case for this method is for large mixed-use programs with equal or near-equal priority for all the functions in the space program.

In the Sequential Assignment approach, the topological structure of the generated zones was uniform, and isolated islands were not found for any of the zones. The method could successfully create zones that can follow a particular growth pattern or form while still following the values from the desirability lattice. However, due to unequal access to the desirability lattice for the agents, the overall benefit for the zones was not as high as the LA approach. The zones which came later in the priority list performed poorly due to the unavailability of the desirable voxels for them. The best use case for this method is for smaller space programs or space programs with strong hierarchical importance for functions in the space program.

CONCLUSIONS

The method showcased in the paper was successful in generating zoning solutions considering multiple criteria together. Spatial considerations such as the generation of specific types of forms for the zones was also possible while maintaining the multi-criteria requirements. The computational agents were able to perform the design operations collaboratively while maintaining multiple spatial considerations. This approach invites potential extensions and instigates research questions for human-machine collaboration in generative design.

LIMITATIONS

In the test case stated, there was no closeness requirement between the zones themselves due to the nature of the space program; however, there are many cases where there is a vital closeness requirement between them (typically described as a graph called the Activity Relations Chart or REL Chart). This method can be adapted to solve those types of problems as well, as seen in (Soman, 2021), but it lies out of the scope of this paper. The main limitation of the proposed multi-agent configuration method is the interactivity between the agents in the system. Currently agents cannot unoccupy a voxel if another agent has a better score for the same position. Behaviours like negotiation can potentially solve the duality of the problem as identified in the comparison between the two methods. If both methods are to be compared reasonably, there could also be a quantification of the topological quality of the zoning solutions.

FUTURE WORK

The multi-agent system can be developed further in terms of the number of behaviors that the agents can perform, and the negotiation aspects between the agents before occupying voxels can be developed. Another promising research direction is implementing a [strict] scoring system for issuing rewards for the computational agents based on their actions. This can open up the possibilities of utilizing Reinforcement Learning models based on a formulation of this design process as a Markov Decision Process.

REFERENCES

Assari, A., Maheshand, T. and Assari, E. (2012) 'Role of public participation in sustainability of historical city: Usage of TOPSIS method', Indian Journal of Science and Technology, 5, pp. 2289–2294. doi: 10.17485/ijst/2012/v5i3.2.

Azadi, S. and Nourian, P. (2020) 'topoGenesis: Initial Pre-release'. Zenodo. doi: 10.5281/ZENODO.4006514.

Azadi, S. and Nourian, P. (2021a) Collective Intelligence in Generative Design: A Human-Centric Approach Towards Scientific Design. doi: 10.13140/RG.2.2.15295.84642.

Azadi, S. and Nourian, P. (2021b) GoDesign: A modular generative design framework for mass-customization and optimization in architectural design.

Dino, I. G. (2016) 'An evolutionary approach for 3D architectural space layout design exploration', Automation in Construction, 69, pp. 131–150. doi: https://doi.org/10.1016/j.autcon.2016.05.020.

Guo, Z. and Li, B. (2017) 'Evolutionary approach for spatial architecture layout design enhanced by an agent-based topology finding system', Frontiers of Architectural Research, 6(1), pp. 53–62. doi: https://doi.org/10.1016/j.foar.2016.11.003.

Koopmans, T. C. and Beckmann, M. (1957) 'Assignment Problems and the Location of Economic Activities', Econometrica, 25(1), pp. 53–76. doi: 10.2307/1907742.

Lobos, D. and Donath, D. (2010) 'The problem of space layout in architecture: A survey and reflections', arquiteturarevista, 6(2), pp. 136–161.

Perron, L. and Furnon, V. (no date) 'OR-Tools'. Available at: https://developers.google.com/optimization/.

Rittel, H. W. J. and Webber, M. M. (1973) 'Dilemmas in a general theory of planning', Policy sciences, 4(2), pp. 155–169. Available at: https://link.springer.com/article/10.1007/BF01405730.

Soman, A. P. (2021) 'GEN:ARCH: Platform for developing Architectural Configurations using Generative design Methodologies', TU Delft Repositories. Available at: http://resolver.tudelft.nl/uuid:c0065aed-6d44-4bee-b6f6-01a3c3f9380f.

Veloso, P. and Krishnamurti, R. (2020) An Academy of Spatial Agents: Generating Spatial Configurations with Deep Reinforcement Learning.

Watrobski, J. (2018) MCDA. Available at: http://mcda.it/.

Wątróbski, J. et al. (2019) 'Generalised framework for multi-criteria method selection', Omega (United Kingdom), 86, pp. 107–124. doi: 10.1016/j.omega.2018.07.004.

Zawidzki, M. (2016) Discrete Optimization in Architecture: Architectural \& Urban Layout. Springer.