Computational Design with Kurtboğaz: The Generation of Timber Structures with an Aggregative Design Algorithm

İlay Beylun Ertan[1], Pınar Çalışır Adem[2]

ORCID NO: 0000-0002-0823-41061[1], 0000-0001-6139-62892[2]

[1,2] Yeditepe University, Faculty of Architecture, Department of Architecture, İstanbul, Türkiye

This paper investigates the form-finding capacity of the traditional timber-joint construction method, Kurtboğaz, aiming to explore new architectural forms and possibilities through computational design techniques to preserve vernacular construction methods and integrate them into contemporary architecture. It presents an Aggregative Design Algorithm (ADA) that creates different structures based on designer rules and simple assembly rules of Kurtboğaz, leading to unique emergent forms through random rule application. The paper also explores how reinforcement learning, a type of machine learning, can improve this design process through a theoretical framework. The study tries to use a rule-based generative algorithm to explore the modular and reconfigurable characteristics of the Kurtboğaz. The ADA enables random rule application, leading to diverse forms. However, several challenges may be encountered during the application of ADA because of its random aggregation, such as self-collision and boundary detection. The study suggests using Reinforcement Learning (RL) in the ADA framework to address these problems. Incorporating RL is anticipated to enable the algorithm to adaptively learn and optimize the form-finding process, enhancing the performance and applicability of the Kurtboğaz method in contemporary architectural practice. In the future, with this generative process described by the study, designs that create spatial differences with the help of walls, floors, and rooms on a human scale can be realized. The study also plans to explore the synergy between craftsmanship and digital fabrication in the future.

Received: 10.07.2024
Accepted: 16.09.2024

Corresponding Author:
ilaybeylun.ertan@gmail.com

Ertan, İ.B. & Çalışır Adem, P. (2024). Computational design with Kurtboğaz: The generation of timber structures with an aggregative design algorithm, *JCoDe: Journal of Computational Design, 5*(2), 235-258. https://doi.org/10.53710/jcode.1514063

Keywords: Aggregative Design Algorithm, Traditional Architecture, Timber-Joint System, Kurtboğaz, Reinforcement Learning.

Kurtboğaz ile Hesaplamalı Tasarım: Birleştirici Tasarım Algoritması ile Ahşap Yapıların Üretimi

İlay Beylun Ertan[1], Pınar Çalışır Adem[2]

ORCID NO: 0000-0002-0823-41061[1], 0000-0001-6139-62892[2]

[1,2] Yeditepe Üniversitesi, Mimarlık Fakültesi, Mimarlık Bölümü, İstanbul, Türkiye

Bu makale, geleneksel ahşap geçme yapım yöntemi olan Kurtboğaz'ın form bulma kapasitesini araştırmakta ve bu yöntemi korumak ve çağdaş mimariye entegre etmek amacıyla hesaplamalı tasarım teknikleri aracılığıyla yeni mimari formlar ve olasılıklar keşfetmeyi hedeflemektedir. Çalışma, tasarımcı tarafından belirlenen hareket kurallarına ve Kurtboğaz'ın basit montaj kurallarına dayalı olarak farklı yapılar oluşturan ve rastgele kural uygulaması yoluyla öngörülemeyen formlar ortaya çıkaran Birleştirici Tasarım Algoritmasını sunar. Ayrıca bu çalışma, bir tür makine öğrenimi olan pekiştirmeli öğrenmenin bu tasarım sürecini teorik bir çerçevede nasıl iyileştirebileceğini araştırmaktadır. Kurtboğaz'ın modüler ve yeniden yapılandırılabilir özellikleri Kurtboğaz'ın form bulma kapasitesini incelemek için güçlü bir temel sağlamaktadır. Birleştirici Tasarım algoritması geleneksel mimariyi hesaplamalı tasarım ile yorumlamak suretiyle Kurtboğaz'ın basit yapım-montaj kuralları üzerinden sayısız kombinasyon oluşturma potansiyelini gösterir. Birleştirici Tasarım Algoritması, rastgele kural uygulamasını mümkün kılarak çeşitli biçimlerin oluşmasını sağlamaktadır. Ancak, Birleştirici Tasarım Algoritması'nın rastgele birleştirme nedeniyle uygulanmasında blokların üst üste çarpışması ya da sınırlar içinde kalma gibi çeşitli zorluklarla karşılaşılabilir. Bu zorlukların üstesinden gelmek için, bu çalışma Birleştirici Tasarım Algoritması çerçevesine Pekiştirmeli Öğrenme'nin (PÖ) teorik entegrasyonunu önermektedir. PÖ'nün entegrasyonu, algoritmanın form bulma sürecini uyarlamalı olarak öğrenmesini ve optimize etmesini sağlayarak, Kurtboğaz yönteminin performansını ve uygulanabilirliğini çağdaş mimari pratikler bağlamında artırabilir. Sonuç olarak çalışma Kurtboğaz'ın algoritmik kural tabanlı bir mantık içerisinde form bulma yeteneğini onun temel özelliklerini keşfederek incelemekte ve PÖ ile desteklenen bir tasarım süreci geliştirmektedir.

Teslim Tarihi: 10.07.2024
Kabul Tarihi: 16.09.2024

Sorumlu Yazar:
ilaybeylun.ertan@gmail.com

Ertan, İ. B. & Çalışır Adem, P. (2024). Kurtboğaz ile hesaplamalı tasarım: Birleştirici tasarım algoritması ile ahşap yapıların üretimi. *JCoDe: Journal of Computational Design, 5*(2), 235-258. https://doi.org/10.53710/jcode.1514063

Anahtar Kelimeler: Birleştirici Tasarım Algoritması, Geleneksel Mimari, Ahşap-Geçme Yapım Sistemi, Kurtboğaz, Pekiştirmeli Öğrenme

1. INTRODUCTION

Knowledge embedded in traditional architecture and construction methods persists through generations by enhancing regional expertise. This expertise, refined and passed down over time, has been tested in various conditions (Golden, 2017). However, traditional methods face challenges in the contemporary world due to the decline of master builders and the growing preference for new construction techniques. Additionaly, buildings constructed with traditional methods are vanishing due to neglect and modification. Traditional timber construction methods offer substantial potential for modularity, demountability and reconfigurability. There are many contemporary architects who get inspired by traditional architecture and use local materials and construction methods in their designs. For instance, Kengo Kuma (**Figure 1**) interpreted Japanese traditional methods of timber for his Pavilion in Paris. In the Swiss Sound Pavilion (**Figure 2**) Peter Zumthor's designs merge contemporary architecture with local materials and methods.

Figure 1: On the left: Kuma's Pavilion in Paris (Morby, 2015). **Figure 2:** On the right: Zumthor's Swiss Pavilion (Juergen, 2000).

In this context, this paper integrates traditional construction methods into contemporary architecture via computational medium. Traditional construction methods have a particular set of rules to generate building elements. The study focuses on how traditional construction methods, specifically the Kurtboğaz, can produce emergent forms using computational medium. In this regard, the study presents an Aggregative Design Algorithm (ADA) which is a rule-based generative algorithm with a combinatorial logic, to study form-finding capacity of the Kurtboğaz. The Kurtboğaz has modular and reconfigurable properties and a potential for generating many combinations with simple assembly rules. Thus, for the form-finding algorithm ADA, one of the timber-joint system, Kurtboğaz was selected as a design case. Since ADA works with a combinatorial logic and depended on the geometric calculations decided by the designer, different polyhedral geometries that are finite and discrete may be also used in this algorithm. In this context, the methodology of the study can be followed in **Figure 3.**

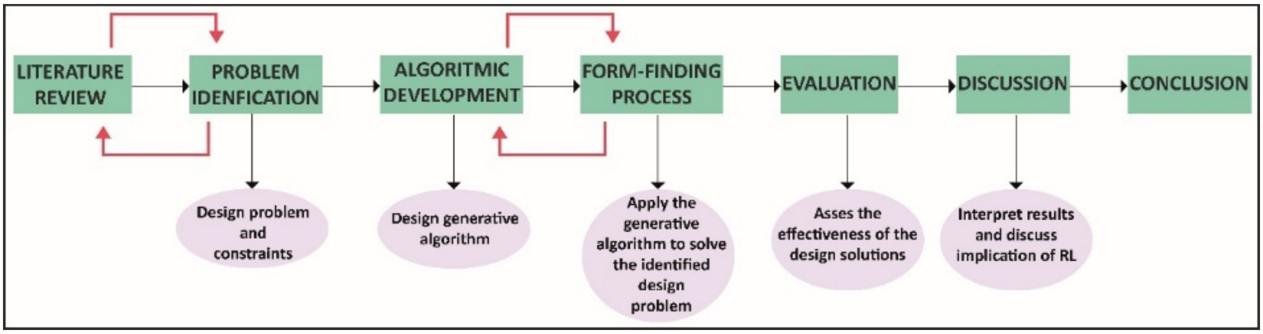


Figure 3: The methodology of the study.

2. COMPUTATIONAL TIMBER

Traditional architecture, which integrates natural elements, forms, and materials, can be an important model for modern architectural practices. Current approaches that engage with traditional architecture seek to understand and incorporate local values and experiences into contemporary designs. This approach actively values and preserves landscape and structural heritage. Innovative designs can be realized using existing technology, skills, and structures (Creangă et al., 2010).

While wood is one of the oldest construction materials, recent innovations in production techniques and design tools have introduced new formal, aesthetic, and structural possibilities, expanding its range of applications (Bianconi and Filippucci, 2019).

There are various studies which digital and parametrical models were used to control the geometry, assess the structural design, and help the management of wooden pieces' production, classification, and assembly. Digitalize the process means to gain a better control over each phase and procedures, transferring the earlier ideas and sketches into an engineered knowledge (Kuma et al., 2019).

For instance, Kuma et al. (2019) introduced KODAMA, a sophisticated wooden structure that forms a "porous sphere" (a polyhedron) constructed from a single type of wood section, assembled without the use of nails or screws. The design, development, and construction processes were iterative, involving a balance of theoretical validation and computational experimentation to optimize the wooden elements, their dimensions, the choice of material, the connections, and, ultimately, the mechanical performance and overall impact of the structure. The wooden façade of the Japan Pavilion at EXPO Milano 2015- Wooden Byobu, designed by Atsushi Kitagawara Architects, exemplifies a digitally crafted wood structure assembled without screws or nails, utilizing a "compressive-tension" effect. This concept draws inspiration from traditional Japanese woodworking techniques and the craftsmanship of intricate wooden toys. The design process reimagines the tradition of Japanese wooden construction, exploring geometries, spatiality, and the interplay of light and shadow, starting from small-scale physical models (Kitagawara et al., 2019).

A comprehensive review by Ottenhaus et al. (2023) presents studies on design principles that promote adaptability in timber buildings through design for disassembly and reuse. They also examine reversible timber connection systems that facilitate adaptability and disassembly. Hua et al. (2022) points out the knowledge embedded in the old modular structures and gets inspired by the East Asian timber architecture to produce re-configurable, modular timber frame with a scheme for re-using of timber elemets for another structure. Adel et al. (2018) concentrate on methods for designing non-standard timber frame structures, facilitated through united multi-robotic fabrication at the building scale. Österlund and Wikar (2019) presents various case studies representing non-linear, fluid timber structures with their compution and fabrication processes. Retsin (2019, p.8), introduces discrete architecture and “computational parts” along with the physical, material and economic features of this paradigm. Although discrete architecture does not specifically focus on timber structures, the logic of traditional timber interlocking structures to create a whole from parts can highlight discrete architecture to digitise traditional timber structures. Xu et al. (2023) explores computational design and the shape grammars to create flexible timber architecture with Mortise-tenon joints, re-imagining Chinese timber frames for modern architecture.

This study focuses on traditional timber-joint method, Kurtboğaz, which we see instances of in Turkey's traditional architecture and tries to compute the features of this joint system to produce various spatial volumes for modern architecture. By focusing on a local and disappearing case, the study tries to mould it into a discrete design system. In this regard, this study, especially inspired by the works of KODAMA (Kuma et al., 2019) and Wooden Byobu (Kitagawara et al., 2019), aims to produce complex geometries from discrete simple parts of Kurtboğaz, while trying to create an open algorithmic flow in which machine learning can be integrated in the future, trying to control geometry with computational tools in wooden architecture and trying to contribute to the literature in search of emergent forms.

3. THE KURTBOĞAZ

In the Eastern and Central Black Sea Region, geographical conditions have necessitated the development of construction technology based on local materials (Akbaş, 2019). The primary building material in the region is wood. Wooden masonry structures can be easily dismantled and reassembled in another location if needed (Tuna, 2008). The wooden construction culture has been passed down through generations via the master-apprentice relationship (Akbaş, 2019). The wooden masonry structures in the region can be made directly from logs or from logs that have been processed into lumber. These structures are assembled by fitting the wood together at specially prepared joints at the corners. The notches prepared at the corners to tightly hold the wood together are called "boğaz." The process of stacking wooden blocks on top of each other to align with these notches is known as "geçme." The wooden pieces that are joined together at these joints do not come apart (Tuna, 2008). In these wooden masonry structures, which are anchored to the ground using different foundational techniques, the walls constructed by stacking wood on top of each other are load-bearing (Akbaş, 2019). According to Tuna (2008), there are different names for the jointing methods depending on the physical dimensions and craftsmanship of the wooden block used:

Karaboğaz: This jointing method uses wood in log form.
Kurtboğaz: In this method, the logs that will form the solid walls are processed into smooth lumber, and the joints are crafted precisely and uniformly. Since the interlocking process is done correctly in the Kurtboğaz method, there are no gaps in the wall, and no plastering is

required. This method, commonly seen in the Black Sea Region, is utilized in the "wooden masonry wall" system (Akbaş and Özcan, 2008). **Çalmaboğaz:** When wooden blocks are not of the desired size, shorter pieces are joined together with the help of posts. The Çalmaboğaz technique is mostly used for partitioning interior spaces (Orhan and Çavuş, 2019).

In this study, the Kurtboğaz method, which is frequently used in the local wooden masonry architecture of the Black Sea Region, is chosen. It is more sophisticated than the Karaboğaz method, more suitable for single-piece construction and scaling than the Çalmaboğaz method, and simpler than other jointing methods (such as mortise-and-tenon). It requires no plastering and leaves no gaps on the façade. The study will explore the potential of this method to produce different forms through a more minimal interpretation, leveraging its nature of being constructed by stacking layers.

To summarize, Kurtboğaz is a technique can be seen in traditional Turkish timber craftsmanship where building elements are interlocked into eachother in a specific way without nails. Interlocking system without nails and the form of the timber elements make Kurtboğaz adaptable, easily assembled and disassembled. Thus, Kurtboğaz is a modular construction method and facilitates the reuse or recycling of materials by allowing easily disassembly and separation of the building elements. Kurtboğaz may be applied in different contexts due to its modularity and the elements of the system can be reused without extensive demolition and reconstruction. This study tries to use the Kurtboğaz method to produce integrated building elements and enclose a single-volumetric space. ADA has a system where structural elements are created by stacking or layering components on top of each other. Therefore, it is suitable for the design of masonry structures. The modular nature and stacked construction method of Kurtboğaz make it well-suited for ADA.

Figure 4: Kurtboğaz detail of an ambar (Özgüner, 2018).

The Kurtboğaz is used on the walls of the ambar, houses, mosques and watermills (**Figure 4**). There are different dimensions about the construction of Kurtboğaz in the literature. While the thickness of the wood to be used varies between 2-10 cm, the protrusion distances of the wooden pieces from the corner points vary between 15-50 cm (Özgüner, 2018; Sözen and Eruzun, 1992; Akbaş, 2015). Additionally, Akbaş (2015) states that if there is a gap of 1.5 cm at the joints, we can ensure the the system's strength.

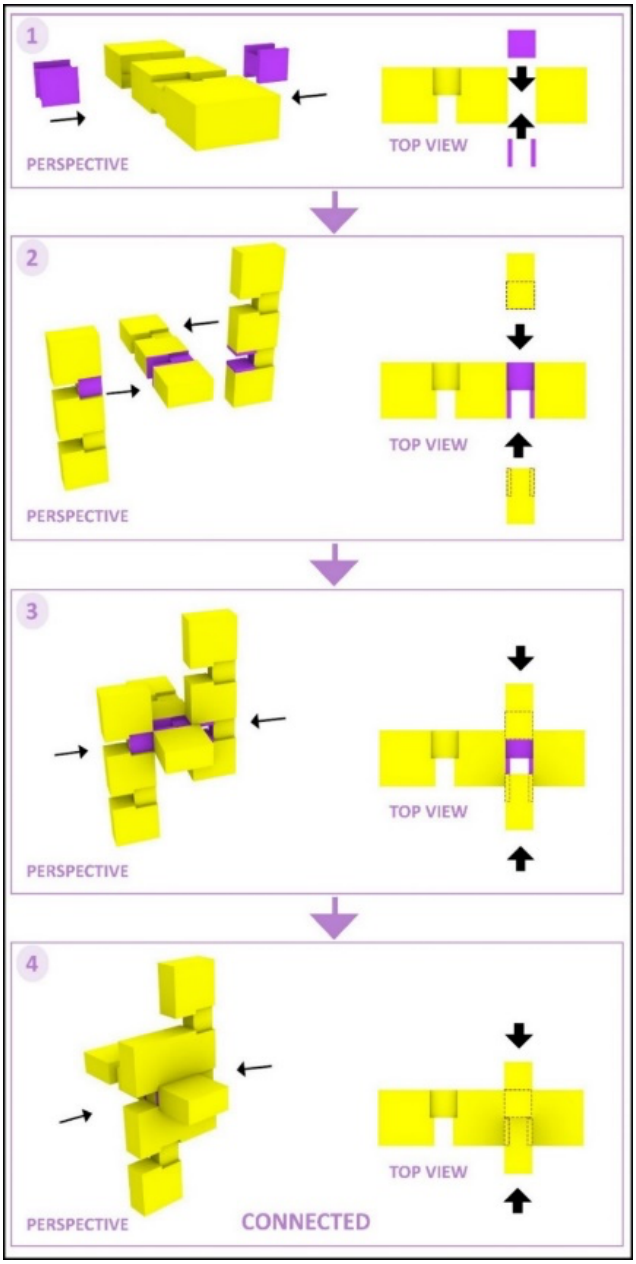


Figure 5: Connection mechanism of Kurtboğaz.

The study develops its building blocks by incorporating reference measurements. It designs building blocks that are 18 cm wide, 66 cm long, and 9 cm thick. The blocks are half-filled and half-empty at the joints, with four joint connections featuring two half-filled and two half-empty points. The half-empty parts merge with the half-filled parts, creating a more secure lock due to the 1.5 cm gaps in the filled sections. These dimensions become important when defining connection points and movement rules, as they determine how much the blocks can be moved. When different dimensions are applied, the movement rules should be adjusted according to the new dimensions (see other trials). The study designs and models these Kurtboğaz building blocks using Rhino software (**Figure 5**).

3. AGGREGATIVE DESIGN ALGORITHM FOR KURTBOĞAZ

In this paper, a selected traditional timber-joint construction method, the Kurtboğaz, will be reinterpreted in a rule-based generative computational process, which is called the Aggregative Design Algorithm (ADA), to produce a variety of architectural forms. ADA exhibits properties of infinite, fluid, and diffuse growth, capable of working with finite and discrete geometric volumes.

The development of the algorithm is empirical, involving numerous trials and experiments to organize the algorithmic flow. The authors benefit from the recursive function techniques of the Plethora Project's (Url-2) and Danil Nagy's (Url-3) code. In addition, the authors overview approaches of Danil Nagy's (Url-3) and Jake Hebbert's (Url-4, Url-5) code for using the Random module. However, ADA integrates knowledge from these references with its techniques to offer a novel solution for Kurtboğaz. The designer determines the types of connections and action rules based on the geometric volume's structure and its aggregation method. ADA randomly selects a connection type and an action rule in each iteration, resulting in different and unique combinations. These selections are made through a recursive function, causing the number of geometric volumes to increase with each iteration.

Aggregate Design Algorithm (ADA) is a Python-based class structure. There are various platforms for architects to create modular aggregations of discrete units, such as WASP (Rossi and Tessman, 2017) and Monoceros (Url-1). Monoceros utilizes Wave Function Collapse algorithm to generate wholes from smaller units and WASP uses iterative user-defined rules for geometric operations to create assemblies. The ADA got inspired WASP, but it is specifically designed for design explorations of simple timber-joint systems, such as Kurtboğaz. It is a custom algorithmic process that tries to compute a construction technique that is based on manual labour and manpower. Since the rules of Kurtboğaz, user-defined connection rules and geometric calculations are printed as an output, the working system of ADA is very open and easy to grasp. Because of printing the outputs of the design process, it is also suitable for reinforcement learning reward-punishment calculations. In the future studies, the authors will try to integrage reinforcement learning into this process.

The study develops the ADA using the GhPython component in Rhino/Grasshopper (**Figure 8**). The study begins by defining the input values of the GhPython component for ADA. The first one is the geometric volume "blockdouble" selected by the designer. For this study, the geometric volume is a four-joint kurtboğaz building block, but ADA is an algorithm that can work with more than one geometric volume. Second, it defines "numbers_of_ConnectionTypes" to control the number of connection types. This input value is the number of iterations set by the designer. As the number value changes, the number of building blocks of the aggregation increases or decreases. Finally, the study defines "seed" to create a different aggregation combination for each number change. In this input value, each "seed" number value represents a different aggregation combination. (**Figure 6**).

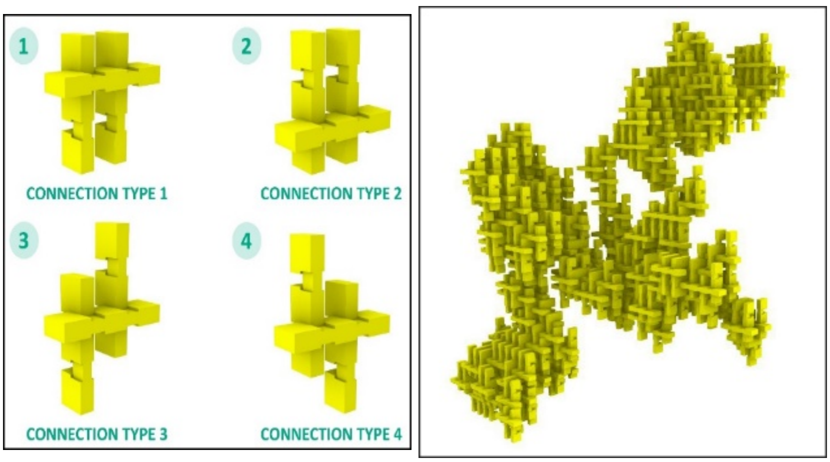


Figure 6: On the left: The connection types designed by the authors; on the right: a large aggregation.

In the pseudocode of ADA (**Figure 7**), *The Aggregative* class starts with basic functions to handle surfaces (e.g., exploding surfaces, getting centroids, duplicating borders). *FromCopytoOrient* function handles the creation and orientation of copies of Kurtboğaz blocks based on certain surface indices, which are passed as arguments. *LoopAggregation* is a recursive function which aggregates the Kurtboğaz blocks defined by connection types, applies transformations, and updates the brep list. It uses a combination of copying, orienting, and moving objects in 3D space. Finally, the *Main Execution* initializes variables, creates an instance of *Aggregative*, and processes the *LoopAggregation* to form the final output.

Figure 7: The pseduocode of ADA.

```
import rhinoscriptsyntax, random
class Aggregative():
  Initialize with an optional general_rules_list
  Function explodeSurface(brep):
    Explode brep into PolySrf
    Return PolySrf
  Function centerPoint (brep):
    Get the surface at given index from explodeSurface(brep)
    Get the centroid of the surface
    Return the centroid point
  Function surfaceBorderLines (brep):
    Get the surface at given index from explodeSurface(brep)
    Duplicate its border and explode into curves
    Return the relevant curve
  Function midPoint(index, brep):
    Get the curve from surfaceBorderLines(brep)
    Get and return the midpoint of the curve
  Function onlyCopy(brep):
    Copy the brep object
    Return the copied object
  Function fromCopytoOrient(brep, index_pairs):
    Initialize an empty list
    For each pair of indices in index_pairs:
      Get the center point and midpoint for each index
      Store the points in a list
    Create a copy of brep
    Orient copies using these point pairs
    Append the copies and oriented objects to the list
    Return the list
  Function loopAggregation(brep, brepList, repeated_number, number_x, number_y, number_z):
    If repeated_number > 0:
      Set vector to (number_x, number_y, number_z)#determines the rules for moving
      Create and move geometry objects from multiple `fromCopytoOrient` calls
      Store these geometries in rulesGeometry
      Select a random geometry and update brepList
      Determine next move based on a random action value
      Print rulesGeometryList and rulesMovingList
      Recursively call loopAggregation with updated parameters
Main Execution:
  Initialize blockList and position variables
  Seed the random generator
  Create an Aggregative instance
  Call loopAggregation and append result to blockList
  Extract general rules into ListConnectionTypesNames and ListActionRules from general_rules_list
```

ADA can be applied to all finite and discrete geometric volumes. It iteratively aggregates geometric volumes by performing the required computations. The study evaluates ADA in three stages. The first one is the formal calculation of geometric volumes to create connection types. The designer determines the appropriate surfaces, the necessary edges and points of these surfaces with the help of ADA to create connection types according to the characteristics of the geometric volumes. ADA manipulates the geometric volumes with these descriptions. This study performs these operations on a single Kurtboğaz building block. The second stage is the creation of connection types. In this stage, the designer decides on the various connection types of the geometric volumes with different orientation actions, which is done by ADA's connection type generation method. This study has four connection types for the Kurtboğaz building block. The final stage is the iteration and aggregation of the geometric volumes. At this stage, ADA uses a function method that calls itself. This method contains the previously defined connection types and the action rules that allow movement in the designer's x, y, and z directions. In the study, for the Kurtboğaz building block, the designer decides the action rules to eight. The method randomly selects a new connection type and action rule at each iteration. In this study, each random selection adds three Kurtboğaz building blocks to the aggregation. This process results in aggregation blocks.

3.1 Form Generations without a Boundary

Each unique ADA seed value produces a distinct combination of randomly selected connection types and action rules. In the initial experiment, the study aims to compare these Kurtboğaz structures to illustrate their differences. By changing the seed value, various combinations of connection types and action rules are tested **(Figure 9)**. The study also investigates the generation potential of ADA with different geometric volumes. The second experiment uses a standard geometric volume, the cube. The third experiment tests a different timber joint method, mortise and tenon. In both experiments, the study makes one hundred generations with seed values from one to one hundred and evaluates ten of these one hundred generations. The results show that, similar to the Kurtboğaz building block, ADA can create different combinations with different seed values by making random selections. The ability to generate similar outcomes using both a standard geometric volume and a different timber joint method, as interpreted by a designer, demonstrates that ADA can work harmoniously with finite and discrete geometric volumes.

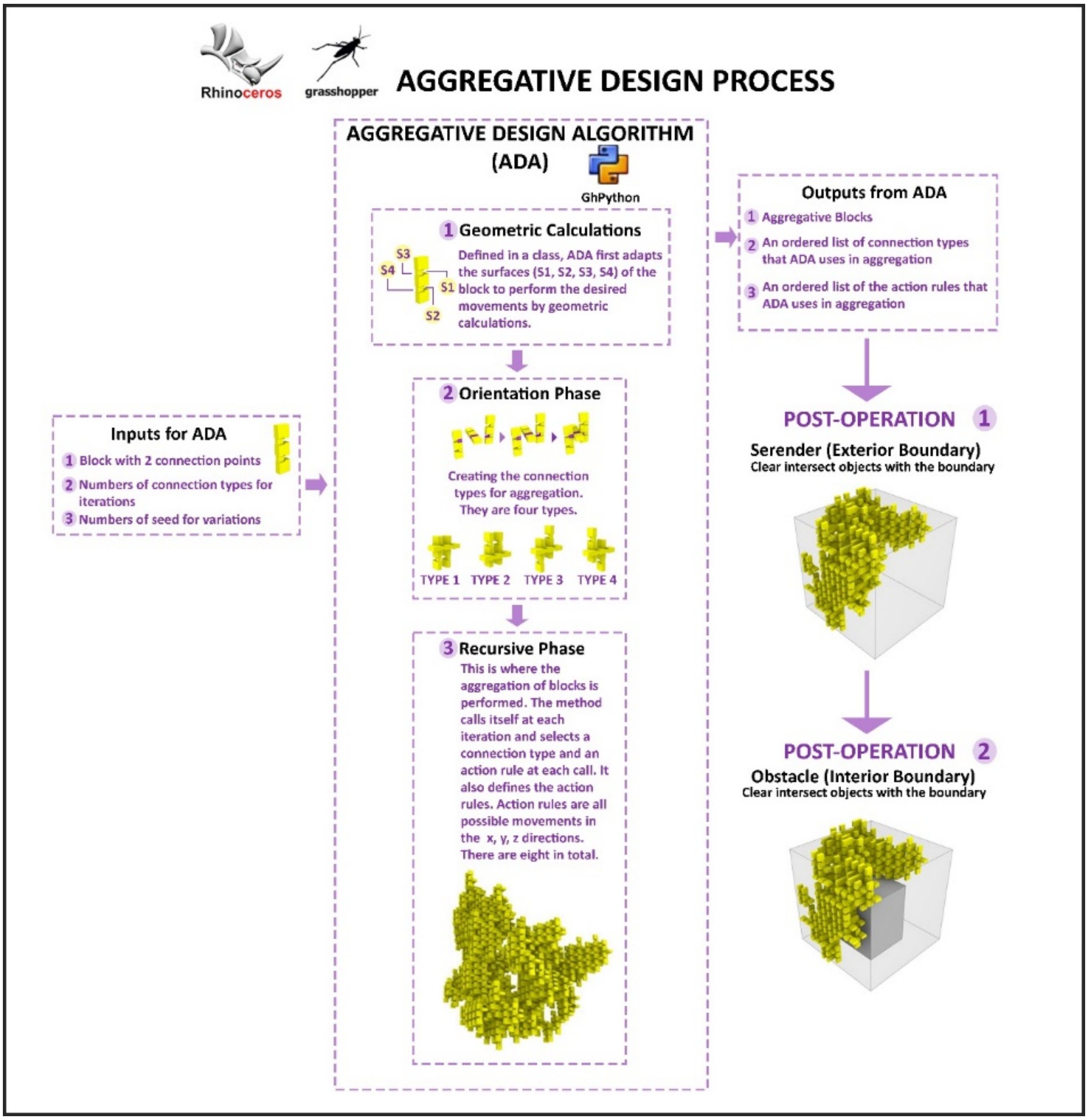


Figure 8: Aggregative Design Process

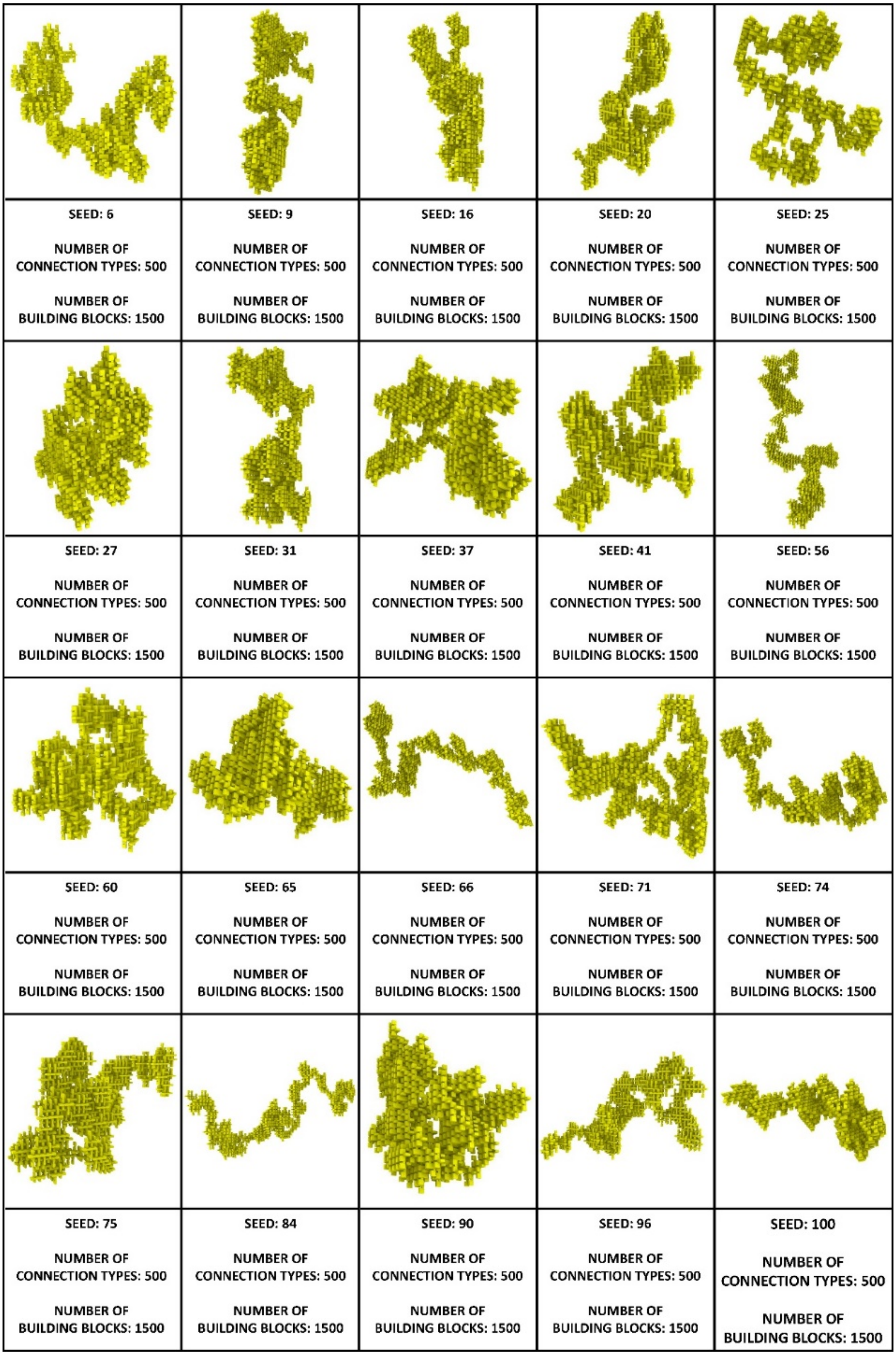
3.2 Form Generations within a Boundary

In this experiment, the operation of the ADA within certain volumetric boundaries was tested. The study shows the effect of each different number of ADA seed values on the form-finding process and tries seed values from one to one hundred (**Figure 10**, top). It uses a total of five different exterior and interior boundaries (**Figure 10**). In the various experiments with different geometric volumes, ADA randomly selects connection types and action rules at each iteration, completing all experiments in five hundred iterations. In the Grasshopper algorithm, the study operates various post-operations for ADA to fit aggregated forms within certain volumetric boundaries. In another experiment

(**Figure 10**, bottom), the authors present the five examples of the seed value:90. The number of connection types are "50", "100", "150", "200", "250", "300", "350", "400", "450" and "500". The counterparts of these numbers in terms of building block numbers are "150", "300", "450", "600", "750", "900", "1050", "1200", "1350" and "1500". The randomly aggregated forms generated by ADA fit within certain boundaries by geometric post-operations performed in the Grasshopper. Since ADA chooses connection types and action rules randomly and geometric boundaries are fixed in this random growth, volumetric boundaries may not always cover adequate building blocks. As the authors will mention in the discussion section, this issue can be solved by adjustments to the Grasshopper interface or the random selection nature of ADA. However, in this study, a theoretical framework with RL is discussed to solve the aggregation issue within a given boundary.

To sum, ADA can generate an infinite and unorganized generative process using Kurtboğaz building blocks, with each aggregated form being unique. While some generations undergo a highly disorganized diffusion process, others follow a cumulative one. The study also explores the form-finding process with ADA for various exterior and interior boundaries. Due to ADA's random selection, it is challenging to use the building blocks efficiently within certain boundaries, often leading to many blocks aggregating outside the exterior boundary and thus not participating in the form-finding process. Additionally, the generated forms may encounter collisions, self-intersections, or assembly problems. The results indicate that, despite different exterior and interior boundaries, similar patterns of increases and decreases in the number of building blocks are observed every fifty iterations.

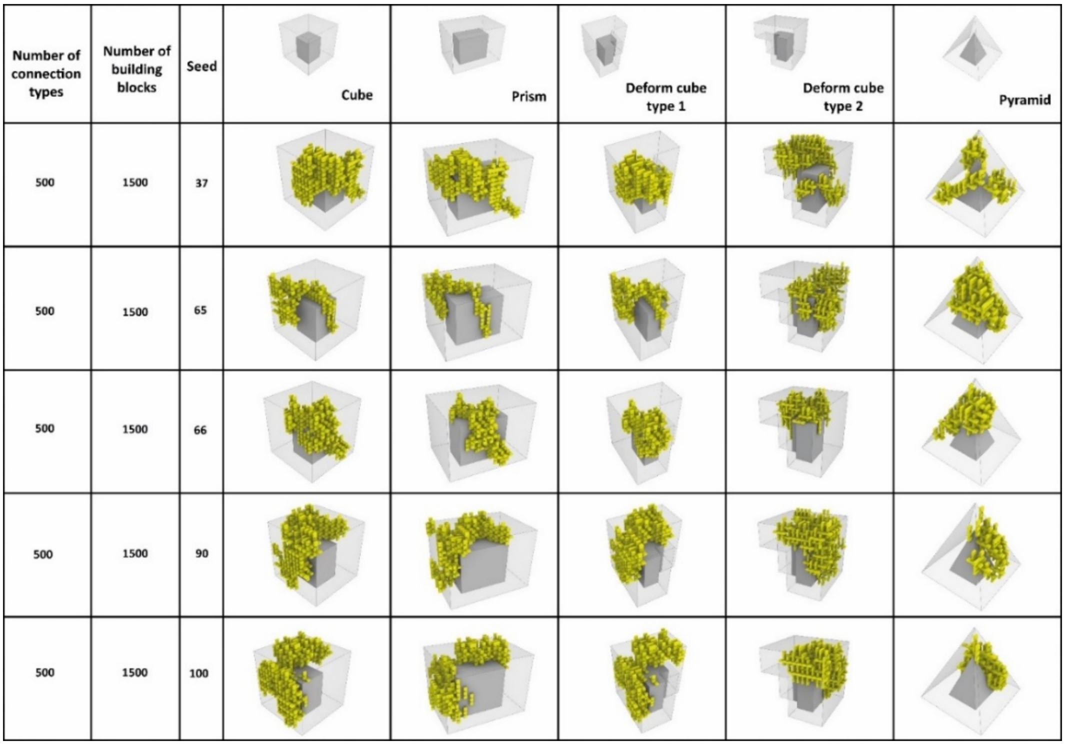
Figure 9: ADA generations without booundary.


SEED: 6
NUMBER OF CONNECTION TYPES: 500
NUMBER OF BUILDING BLOCKS: 1500
SEED: 9
NUMBER OF CONNECTION TYPES: 500
NUMBER OF BUILDING BLOCKS: 1500
SEED: 16
NUMBER OF CONNECTION TYPES: 500
NUMBER OF BUILDING BLOCKS: 1500
SEED: 20
NUMBER OF CONNECTION TYPES: 500
NUMBER OF BUILDING BLOCKS: 1500
SEED: 25
NUMBER OF CONNECTION TYPES: 500
NUMBER OF BUILDING BLOCKS: 1500
SEED: 27
NUMBER OF CONNECTION TYPES: 500
NUMBER OF BUILDING BLOCKS: 1500
SEED: 31
NUMBER OF CONNECTION TYPES: 500
NUMBER OF BUILDING BLOCKS: 1500
SEED: 37
NUMBER OF CONNECTION TYPES: 500
NUMBER OF BUILDING BLOCKS: 1500
SEED: 41
NUMBER OF CONNECTION TYPES: 500
NUMBER OF BUILDING BLOCKS: 1500
SEED: 56
NUMBER OF CONNECTION TYPES: 500
NUMBER OF BUILDING BLOCKS: 1500
SEED: 60
NUMBER OF CONNECTION TYPES: 500
NUMBER OF BUILDING BLOCKS: 1500
SEED: 65
NUMBER OF CONNECTION TYPES: 500
NUMBER OF BUILDING BLOCKS: 1500
SEED: 66
NUMBER OF CONNECTION TYPES: 500
NUMBER OF BUILDING BLOCKS: 1500
SEED: 71
NUMBER OF CONNECTION TYPES: 500
NUMBER OF BUILDING BLOCKS: 1500
SEED: 74
NUMBER OF CONNECTION TYPES: 500
NUMBER OF BUILDING BLOCKS: 1500
SEED: 75
NUMBER OF CONNECTION TYPES: 500
NUMBER OF BUILDING BLOCKS: 1500
SEED: 84
NUMBER OF CONNECTION TYPES: 500
NUMBER OF BUILDING BLOCKS: 1500
SEED: 90
NUMBER OF CONNECTION TYPES: 500
NUMBER OF BUILDING BLOCKS: 1500
SEED: 96
NUMBER OF CONNECTION TYPES: 500
NUMBER OF BUILDING BLOCKS: 1500
SEED: 100
NUMBER OF CONNECTION TYPES: 500
NUMBER OF BUILDING BLOCKS: 1500

4. DISCUSSION

In the process of ADA’s generations, designer defines the connection types and action rules, and ADA randomly chooses these two main parameters. Thus, ADA’s randomization of connection types and action rules leads to infinite and unorganized generations. Self collision and the number of parts to form a space within the specified boundaries (usable block number) are two of the important problems of efficiency. In form generation processes where boundaries are not defined (**Figure 9**), it was calculated that 25-45% of the parts in the resulting aggregative forms self-collided. The self-collision rates in parts with seed numbers 56, 71, and 84 are lower compared to the others. While there is vertical and linear growth in seed 56, there is horizontal and linear growth in seed 84. In seed 71, linear growth occurs in four directions: left, right, up, and down. In similar forms like seed 9 and seed 16, it is observed that in formations where the direction of linear and form flow changes, such as in seed 9, the number of repeating parts is lower.

When specific internal and external boundaries are introduced in form generation to achieve spatial differentiation (**Figure 10**, top and bottom), the issue of unused parts remaining outside the boundaries arises. When different combinations are produced with different seed numbers to examine the percentages of unusable building blocks outside the exterior boundary (**Figure 10**, top), it was calculated that 40-80% of the parts remain outside the exterior boundary in cube-prism and deformed cube 1 and 2. This rate increases in pyramid forms that narrow in a specific direction. Looking at the increasing number of parts within the same seed number, it was also calculated that more parts remain outside the boundary in the pyramid compared to other spatial limits (**Figure 10**, bottom). When looking at the number of parts used to enclose the space and serve as the shell of the volume, no significant difference was observed among the various formal boundaries, although it was slightly lower in the pyramid form. These problems in ADA can cause weaknesses in ensuring spatial closure or in the structural integrity of the form. These issues can be resolved by imposing constraints on ADA's random rule selections. However, in this study, instead of solving these problems through adjustments in the algorithm, to solve these challenges, the study offers the theoretical framework of Reinforcement Learning (RL)-supported ADA. The RL-supported aspect is the second phase of this study and will be implemented in future research.



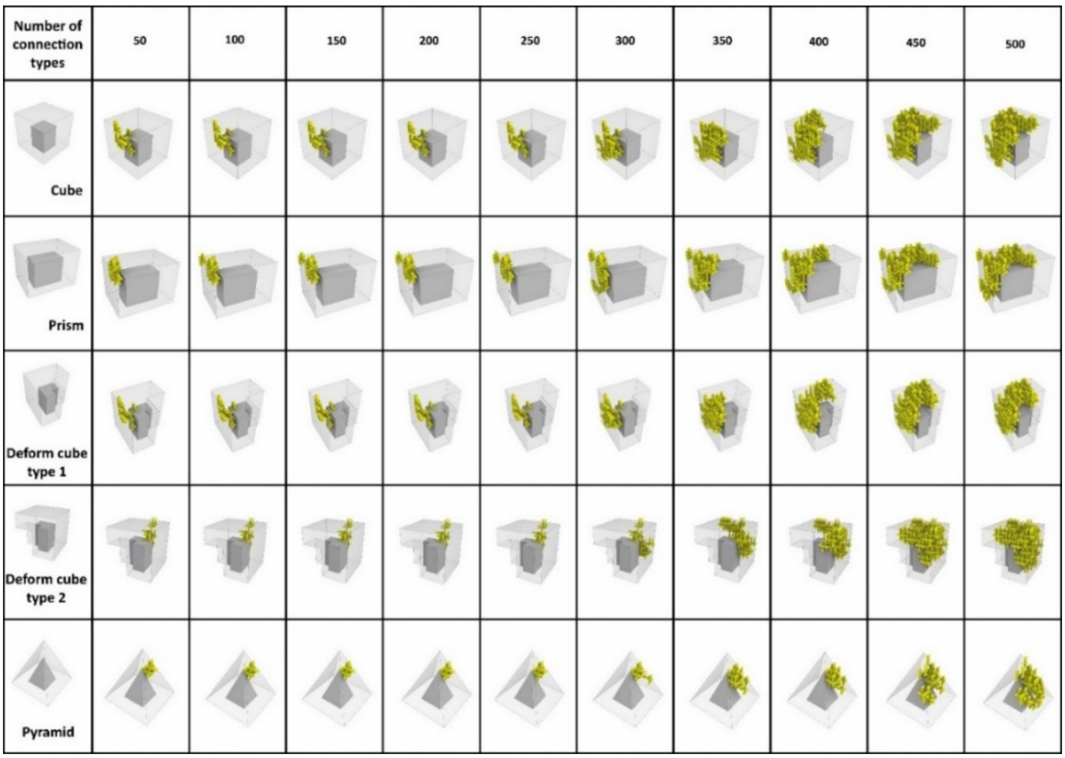


Figure 10: On the top: different seed value generations within different boundaries; on the bottom: increasing number of blocks within the same seed value.

Several form-finding trials demonstrate that ADA has significant potential to generate diverse combinations by randomly selecting connection types and action rules. However, because of the random aggregative process, the ADA process encounters design issues such as self-collision. Additionally, while the study conducts form-finding processes with the Kurtboğaz building block within boundaries, ADA's infinite and unorganized nature due to its random selections prevents the efficient use of the building blocks. To address these problems, designers can intervene to the process in several ways. For instance, they can modify ADA by evaluating and editing the generated forms, such as by making manual additions and deletions. Adding checking algorithms to detect and correct self-collision issues can also be effective. Optimization methods can be tested for boundary detection, and alternative design scenarios can be created to handle collisions and adjust generations accordingly. Additionally, structural issues can be addressed by testing with structural analysis tools. On the other hand, the machine learning (ML) algorithms, which have been included in Generative Design (GD) in recent years, can help the designer and maximize the generative potential of ADA. This study proposes a theoretical framework for supporting ADA with reinforcement learning (RL) algorithms, one of the ML methods.

In parallel with the rapid developments in machine learning, the integration of reinforcement learning algorithms into generative design has also accelerated. Wang and Snooks (2021) aim to exploit the generative potential of reinforcement learning by interpreting the spatial and structural system of Le Corbusier's Domino in a new form using a Random Walk model. In their three-dimensional study, Lye and Andrasek (2021) suggest using reinforcement learning algorithms to find complex and high-performance combinations of connection states of discrete components. Huang (2021) tries to combine complex geometries with reinforcement learning. The three-dimensional study uses waste plastic chips. Wibranek et al. (2021) focus on the problem of finding the arrangement of blocks in a sequence that can relate to different combinations. The study arranges the sequence along a given curve and utilizes SL Blocks. Reinforcement Learning-based generative models can work in 2-dimensions and develop plans or elevation proposals.

In the future studies, this paper aims to achieve three objectives by offering a Reinforcement Learning (RL)-supported framework**.** First, it seeks to produce optimal results using the identified outer and interior boundaries. Second, it aims to address the issues caused by the Aggregative Design Algorithm's (ADA) random selection of connection types and action rules. Third, it intends to integrate an autonomous design process into generative design by leveraging RL's intelligent, intuitive, and predictive nature. The RL-supported framework can enhance ADA's capabilities through autonomous processes and solve its design problems by evaluating observations and selecting the best actions. Accordingly, the framework can conduct the form-finding process with ADA within set boundaries using rewards and punishments and correctly aggregating building blocks.

5. CONCLUSION

This study evaluates the traditional timber joint method, Kurtboğaz, using computational tools to link traditional and contemporary architecture. By examining the Kurtboğaz's key features and presenting its form-finding capabilities within an algorithmic rule-based logic, the paper develops a generative design process enhanced by machine learning theoretically.

While several trials have shown ADA's ability to generate diverse aggragations through random selections, they have also highlighted design issues such as self-collision. Additionally, ADA's infinite and unorganized nature has led to inefficient use of building blocks within set boundaries. To address these, in the future studies, the study proposes evaluating ADA within a theoretical framework supported by reinforcement learning (RL) algorithms. The RL-supported theoretical framework has three objectives: first, to achieve an efficient form-finding process within defined external and internal boundaries; second, to resolve ADA's inherent design issues; and third, to establish an autonomous generative process that aids the designer.

The study outlines how RL can theoretically support this generative process to create a more autonomous and optimized system. It holistically integrates building parts with the Kurtboğaz construction method, exploring its potential as a modular and contemporary approach. By describing this traditional method within an RL-supported autonomous framework, the study combines historical knowledge with computational design tools to develop contemporary and sustainable architectural solutions, preserving traditional construction methods while fostering emergent forms.

Conflict of Interest Statement

The manuscript sent has not been presented at any meeting before. The manuscript is entitled “Computational Design with Kurtboğaz: The Generation of Timber Structures with an Aggregative Design Algorithm” has not been published elsewhere and that it has not been submitted simultaneously for publication elsewhere.

Author Contribution

This study is based on the master thesis entitled " THE GENERATION OF TIMBER-JOINT STRUCTURES WITH AN AGGREGATIVE DESIGN ALGORITHM" by İlay Beylun Ertan under the supervision of Assist. Prof. Dr. Pınar Çalışır Adem (Yeditepe University Graduate School of Natural and Applied Sciences, Dept. of Architecture, Istanbul, Turkey, 2024). The second author contributed as a guiding and supervising expert in the scientific and editorial process of the paper. The computational work and algorithms in the paper were implemented and analysed by the first author.

References

Adel, A., Thoma, A., Helmreich, M. & Gramazio, F. (2018). Design of Robotically Fabricated Timber Frame Structures, In Ed.: Anzalone, Phillip, Del Signore, Marcella John Wit, Andrew, *Proceedings of the 38th Annual Conference of the Association for Computer Aided Design in Architecture* (pp. 394-403), Universidad Iberoamericana, Mexico City.

Akbaş, G. (2015). Geleneksel Yapım Teknikleri ve Mekan İlişkisi: Uzungöl ve Taşkıran Örnekleri. (Master’s Thesis, Atılım University).

Akbaş, G. (2019). Vernaküler Mimaride Süreklilik ve Kurtboğaz Tekniği İlişkisi. In Erçetin, A., Aydemir, D. (Eds.), *Geleceğin Dünyasinda Bilimsel ve Mesleki Çalışmalar 2019: Mimarlik ve Tasarım*, (1-10). Ekin Basım Yayın Dağıtım.

Akbaş, G., Özcan, Z. (2018). Yapım tekniği farklılıklarının mekana yansıması: Uzungöl ve Taşkıran örneği. *ATA Planlama Ve Tasarım Dergisi,* 2(2), 47-58. https://dergipark.org.tr/tr/pub/ataplanlamavetasarim/issue/41536/471228

Bianconi, F., Filippucci, M. (2019). Digital Wood Design. *Lecture Notes in Civil Engineering (24).* Springer Nature. https://doi.org/10.1007/978-3-030-03676-8_51

Creangă, E., Ciotoiu, I., Gheorghiu, D., Nash, G. (2010). Vernacular architecture as a model for contemporary design. *WIT Transactions on Ecology and the Environment*, *128,* 157-71. https://www.researchgate.net/publication/271438016_Vernacular_architecture_as_a_model_for_contemporary_design

Golden, EM. (2017). *Building from tradition: local materials and methods in contemporary architecture*. Routledge.

Hebbert, J. (2013, September 17). *NOC Python Grasshopper: 01* [Video]. YouTube. https://www.youtube.com/watch?v=Kyi_K85Gsm4

Hebbert, J. (2013, September 17). *NOC Python Grasshopper: 02* [Video]. YouTube. https://www.youtube.com/watch?v=FnBKISoPRAE&t=11s

Hua, H., Hovestadt, L. & Li, B. (2022). Reconfigurable Modular System of Prefabricated Timber Grids. *Computer-Aided Design, 146*, May 2022, 2-13. https://doi.org/10.1016/j.cad.2022.103230.

Huang, CH. (2021, March). Reinforcement Learning for Architectural Design-Build: Opportunity of Machine Learning in a Material-informed Circular Design Strategy. In *Proceedings of the CAADRIA* [Volume. 1]. pp: 171-180.

Juergen, G. (2000). *Swiss Sound Pavilion*. zumthor.bjorkan. Retrieved 2024, June 6. https://zumthor.bjorkan.no/project/pavilion/

Kuma, K., Imperadori, M., Clozza, M., Hirano, T., Vanossi, A., & Brunone, F. (2019). *KODAMA: A polyhedron sculpture in the forest at Arte Sella: Innovative techniques of representation in architectural design.* http://dx.doi.org/10.1007/978-3-030-03676-8_51

Kitagawara, A. & Imperadori, M. & Kuwabara, R. & Brunone, F. & Matsukawa, M. (2019). Wooden Byobu. *From Architectural Façade to Sculpture: Innovative Techniques of Representation in Architectural Design.* http://dx.doi.org/10.1007/978-3-030-03676-8_52

Lye, J., Andrasek, A. (2021). Machine Learning Combinatorial Frameworks for Architecture. *International Journal of Innovation and Economic Development*, 7(2), 20-29. https://doi.org/10.18775/ijied.1849-7551-7020.2015.72.2002.

Monoceros. (2021). *Monoceros.* Retrieved September 2, 2024, from https://monoceros.sub.digital/

Morby, A. (2015, October 20). *Kengo kuma installs climbable wooden pavilion in Paris park*. Dezeen. Retrieved 2024, June 6, https://www.dezeen.com/2015/10/20/kengo-kuma-installs-climbable-wooden-yure-pavilion-jardins-des-tuileries-paris-fiac/

Nagy, D. (2014, March 2). *Python scripting in Grasshopper – Scripting form* (2/2) [Video]. YouTube. https://www.youtube.com/watch?v=Za5qPUeb0Mo

Österlund, T., & Wikar, M. (2019). Freeform timber structures: Digital design and fabrication. In M. Hudert & S. Pfeiffer (Eds.), *Rethinking wood: Future dimensions of timber assembly* (pp. 133-150). Birkhäuser.

Ottenhaus, L. M., Yan, Z., Brandner, R., Leardini, P., Fink, G., & Jockwer, R. (2023). Design for adaptability, disassembly and reuse: A review of reversible timber connection systems. *Construction and Building Materials, 400* (2023), 2-16. https://doi.org/10.1016/j.conbuildmat.2023.132823

Orhan, F., Çavuş, A. (2019). Artvin ilindeki geleneksel kırsal mesken mimarisine bir örnek: ahşap yığma evler. *1st İstanbul International Geography Congress Proceedings.* Istanbul University Press. https://doi.org/10.26650/PB/PS12.2019.002.042

Özgüner, O. (2018). *Köyde mimari doğu karadeniz.* Dergah Yayınları.

Plethora Project. (2013, May 22). *PYTHON in Grasshopper 02: Recursive aggregation* [Video]. YouTube. https://www.youtube.com/watch?v=3Z4YVCOk02k

Retsin, G. (2019). Discrete architecture in the age of automation. In G. Retsin (Ed.), *Discrete: Reappraising the digital in architecture. Architectural Design, 89*(2), 7-13. John Wiley & Sons.

Rossi, A., & Tessmann, O. (2017). Designing with digital materials: A computational framework for discrete assembly design. In P. Janssen, P. Loh, A. Raonic, & M. A. Schnabel (Eds.), *Protocols, flows and glitches: Proceedings of the 22nd Conference of The Association for*

Computer-Aided Architectural Design Research in Asia (CAADRIA). Hong Kong.

Sözen, M., Eruzun, C. (1992). *Anadolu'da ev ve insan.* Emlak Bankası & Creative Yayıncılık.

Tuna, C. (2008). *Orta Karedeniz Bölgesi Sahil Kesiminde Geleneksel Mimari*. Arkeoloji ve Sanat Yayınları.

Wang, D., Snooks, R. (2021). Artificial intuitions of generative design: an approach based on reinforcement learning. In *Proceedings of the 2020 DigitalFUTURES: The 2nd International Conference on Computational Design and Robotic Fabrication (CDRF 2020).* pp: 189-198. Springer Singapore.

Wibranek, B., Liu, Y., Funk, N., Belousov, B., Peters, J., Tessmann, O. (2021). Reinforcement learning for sequential assembly of SL-blocks-self-interlocking combinatorial design based on machine learning. In *Proceedings of the 39th eCAADe Conference [Volume. 1]* (pp: 27-36).

Xu, Q., Teixeira, F. F., & Shafiei, M. (2023). Frameworks in flexible timber design systems combining mortise-tenon joints and discrete timber blocks. In W. Dokonal, U. Hirschberg, & G. Wurzer (Eds.), *Proceedings of the 41st Conference on Education and Research in Computer Aided Architectural Design in Europe, Volume 1 – Digital Design Reconsidered* (pp. 283-292), Graz University of Technology. https://doi.org/10.52842/conf.ecaade.2023.1.283