Proceedings of the IASS 2024 Symposium
Redefining the Art of Structural Design
August 26-30, 2024, Zurich Switzerland
Philippe Block, Giulia Boller, Catherine DeWolf,
Jacqueline Pauli, Walter Kaufmann (eds.)

Augmented Morphological Designs Enabled Through Network-Based Genetic Algorithms

Paul R. KALNITZ*, Shai S. SHEN-ORR[a], Aaron SPRECHER [b]

* Material Topology Research Laboratory, Faculty of Architecture and Town Planning, Technion – Israel Institute of Technology
Amado Building, Technion City, Haifa, 32000, Israel
paul.kalnitz@campus.technion.ac.il

[a] Faculty of Medicine, Technion-Israel Institute of Technology
[b] Material Topology Research Laboratory, Faculty of Architecture and Town Planning, Technion – Israel Institute of Technology

Abstract

Biologically inspired techniques have by now entered the mainstream of the architectural design process [1]. An example is the genetic algorithm (GA), inspired by Darwinian evolution. Genetic algorithms in the AEC field can optimize a wide variety of performative solutions, including minimizing structural mass. GAs assume one-gene-to-one-trait [2] and thus can operate on scalar parameters. However, due to this assumption, the powerful combinatory nature of genetic assemblies is lost. Therefore, GAs are limited in what they can model [3]. Here we present an enhanced version of the GA, called eNEAT (Enhanced Neuroevolution of Augmented Topologies) [4] which is based on neural networks and machine learning. By evolving networks rather than simple scalar parameters, eNEAT has the capacity to model combinations of genes, thereby simulating more closely the way DNA regulates the construction of natural organisms. As applied to architecture, this allows for complexity and hierarchical morphology, responsive to parameters that are internal to the system itself and external to the environment [5]. The non-linear aspect of eNEAT allows for the manifold attributes of the built form to be developed concurrently in a unified phase, rather than via the conventional sequential architectural design process. By simultaneously considering previously independent parameters, architectural responses to these forces become dependent upon each other and may influence one another directly. Along with internal requirements, in combination with advanced environmental modeling tools [6], eNEAT provides a powerful platform for considering epigenetic effects, the influence of environmental factors. The far-reaching conclusion of this project is the potential of developing a computational protocol that is capable of negotiating multiple types of independent information in the designing of the architectural object.

Keywords: architectural design, hierarchical morphology, evolutionary development, genetic algorithm, neuroevolution

1. Introduction

While commenting on the current state of computational architecture, architect George Legendre remarks in his *Book of Surfaces*: “Parametric relationships are not parts (...) Thus a form shaped by parametric modulation has no discrete limb to speak of... you cannot chop it into pieces, nor indulge in the separate application of permutation, substitution, and scaling of parts.” [7] This is due to the reductionist nature of the GA which is based on the outmoded biological theory, namely one-gene-to-one-trait. [8] A truly evolutionary model would overcome this limitation, and instead support a hierarchical morphology. It would navigate a space of design possibilities vastly exceeding what has previously been available in architecture.

The model of computational evolution of architecture, as proposed in this article, offers such a possibility. It is modeled after the mechanism of evolutionary development (evo-devo). Evo-devo has revealed that a common set of genes related to the regulation of the growth and development of organisms is shared across very distantly related organisms. Furthermore, "in the natural world, modular design is the key to diversity. Nature never starts from scratch. Rather, it builds up complexity from a low-level toolbox of components." [9] Recent discoveries in biology have uncovered multi-layered networks that regulate organic form generation - gene regulatory networks, protein interaction networks, cell-signaling networks, metabolic networks, and ecological networks [10]. The emergence of the field known as epigenetics is also a critical factor in influencing architectural form. Epigenetics describes the relationship between the built form as being environmentally responsive in a complex feedback loop. It integrates gravity, load, solar radiation, wind, natural light, thermal loading, ventilation, etc. [11].

The algorithm proposed in this paper augments the standard GA by replacing populations of simple scalar data structures ("chromosomes") with populations of networks (genomes). The system remains reflective of the neo-Darwinist concepts of mutation, crossover, and natural selection, but it becomes capable of modeling combinations of genes allowing for vast possibilities of form, as found in nature.

This system offers the possibility to evolve rich, novel morphologies. Due to its inherent non-linearity, the framework integrates multifunctional performance. It becomes possible to model manifold architectural systems concurrently, rather than via the conventional sequential process of designing components independently.

This article first describes a few selected historical studies in the field of genetics as related to architecture. It then describes the current state-of-the-art in terms of the application of GAs and advanced geometric modeling. It further details eNEAT, the augmented version of the GA, and its protocol. A first experiment is presented investigating the evolution of structural trusses according to mutation, crossover, and selection. Preliminary results of emergent truss topologies are evaluated.

2. A selected view of evolutionary architecture

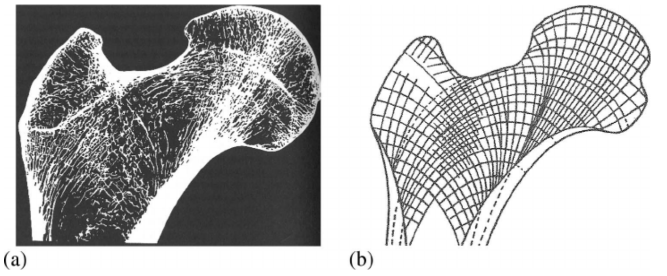

Figure 1. Diagrams of Force which have influenced morphological evolutionary changes (Thompson 1917)

Beginning with English naturalist Charles Darwin and his theory of evolution, the explanation for species growth and development is accounted for. Essentially the survival of a species is determined by adaptability [12]. Scottish biologist and mathematician D'Arcy Thomson (figure 1) described evolutionary change as the negotiation of pressure, due to both internal forces and changing external environmental conditions, as is the case in architecture.

Subsequent to these historical scientific milestones, five successive waves of evolutionary architecture may be observed. The first wave at the turn of the 19th century, organic architecture, was influenced by the idea of an inner logic that tended to bring unity to a design. In the second wave, during the post-war period, architecture was inspired by theories of cybernetics and feedback loops. The third wave, beginning in the 1980s, saw the early integration of computer technologies such as animation and genetic algorithms. Beginning also in the mid-1980s, the development of the Bézier spline curve and NURB technology ushered in the fourth wave, now known as parametricism [10]. In parallel to the fourth wave, the fifth wave of evolutionary architecture may have been heralded by architect John Frazer who stated that "DNA does not describe the phenotype but encompasses instructions, the genotype, that describe the process of building the phenotype." [13]

2.1 The Genetic Algorithm

The GA, invented in 1975 by computer scientist John Holland, was developed to solve problems by simulating the evolutionary process of natural selection. In nature, DNA is a non-linear network of interconnected genes whose combinations regulate the outward traits of organisms. The GA is a reductive version of DNA based on the outmoded theory of one-gene-to-one trait. Its original design was as a classifier system. Functioning as a supervised machine-learning system, it referenced a database that contained a large array of attributes, and categories each of which contained a subset of the attributes. An object was classified based on how closely its attributes matched one of the categories. As a linear system applied to architecture, the GA models a direct mapping of parameters to building attributes. It can adapt and optimize but to a very limited extent because its underlying model is non-hierarchical. The space of possible solutions generated by GAs is constrained to basic variations of the original form. The space of possible solutions generated by GAs is constrained to basic variations of the original form. Architect John Frazer was a pioneer in the use of the GA to optimize form. As an example, in his book, *Evolutionary Architecture*, he investigated the evolution of classical proportions of Tuscan columns (figure 2).

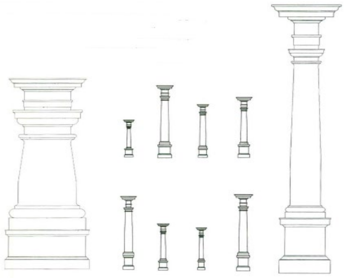

Figure 2. Evolving proportion in Tuscan columns using genetic algorithms (Frazer 1995)

2.2 Geometric form

The C-chair, developed by the authors in 2008, was an early experiment in applying evolutionary development protocol to architectural design and in the formation of complex geometries (figure 3). [14] The elements of the chair consisted of networked nodes and edges. It was composed of two topologies - a surface of concentric circular forms and a tree-like support. As in evo-devo, both topologies shared common attributes, therefore the interface formed a series of connecting points. Rather than a GA, this model incorporated a K-Means clustering algorithm. The model was purely formal; it did not consider materiality or structural characteristics. It raised an important question. Rather than mathematical precision and complex formulaic geometries, can the organizing principles of nature be applied to the development and evolution of architectural forms?

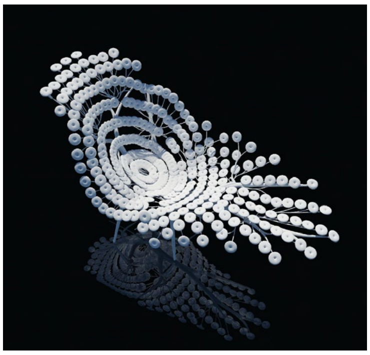

Figure 3. C-chair, an experiment in evo-devo inspired geometry (Kalnitz, Sprecher 2008)

2.3 Hierarchical Form

Moving from a series of connections of two distinct topologies to a recursive hierarchical morphology, the series below (Figure 4) illustrates a sequence of increasing morphological complexity. The series begins with two basic forms, a ‘cap’ and a ‘tube’ (figure 4a). They are simple forms which can be characterized by parametric formulae. As such, their regulating parameters can be manipulated in a linear manner by a GA – stretching, growing, shrinking, etc. Next in the series is a recursive hierarchy of the ‘cap’ form (figure 4b), with caps connected to other caps, in a fractal-like manner. The various dimensions of the caps are parametric, and the connection points are specified algorithmically. This recursive hierarchical form requires a non-linear model which is not supported by the GA. The third form in the series illustrates a further level of complexity by combining multiple subcomponents in a recursive hierarchical manner (figure 4c).

Genetic algorithms optimize forms resizing and reshaping self-same components. [12] However, it is not possible to create independent forms which can interact with each other. There is no possibility of recursive hierarchical modules. [3] To model hierarchical morphologies, an improvement of the GA is required.

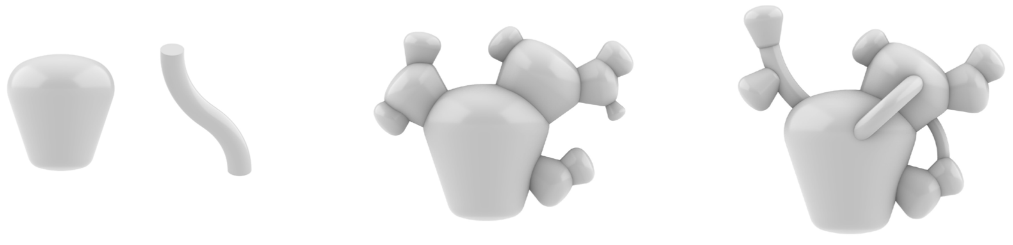

Figure 4a Figure 4b Figure 4c

Figure 4. A series of increasing morphological complexity.
4a) Two parametric forms 4b) Recursive hierarchy of one subcomponent 4c) Recursive hierarchy of multiple subcomponents

3. Introducing NEAT

The NEAT algorithm (NeuroEvolution of Augmenting Topologies), was originally developed in 2002 by computer scientists K. Stanley and R. Mikkulainen at the University of Texas. [4] NEAT was developed as a way of increasing the efficiency of machine learning systems. Rather than fixed-topology networks, NEAT reconfigures dynamic-topology networks as a way of evolving solutions. The non-linear networked protocol of NEAT models hierarchical relationships. It has been shown that NEAT offers the possibility of both optimizing and *complexifying* solutions simultaneously. It is unique because structures become increasingly more complex as they become more optimal, strengthening the analogy between computational and natural evolution.

The following diagrams (figures 5a-c) illustrate the difference between the standard GA and NEAT.

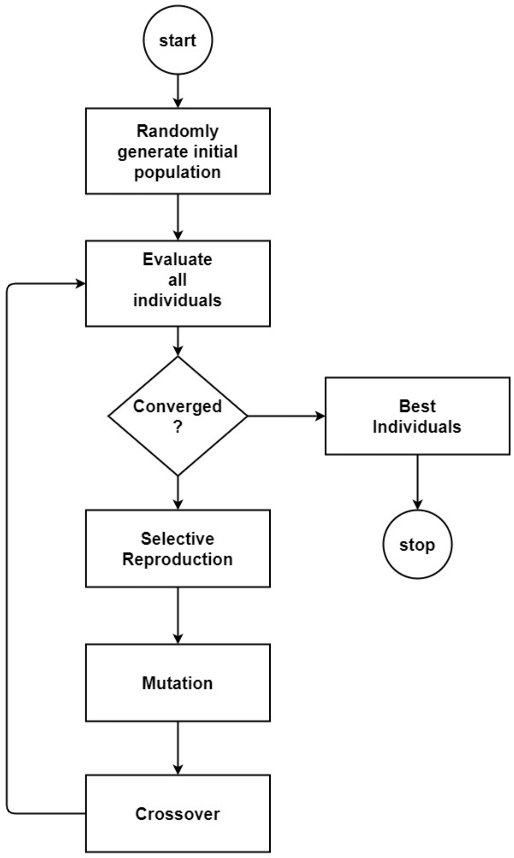

Figure 5a: The genetic algorithm flowchart (Kalnitz, MTRL 2024)

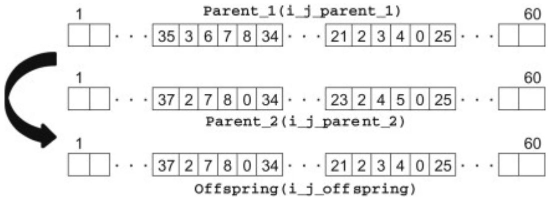

Figure 5b: The standard GA chromosome containing simple scalar values. (Carroll, 2005)

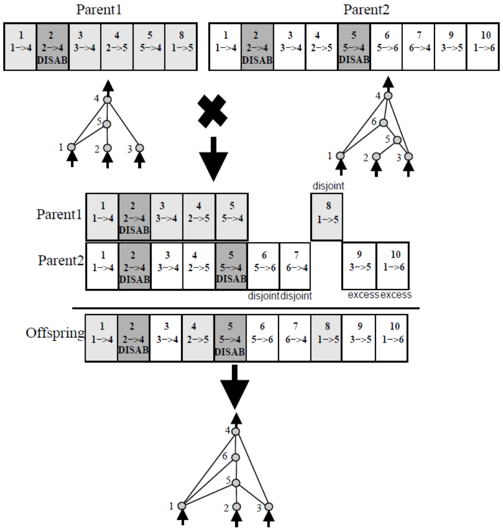

Figure 5c: The NEAT chromosome containing non-linear networks (Stanley and Mikkulainen, 2024)

The NEAT algorithm reuses part of the neo-Darwinian GA algorithm, namely an initial population of possible solutions, an evaluation of fitness, a selection, crossover, and mutation (figure 5a). In addition, it addresses the major constraint of the GA, its linearity. GAs manipulate scalar-based data structures (“chromosomes”) that are limited in what they can model. [3] An example shows the crossover of two scalar genomes where the right half of parent 1 and the left half of parent 2 combine to form the offspring genome (figure 5b). Illustrating the crossover of the NEAT genome, two parents are networks instead of arrays. The parent genomes are combined in a more sophisticated way that closely resembles the recombination mechanism found in DNA. Genes are aligned and the dominant ones are inherited (figure 5c). [17] The NEAT model gains the capacity to model combinations of genes, in the same way that DNA is a non-linear multi-layered combinatory system.

3.1 NEAT as a modified NN

This section describes the adaptation of the data structure of the standard neural network (NN) to NEAT, in order to model the evolution of a physical architectural object.

A traditional static NN uses backpropagation for learning and contains fixed input and output layers, hidden nodes, directed weighted edges, and sigmoid activation functions (figure 7). A dynamic NEAT network is based on a traditional NN (figure 8). The key modification is that hidden nodes and edges are dynamic – they can be added and deleted, thus augmenting the mechanism of machine learning. It has been shown that evolving dynamic NNs can significantly enhance their performance. [4]

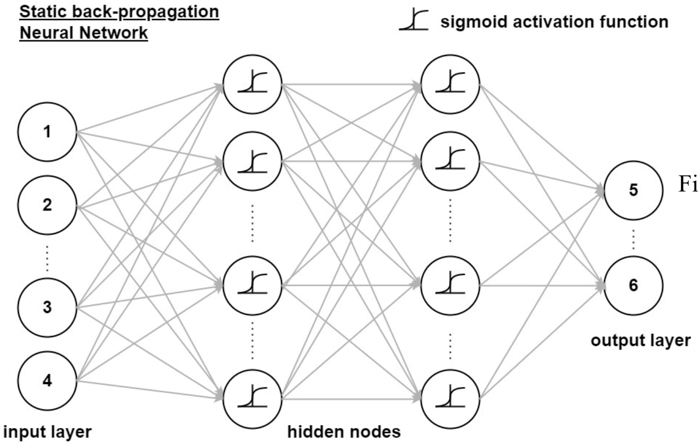

Figure 5b

Figure 7: A standard static NN with four layers
(Kalnitz, MTRL 2024)

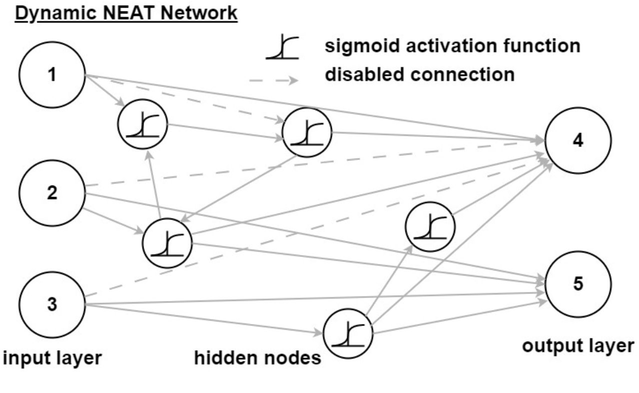

Figure 8: NEAT NN with a dynamic inner layer
(Kalnitz, MTRL 2024)

3.2 eNEAT adapted to the evolution of a truss

To model the evolution of structural trusses, NEAT has been repurposed and shall be renamed eNEAT. All of the advantages that NEAT offers in the modeling of evolution have been retained, with two modifications. Rather than purely virtual mathematical entities, eNEAT's networks are analogous to genomes in nature, containing abstracted information that regulates the outward form. Nodes no longer contain traditional activation functions. Rather, they are repurposed to encapsulate information such as the coordinates of the truss load points, support points, joints, and their associated degrees of freedom. Edges are also repurposed. While edge weights in a traditional NN hold virtual information learned via backpropagation, the repurposed edge weights contain embedded physical information such as the mass of each truss member.

The next two diagrams illustrate the relationship between an eNEAT network functioning as a genotype (figure 9), and its corresponding 2-dimensional truss, its phenotype (figure 10). Nodes, corresponding to truss joints, are labeled 'a' through 'f' and contain the X, and Y coordinates of each joint. Edges represent the connecting truss members and their mass. For instance, truss member a-c starts at joint 'a' located at (0.5, 1.0) and connects to joint 'c' at (0, 0), with a weight of 1.8kg.

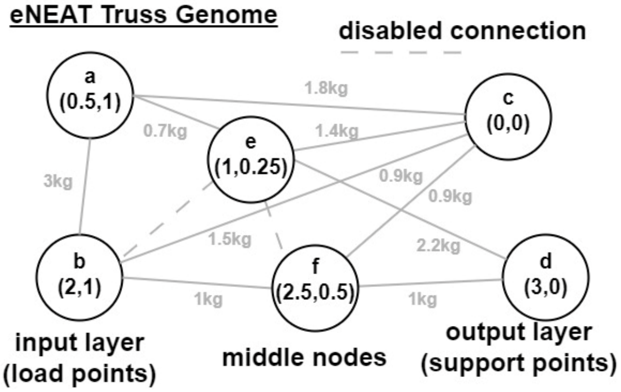

Figure 9: A 2-dimensional truss genome
(Kalnitz, MTRL, 2024)

In the corresponding truss (figure 10), the labels of the joints ('a' through 'f') and the coordinates of each joint correspond to the labeled nodes in the genome network (figure 9). Also, the weights of each truss member correspond to the edge weights of the genome.

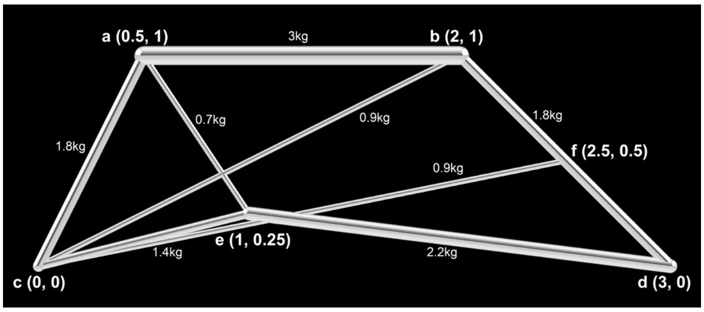

Figure 10: A 2-dimensional truss phenotype
(Kalnitz, MTRL, 2024)

Darwinian evolution, analogous to the optimization or adaptation of trusses, involves spawning successive generations with the goal of minimizing the overall mass of the population of truss solutions. As generations are created, a percentage of parent trusses are mutated by adding and deleting nodes (joints) and edges (truss members). Crossover of two high-performing parent trusses also results in child trusses. The offspring are then evaluated using a finite-element-analysis engine that calculates the mass of each child truss. A distance metric is used to categorize the resulting trusses into species. This is done to protect novel forms so that they can survive during the evolution process. Eventually, the evolutionary algorithm converges to a set of solutions that are generally close to optimal, a result that is similar to gradient-based optimization algorithms. The next section details the high-level protocol of the eNEAT algorithm which implements the mechanism of evolution.

3.3 The eNEAT protocol

The high-level protocol of the repurposed eNEAT algorithm is illustrated below (figure 11). The eNEAT algorithm is implemented as a proprietary Grasshopper component, written in C#, based on SharpNEAT. [15] It interacts with the physics engine, Karamba, which is used for the evaluation of trusses using the finite- element-method. [16]

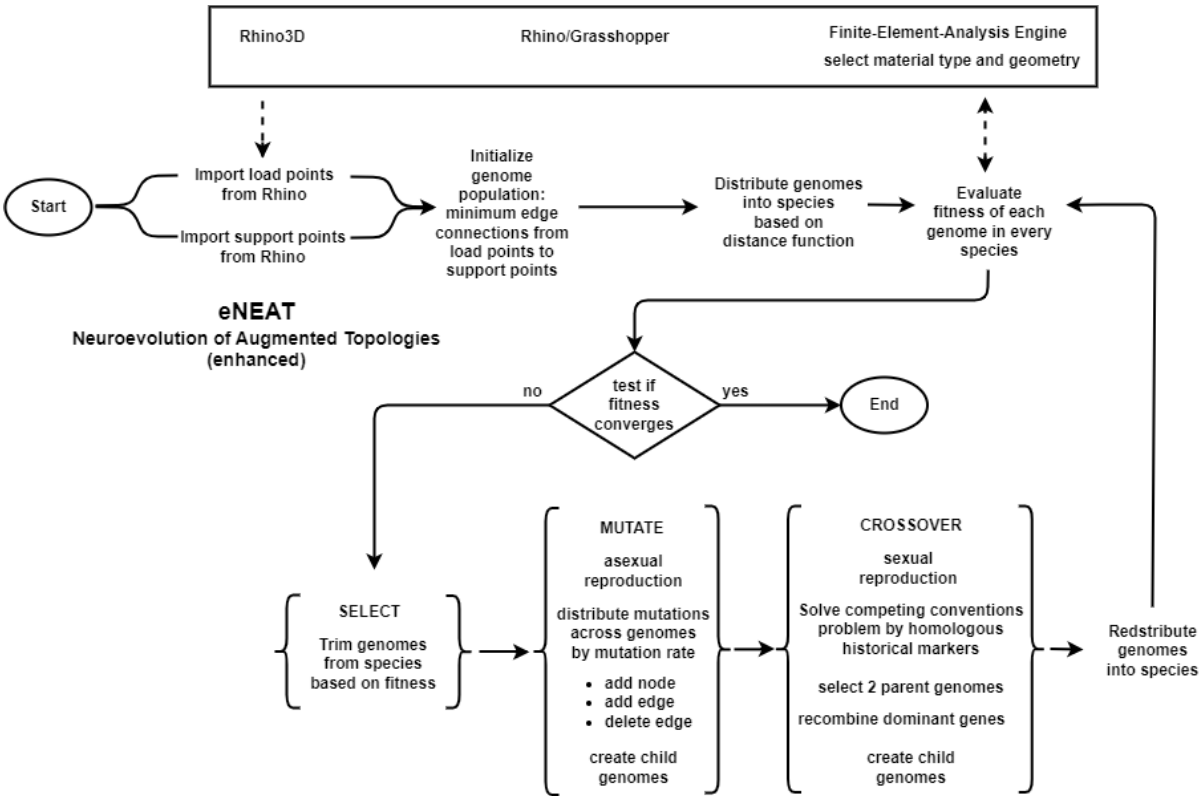
Rhino3D
Rhino/Grasshopper
Finite-Element-Analysis Engine
select material type and geometry
Start
Import load points from Rhino
Import support points from Rhino
Initialize genome population: minimum edge connections from load points to support points
Distribute genomes into species based on distance function
Evaluate fitness of each genome in every species
eNEAT
Neuroevolution of Augmented Topologies (enhanced)
no
test if fitness converges
yes
End
SELECT
Trim genomes from species based on fitness
MUTATE
asexual reproduction
distribute mutations across genomes by mutation rate
• add node
• add edge
• delete edge
create child genomes
CROSSOVER
sexual reproduction
Solve competing conventions problem by homologous historical markers
select 2 parent genomes
recombine dominant genes
create child genomes
Redstribute genomes into species

Figure 11: The high-level protocol of eNEAT
(Kalnitz, MTRL, 2024)

In a typical evolutionary run, the designer first specifies the load and support points and the materiality, cross-section, degrees of freedom, etc. of the desired truss. A population of trusses is initialized with each individual having random edge connections that traverse all of the input points. The population is categorized into species and all trusses are evaluated. At each generation the population is tested for convergence – has the mass stabilized to within a specified threshold? If not, trusses with the lowest mass are selected, and subjected again to mutation and crossover, thus creating the next generation. Once the model converges or reaches a specified number of generations, the algorithm stops and a set of close-to-optimal trusses emerges.

4. Truss Experiment #1

This experiment aims to create a first iteration of a series of architectural objects with structural capabilities as a product of an evolutionary-based NEAT model. The specific goal is to create an evolutionary lineage that minimizes the overall mass of 2-dimensional and 3-dimensional trusses.

4.1 Experimental Protocol

In this experiment, a truss is evolved with the objective of finding a minimum mass, while supporting a certain load and with given support points. The basic algorithm first creates a population of random solutions. Each solution is then evaluated by a finite-element-analysis engine to calculate the mass of each truss in the population. Once all individuals are evaluated, the best performers are categorized into species, with a distance function determining their classification. The best performers then go on to form the next generation. A percentage remains as is, and others are subjected to various mutations and crossovers. Rates of mutation and crossover, types of mutations, population size, maximum number of generations, and materiality are some of the parameters that can be used to tune the algorithm to arrive at reasonable results.

Illustrating the evolutionary sequence of a truss, its minimal weight is 116.04 kg due to a delete mutation (figure 12a). Its parent truss weight is 138.0kg due to another delete mutation (figure 12b). The initial truss weight is 210.98kg (figure 12c).

The matrix below depicts three species horizontally and three generations vertically. Trusses g and h crossover to form truss d. Truss h mutates to e and then to b through mutation (addition and deletion of edges). Trusses j, f, and c are novel formations during the evolution process (figure 13).

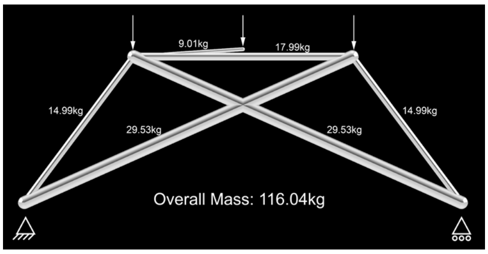

Figure12a: Minimal truss.

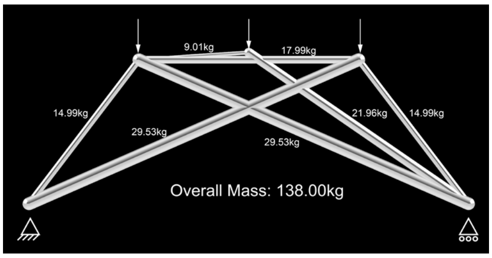

Figure 12b: Evolving truss with less members.

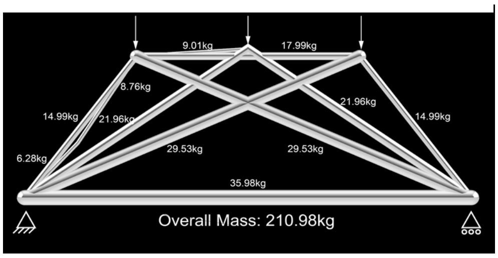

Figure 12c: Initial truss configuration.

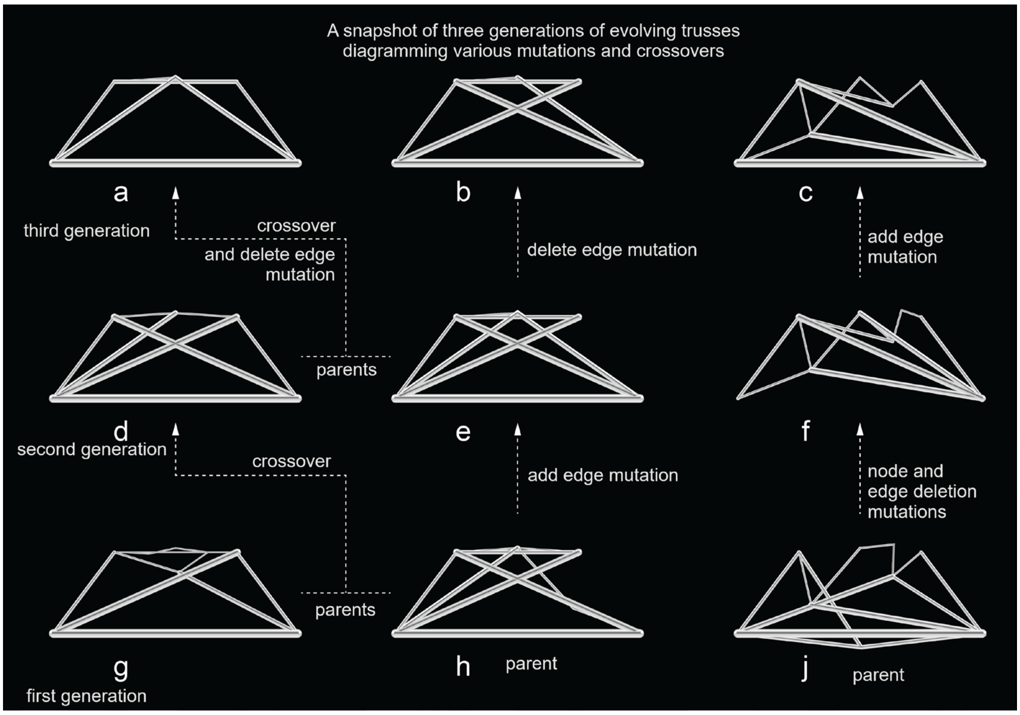

Figure 13: Three species in three generations of evolution showing various mutations.

5. Human Interaction

Typical truss configurations often have a certain aesthetic. The regularity of their geometry may be considered somewhat artificial as it relates to an organic roof form (figure 14). The evolutionary system described in this paper generates solutions which are not constrained by artificial geometries or shape grammars. Also, as the form evolves, it responds concurrently to multiple influences. An evolved pergola (figure 15) illustrates an irregular organic truss structure which is responsive to both structure and light penetration.

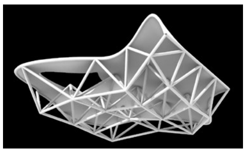

Figure 14: A regular truss supporting an organic roof form

The algorithm, by design, allows for the designer or architect to be involved at any point during the generation phase. Samples can be viewed and selected for enhanced fitness based on aesthetic or other criteria. Once the algorithm converges, a set of suitable solutions emerges from which the designer can select an option that they desire.

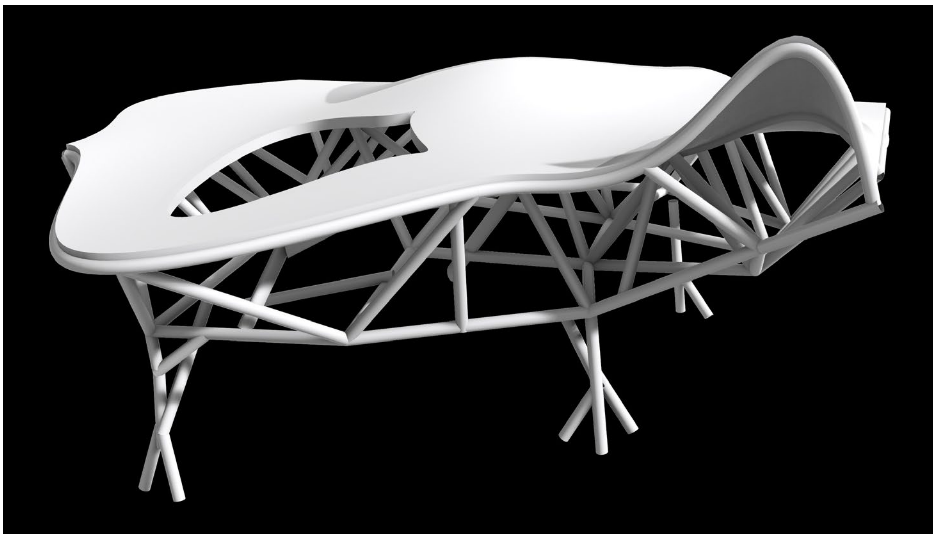

Figure 15: A pergola featuring an irregular evolved truss aesthetic

6. Results and Observations

Preliminary experiments have confirmed that the algorithm is more efficient in terms of computation time than standard backpropagation machine learning systems. This is due to the learning mechanism of modifying the topology of networks. Despite this gain in efficiency, further improvements can be made. For instance, during the evolution process, often one mutation negates a previous one which leads to inefficiency. The emergence of a number of distinct species of truss topologies leads to a series of similarly fit solutions from which the designer can select. Still early in the experimental stage, convergence to close-to-optimal solutions can be improved.

7. Discussion

We are now in a position to delineate four principles that may serve as a vision for future architecture based on the evolutionary model as expounded by NEAT:

a) Internal and External Pressure: Buildings, akin to organisms, are built around the concept of optimal balance, creating an ontological ability to adapt to changing conditions.

b) NEAT versus GA: Architecture is built around the complexity of forms and abundant possibilities, rather than constrained by a reductionist relation between gene and trait.

c) Complexity: The power of combining elements produces architectural forms with structural cohesion and hierarchy.

d) Differentiation and Novelty: Populations of built forms exhibit diversity and creativity, with variation and novelty emerging within the population.

8. Future Work

The NEAT algorithm has many control parameters such as population size, number of species, types of mutations, mutation rates, etc. Tuning is necessary to investigate the limits of the algorithm. Structural performance such as multiple unbalanced loads and deformation need to be considered. A graph of fitness vs. generation is needed to observe convergence rates. Environmental analysis tools such as Ladybug [6] can be integrated. Also, further support for modelling 3-dimensional trusses is necessary.

9. Conclusion

Since Darwin's time, architectural discourse has been influenced by the theory of evolution. The genetic algorithm is a powerful computational tool for generating and optimizing designs. Although it has become a mainstay in the architectural design process, due to its reductionist perspective, it has inherent limitations. We have proposed an enhancement called eNEAT which augments the GA by evolving networks rather than simple scalar parameters. Thus, eNEAT has the capacity to model combinations, thereby simulating more closely the way DNA regulates the construction of natural organisms. Its enhanced complexity allows for the modeling of hierarchical morphologies, responsive to parameters that are internal to the architectural artifact and external to the environment. Moreover, manifold attributes of the built form can be developed concurrently in a unified phase, rather than via the conventional sequential architectural design process. The multidisciplinary framework of this project, using biological circuitry in conjunction with computer science, engineering, and architectural design and planning offers the potential of developing a computational protocol that is capable of negotiating multiple classes of independent information in the designing of the architectural form.

Acknowledgements

The authors wish to thank researchers at the MTRL laboratory, Technion. The patient feedback and support given by Ofer Asaf, Tamar Nix, Anna Boim, and Roni Hillel is much appreciated. We want to thank Erez Hochman for his tutelage on free body diagrams, as well as research assistant Sivan Shafir for his timely illustrations. We wish to also thank Dr. Daniel Shorkend for his literary contribution.

References

[1] Kamaoğlu, M. (2023). The idea of evolution in digital architecture: Toward united ontologies? In *International Journal of Architectural Computing* (Vol. 21, Issue 4).

[2] Holland, J. H. (1975). Adaptation in natural and artificial systems : an introductory analysis with applications to biology, control, and artificial intelligence. In *Ann Arbor University of Michigan Press.*

[3] Kumar, S., & Bentley, P. j. (2003). *On Growth Form and Computers*. MIT Press.

[4] Stanley, K. O., & Miikkulainen, R. (2002). The MIT Press Journals Evolving Neural Networks through Augmenting Topologies. *Evolutionary Computation*, *10*(2).

[5] Thompson, D. W. (1917). *On Growth and Form*. Cambridge [Eng.] University Press, 1917.

[6] Mostapha, S. R. (2013). *Ladybug: A parametric environmental plugin for grasshopper to help designers*.

[7] Picon, A. (2013). *Ornament: The Politics of Architecture and Subjectivity*.

[8] Cogdell, C. 2018. "Toward A Living Architecture?", University of Minnesota

[9] Coyne, J. (2005). Endless Forms Most Beautiful: The New Science of Evo Devo by Sean B Carroll. In *Nature -London-* (Vol. 1, Issue 7045).

[10] Alon, U. (2019). An Introduction to Systems Biology: Design Principles of Biological Circuits, Second Edition. In *An Introduction to Systems Biology: Design Principles of Biological Circuits, Second Edition*.

[11] Dörstelmann, M., Prado, M., Menges, A., Parascho, S., & Knippers, J. (2016). *Integrative Computational Design Methodologies For Modular Architectural Fiber Composite Morphologies Stefana Parascho École Polytechnique Fédérale de Lausanne*

[12] Thompson, D. W. (1917). *On Growth and Form*. Cambridge [Eng.] University Press, 1917.

[13] Frazer, J. (1995). *An Evolutionary Architecture*. Architectural Association.

[14] Sprecher, A., & Kalnitz, P. (2008). Degrees and switches. *ACADIA 08: Silicon + Skin: Biological Processes and Computation: Proceedings of the 28th Annual Conference of the Association for Computer Aided Design in Architecture*.

[15] Green, C. (2023). *SharpNEAT – Evolution of Neural Networks*, https://github.com/colgreen/sharpneat

[16] Preisinger, C., & Heimrath, M. (2014). Karamba - A toolkit for parametric structural design. *Structural Engineering International: Journal of the International Association for Bridge and Structural Engineering (IABSE)*, *24*(2).

[17] Dewey, C. N. (2019). Whole-genome alignment. In *Methods in Molecular Biology* (Vol. 1910).