Generative Design

PARK

Index

01

Defining Generative Design

GENERATIVE

adjective, able to produce or create something.

What is Generative Design?

Among the various definitions of generative design, one of the most common describes it as a design methodology that employs computational processes to explore, evaluate and refine design solutions within a defined rule-based framework.

Rather than directly shaping the final outcome, designers establish a set of rules and constraints—such as geometrical relationships, material properties, or other evaluation criteria, allowing the system to iteratively generate (and assess) multiple design variations.

The process begins with an abstracted idea that is translated into a sequence of algorithmic operations, forming a source code that the computer can interpret (*ref. Generative Gestaltung - Lazzeroni, Bohnacker, Groß and Laub 2009*). This code acts as a set of instructions that guide the generation of different design alternatives. By integrating evaluation criteria directly into the system and utilising a special class of algorithms (i.e. evolutionary algorithms) designs can be tested and optimized against predefined objectives, leading to solutions that might not have been conceived through conventional approaches.

Before examining its role in the AE industry, it is important to acknowledge that the generative approach extend beyond this field. In art, generative software allows artists to create evolving visuals through code. In music, composers like Brian Eno have experimented with generative systems to create compositions that shift and transform over time. The manufacturing industry also applies generative strategies to develop products that are both lightweight and structurally efficient.

This broad application highlights the versatility of generative principles, demonstrating how these computational methods can enhance creativity, optimize processes, and redefine problem-solving across disciplines.

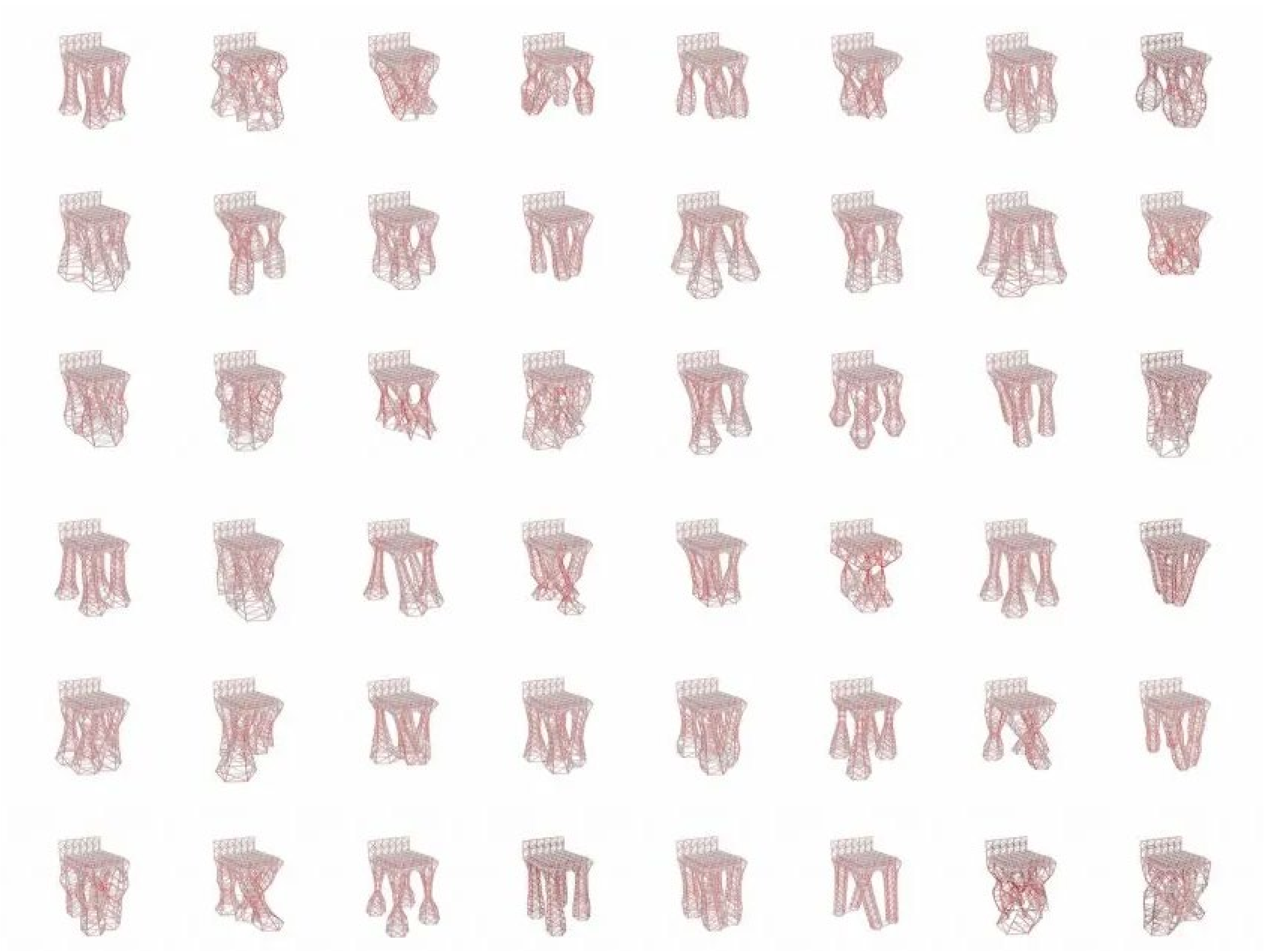


Figure 1. Generative chair design variations matrix

Key features

Across all disciplines that use generative processes, there are some common characteristics to be found. At the core always lies an an algorithm-driven approach, where rule-based models automate the generation and refinement of design variations. Iterative exploration is another shared feature, where solutions evolve through continuous computational cycles, leveraging feedback loops to address complex, often multi-objective, design challenges.

Another recurring theme is the negotiation between the designer and the machine, where human-defined inputs meet algorithmic optimization to generate design variations. In this context, input constraints play a crucial role with each discipline - whether architecture, engineering, or art - adhering to specific parameters that respond to formal, material, structural, or environmental criteria.

Finally, these processes embrace output diversity, generating multiple solutions rather than a single predefined result, fostering a broader exploration of alternative outcomes.

These characteristics are further detailed in the paragraphs below.

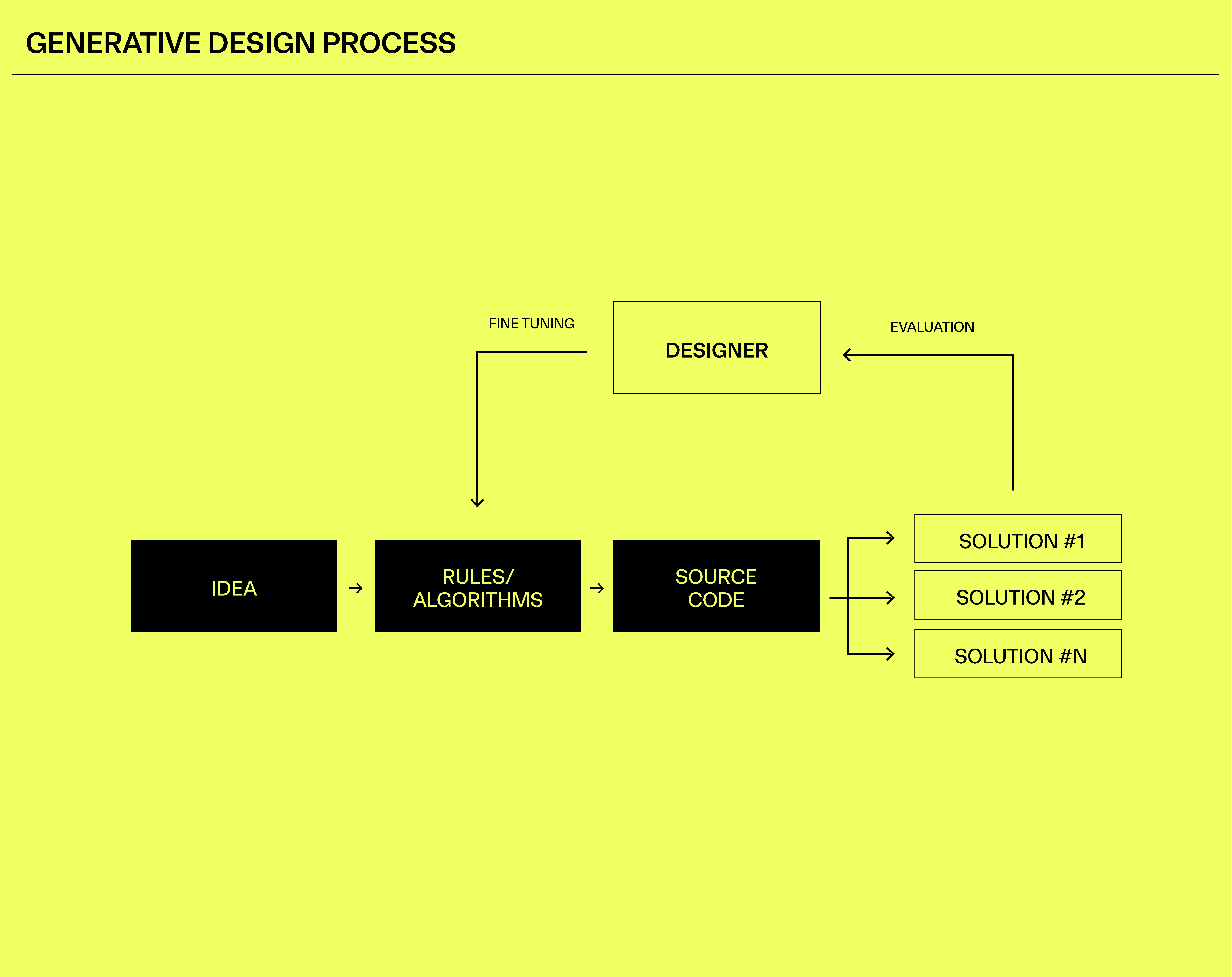


Figure 2. Generative design process flowchart

Algorithmic-driven process

Generative design is fundamentally an algorithm-driven process, relying on computational models and parametric logic to generate and optimize design solutions. Unlike traditional design workflows that rely on manual iteration, generative design automates the exploration of vast solution spaces through advanced algorithms. This enables designers to analyze thousands of potential configurations and identify the most effective solutions based on predefined performance criteria.

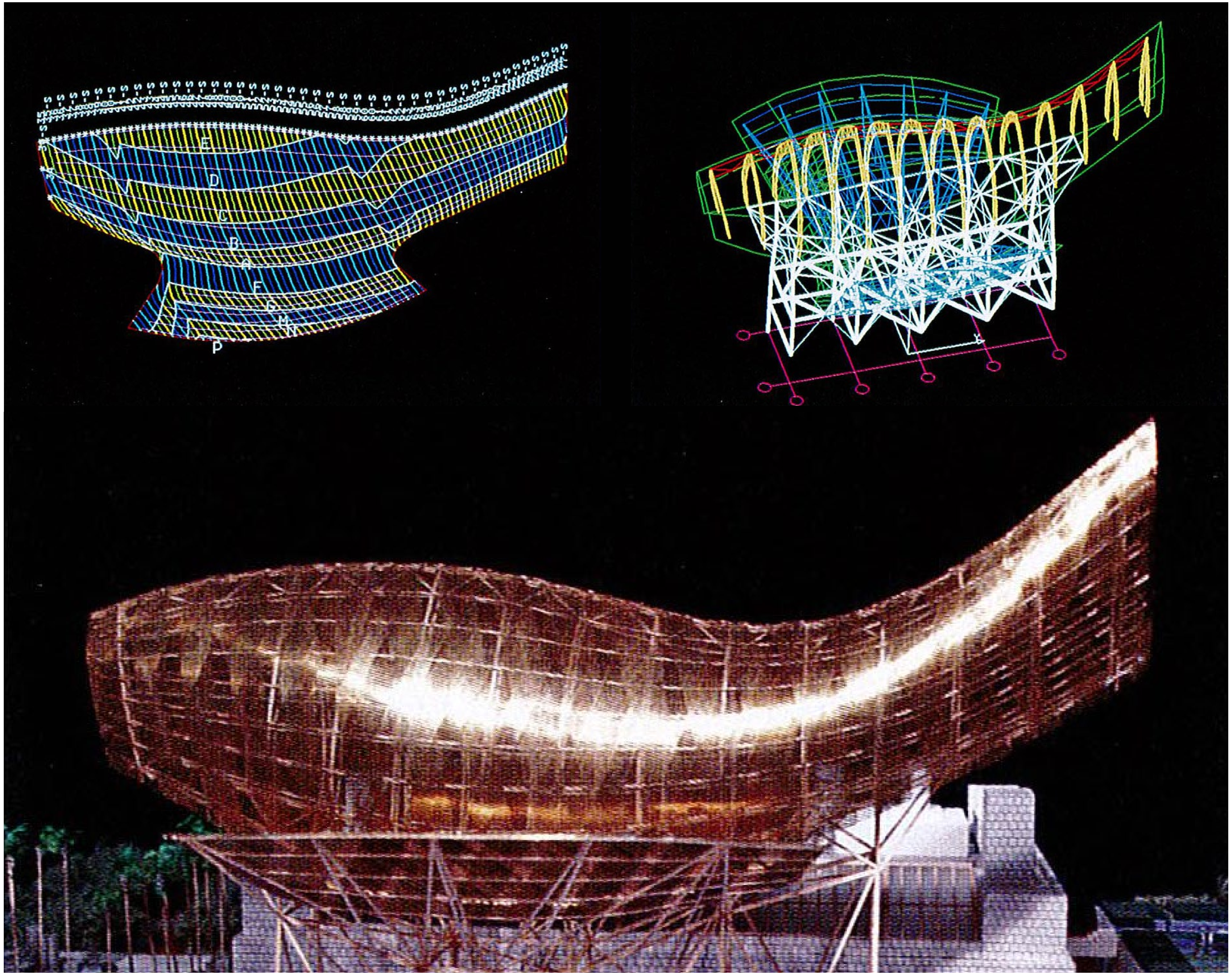


Figure 3. Frank Gehry's Barcelona fish sculpture advanced 3D modeling anticipated elements of today's algorithmic design

Iterative exploration

Generative design is inherently iterative, with solutions evolving through continuous refinement cycles. Using optimization techniques, the system progressively enhances each iteration, discarding suboptimal configurations and converging toward solutions that best balance competing objectives. This feedback-loop approach is particularly valuable in complex applications across architecture and engineering, where multiple constraints and optimization goals must be addressed simultaneously.

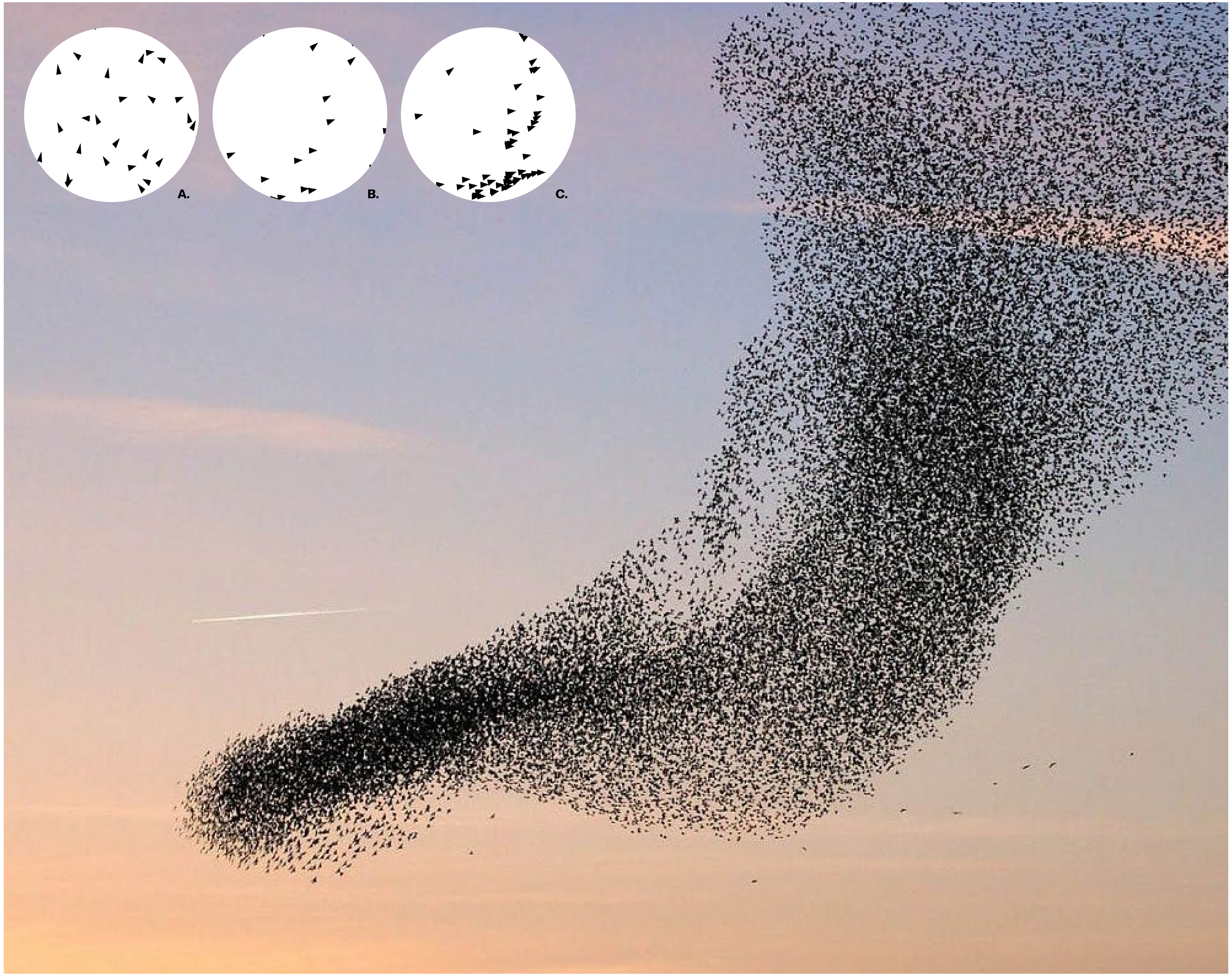


Figure 4. (above) Craig Reynolds' Boids simulates flocks using simple interaction rules

Figure 5. (right) A flock of birds in coordinated flight

Input constraints

Input constraints play a crucial role in shaping the generative design process, as designers define key parameters such as spatial, structural, and manufacturing requirements, among others. These constraints provide a structured framework within which the algorithm explores viable solutions, ensuring that all generated outcomes align with the specified criteria.

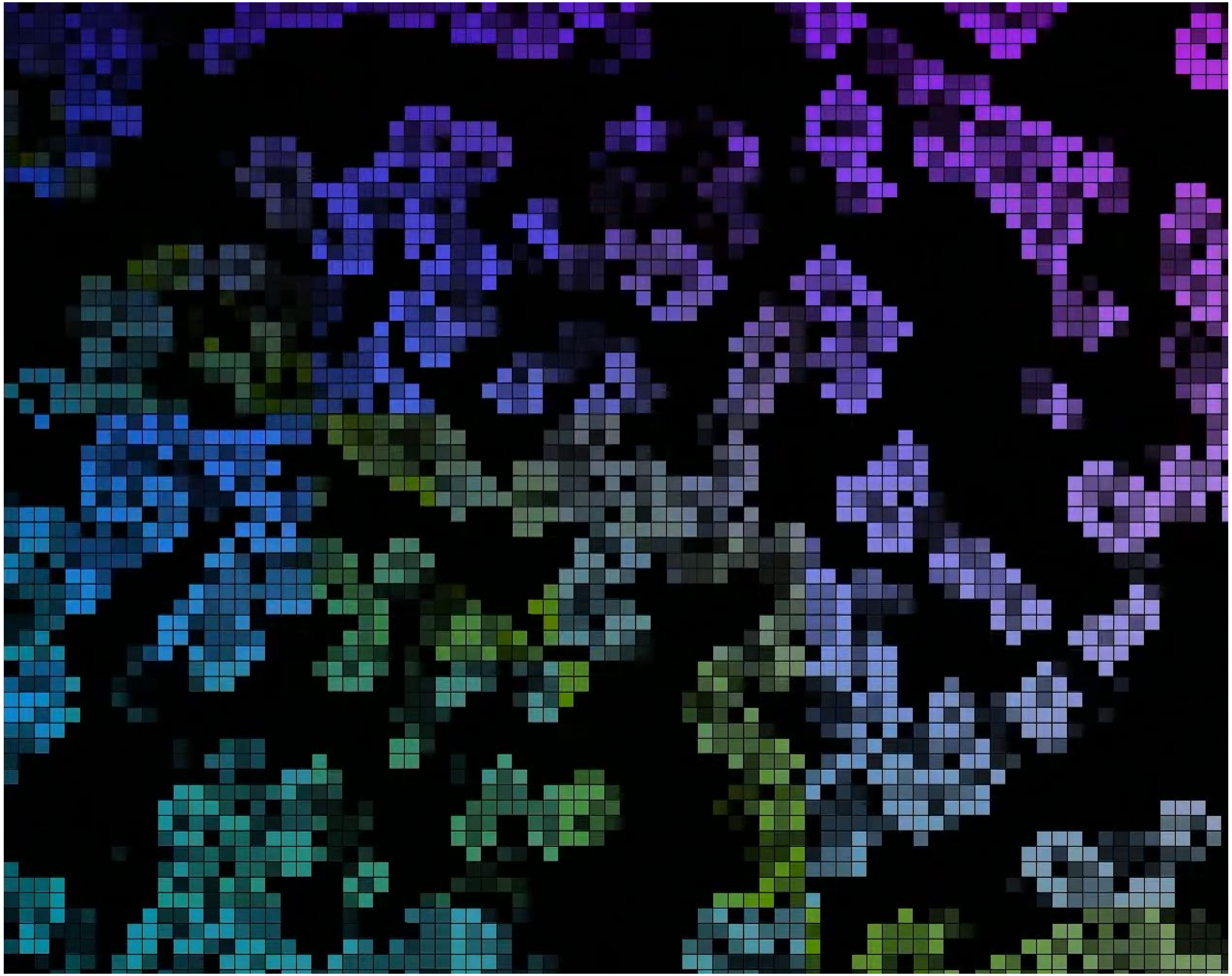


Figure 6. Conway's Game of Life evolution is shaped by its initial setup

Designer-Machine negotiation

Generative design operates as an ongoing negotiation between the designer and the machine, where human-defined rules and inputs interact with algorithmic logic to shape the design process. Rather than simply setting fixed constraints, the designer actively engages with the system, interpreting, adjusting, and responding to generated variations. This exchange allows for a dynamic interplay between human intent and computational exploration, requiring iterative refinement to guide the algorithm toward meaningful and contextually relevant outcomes.

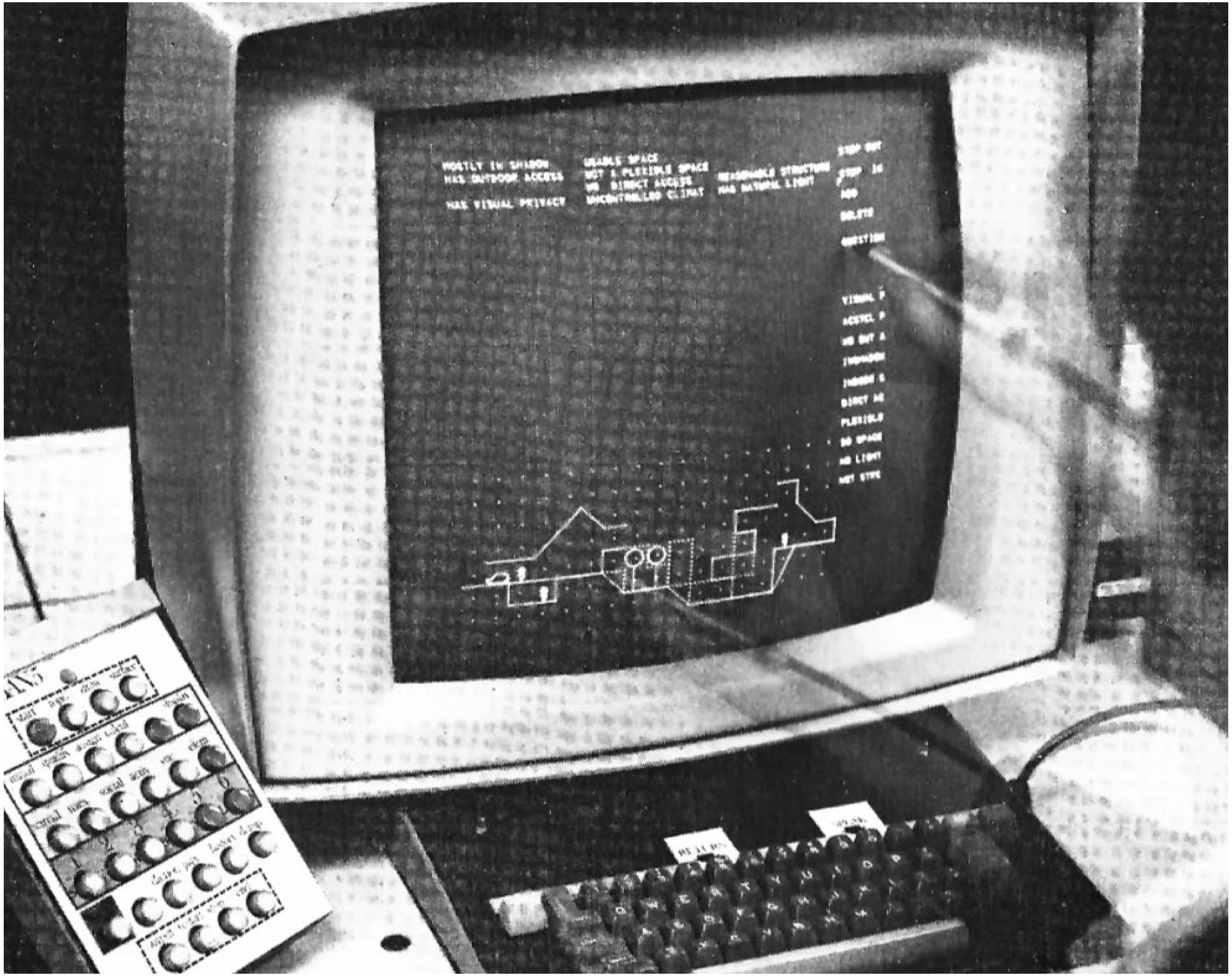


Figure 7. URBAN5 by Nicholas Negroponte is a pioneering system that reflects users' own design choices

Output diversity

A key feature of generative design is its ability to generate a wide array of optimized solutions instead of a single fixed outcome. These variations represent different trade-offs across the defined parameters and performance criteria, providing designers with a broad selection of potential solutions that may not arise through conventional design approaches.

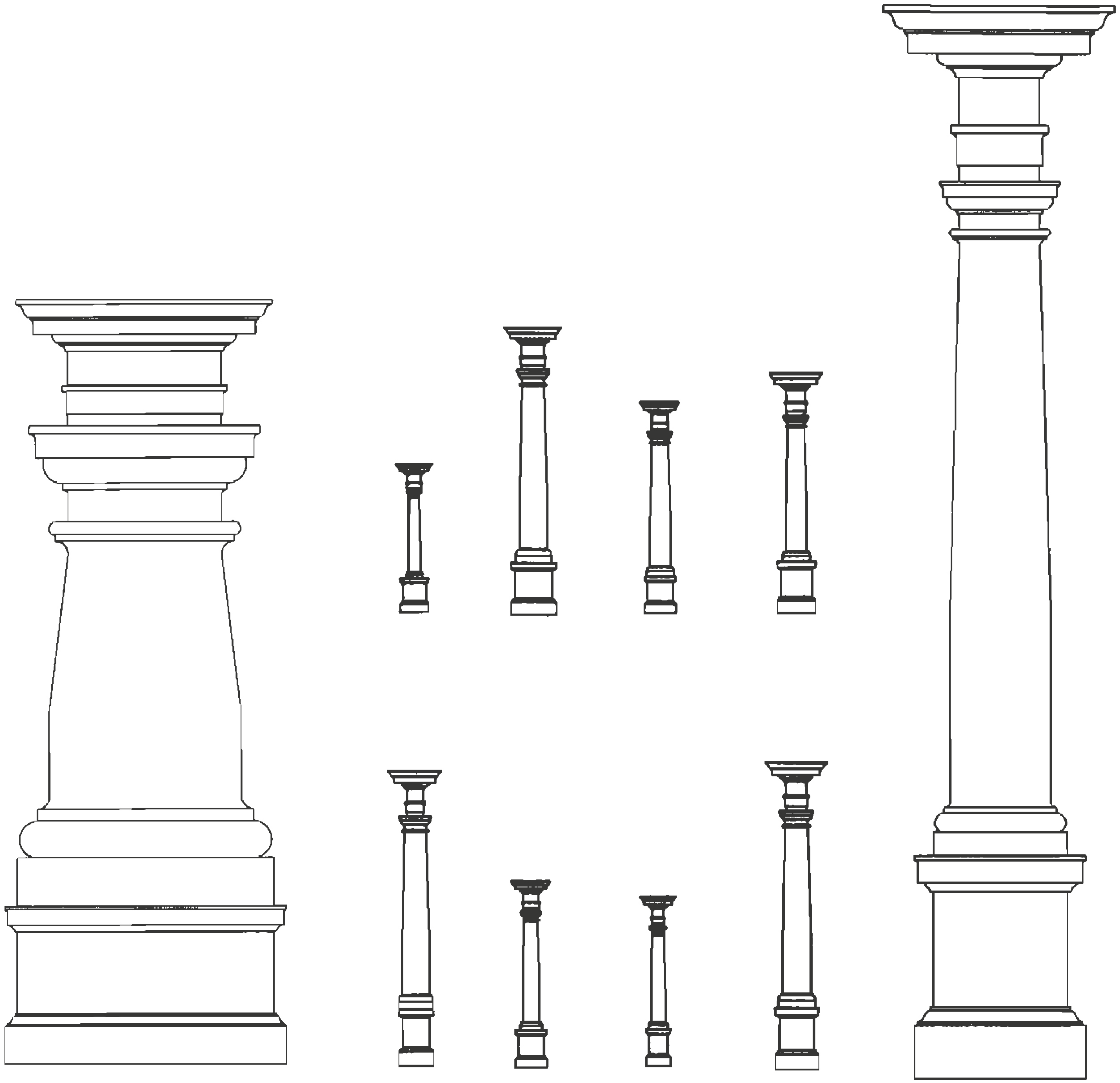


Figure 8. Evolution of Tuscan columns by genetic algorithms by John Frazer with Peter Graham

02

Pioneers

Pre-Digital Generative Design

Before the advent of digital tools, the essence of generative design was already alive in the methods of some architects, engineers, and designers who engaged in form-finding through physical models. This empirical process involved manipulating material to explore structural and formal possibilities, often guided by intuition and hands-on experimentation. The physical interaction with materials allowed designers to intuitively understand and optimize forms based on how they behaved under different forces - gravity, tension, or compression.

Material and structural optimization was often achieved through a blend of craftsmanship and scientific insight. Think of Gaudí's hanging chain models for the Sagrada Família or Frei Otto's soap bubble experiments to discover minimal surfaces. These analog techniques were early explorations of optimization principles - seeking the most efficient structure with the least material. They inherently balanced form and function, much like today's generative algorithms that weigh multiple criteria.
This empirical and iterative approach was central to these explorations. Designers would build, test, observe, and refine, learning through each cycle of creation. This hands-on iterations mimicked the evolutionary processes now echoed in generative design algorithms, where solutions are tested, mutated, and selected for fitness.
These pre-digital practices laid the conceptual groundwork for what would later be encoded into software. The shift to digital generative design did not erase these traditions but rather encoded and expanded them - transforming tactile, iterative exploration into a computationally powered expansion of possibilities. The same principles of experimentation, optimization, and form-finding endure, now accelerated and amplified by the power of computation.

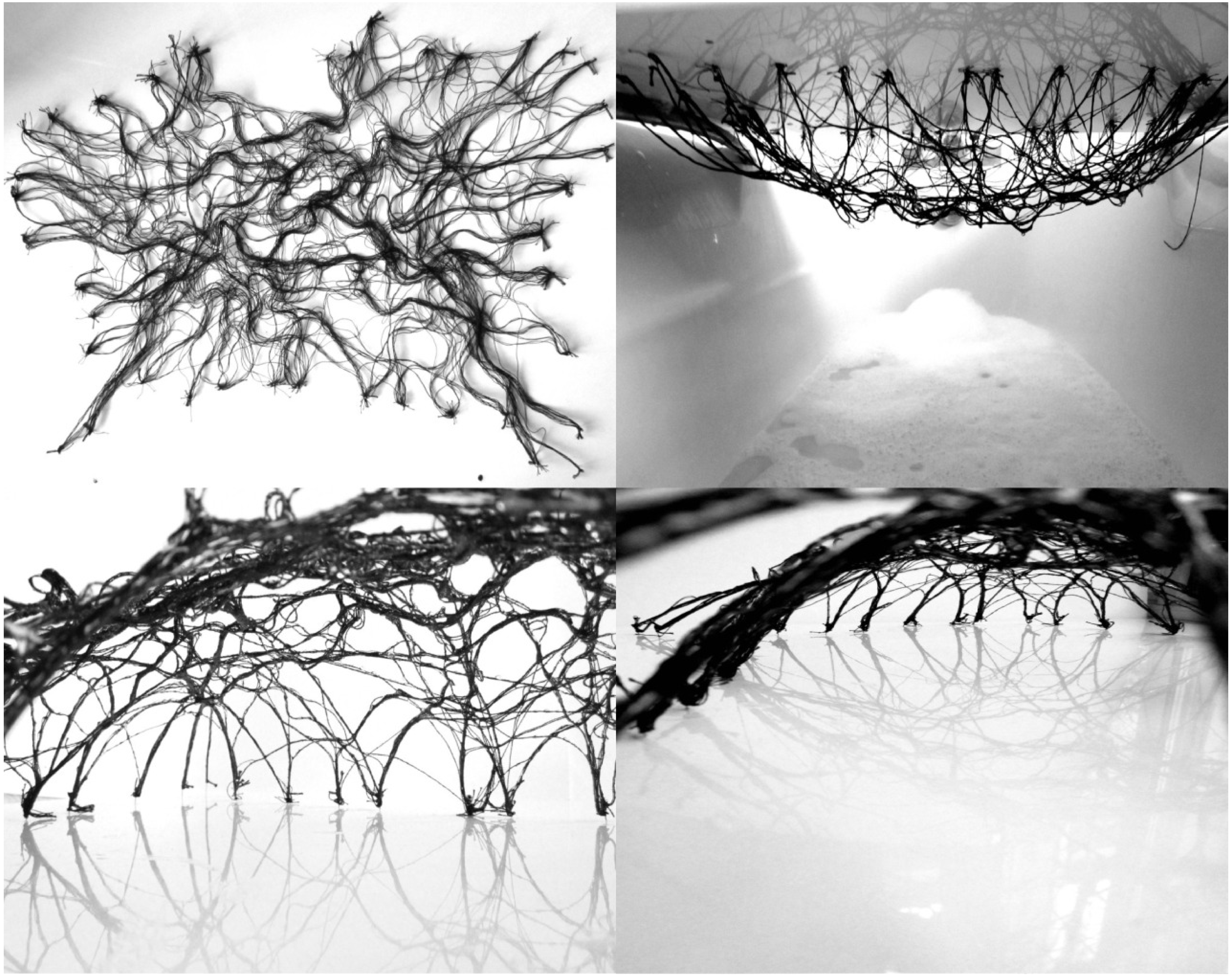


Figure 9. Experiment by Carolyn Olivia Butler inspired by Frei Otto's minimal path research

Frei Otto

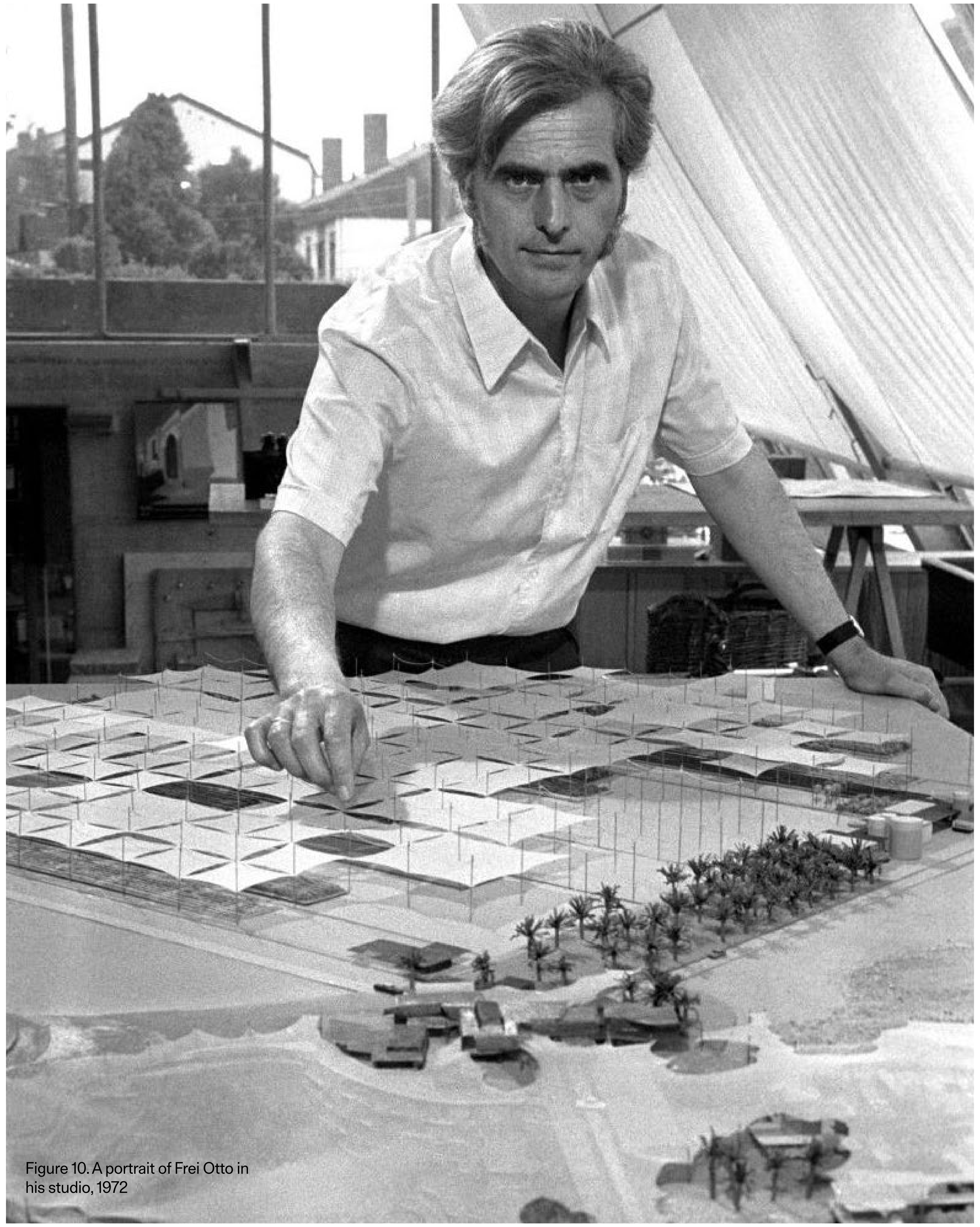


Figure 10. A portrait of Frei Otto in his studio, 1972

Figure 11. Tanzbrunner Pavillion by Frei Otto

Frei Otto

Frei Paul Otto (1925-2015) was a German architect and structural engineer noted for his use of lightweight structures, in particular tensile and membrane structures, including the roof of the Olympic Stadium in Munich for the 1972 Summer Olympics.

In Frei Otto perspective, architecture was a process of discovery rather than predetermined design, deeply rooted in natural principles, structural efficiency, and material economy. He saw design as an evolutionary process, where forms emerged through empirical experimentation and adaptation rather than preconceived top-down outcomes. His work focused on lightweight, adaptable structures, drawing inspiration from nature's efficiency - such as spiderwebs, soap films, and biological growth patterns. His approach was deeply interdisciplinary, merging engineering, physics, and material science to create architectures that were structurally innovative.

Frei Otto can be considered a pioneer of generative design due to his exploratory, empirical, and optimization-driven approach to architectural form-finding.

His work was grounded in physical experimentation, using soap films and tensile membranes to naturally determine structures that were both materially efficient and structurally performative. These analog models functioned as early generative systems, where natural forces such as tension, gravity, and surface tension "computed" the most efficient forms - prefiguring today's algorithm-driven optimization techniques. By embracing an iterative, rule-based process, Otto's research laid the foundation for computational generative design, demonstrating how natural principles and mathematical logic could inform architecture in a way that balances structural performance, resource efficiency, and aesthetic elegance. His influence persists in contemporary digital workflows that use parametric modeling and evolutionary algorithms to achieve similar goals through computational means.

Antoni Gaudì

Figure 12. Reproduction of a hanging model of the church of Colònia Guell

Figure 13. Interiors of Sacrada Familia by Antoni Gaudì, from 1883

Antoni Gaudì

Antoni Gaudí (1852-1926) was a Catalan architect known for his distinctive approach to modernist architecture, blending organic forms, intricate ornamentation, and structural innovation, with landmark works such as the Sagrada Família in Barcelona.

Antoni Gaudí envisioned architecture as a synthesis of nature, structure and spirituality, where form followed the inherent logic of natural systems. He saw buildings as organic entities, shaped by the same geometric and structural principles that govern the natural world. His designs were deeply influenced by biomimicry, using catenary arches, hyperboloids, and ruled surfaces to create both ornamental and structural elements. Gaudí also viewed architecture as an evolving, experimental process, relying on physical models and hands-on craftsmanship rather than rigid, pre-defined blueprints. His work was not just about functionality but about creating spaces that embodied a deeper spiritual and symbolic meaning, as seen in the intricate, nature-inspired forms of the Sagrada Família.

Antoni Gaudí's innovative form-finding methods and empirical approaches to structural optimization position him as a pioneer of generative design. Rather than designing purely from drawings, he used physical models and natural principles to develop highly efficient and organic architectural forms.

His most notable form-finding experimentation that embeds generative principles was the hanging chain model, where he suspended weighted chains to simulate catenary arches and structural loads, allowing gravity to determine the most efficient form. By inverting these models, he achieved structurally optimized designs for projects like the Sagrada Família and Colònia Güell Chapel.

This iterative, evolutionary-like design method prefigured modern computational generative design, where algorithms now perform similar gravity-based optimizations. Gaudí's work exemplifies an early analog version of generative thinking - using natural forces, experimentation, and rule-based form-finding to create structurally sound and highly expressive architectures.

Luigi Moretti

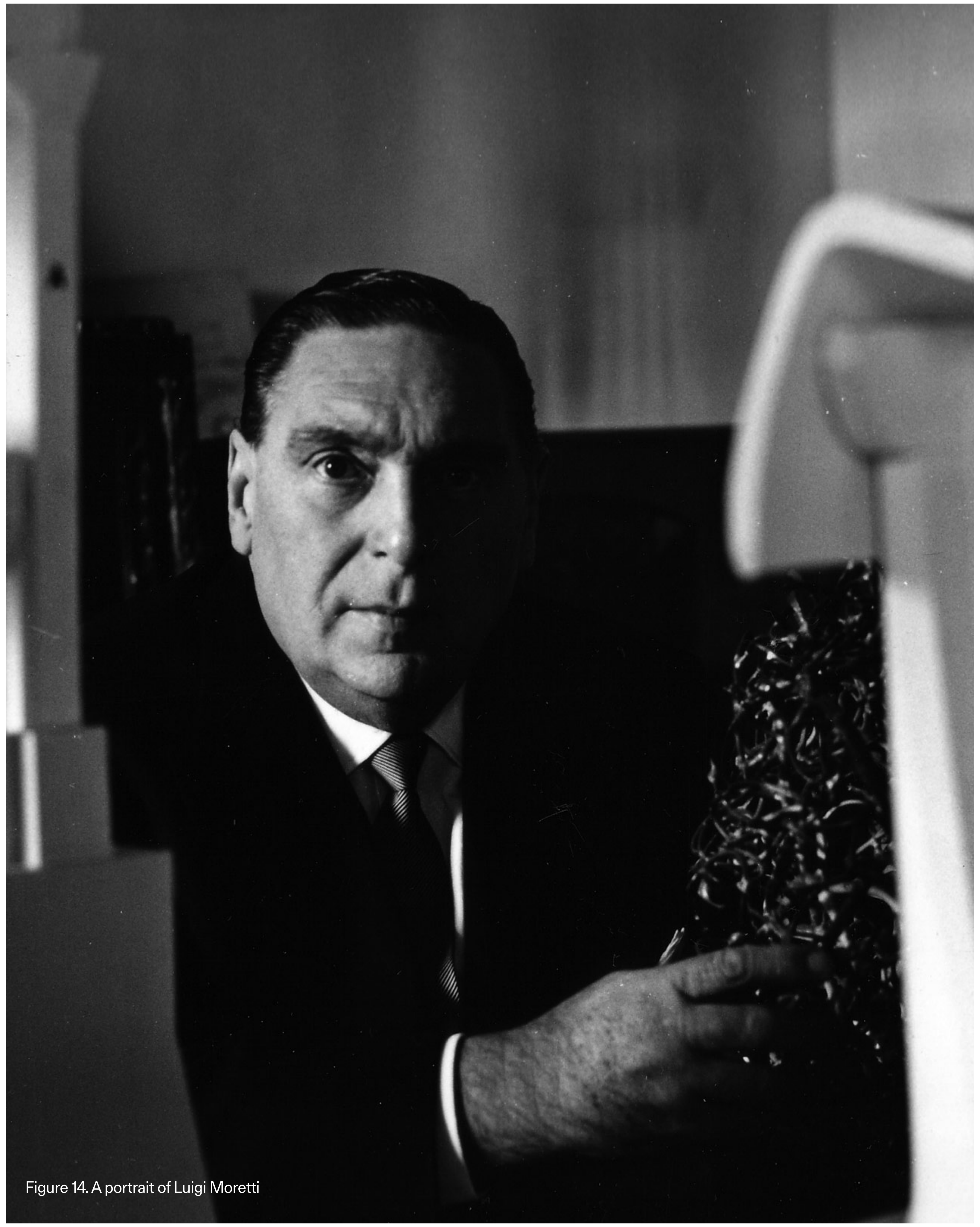


Figure 14. A portrait of Luigi Moretti

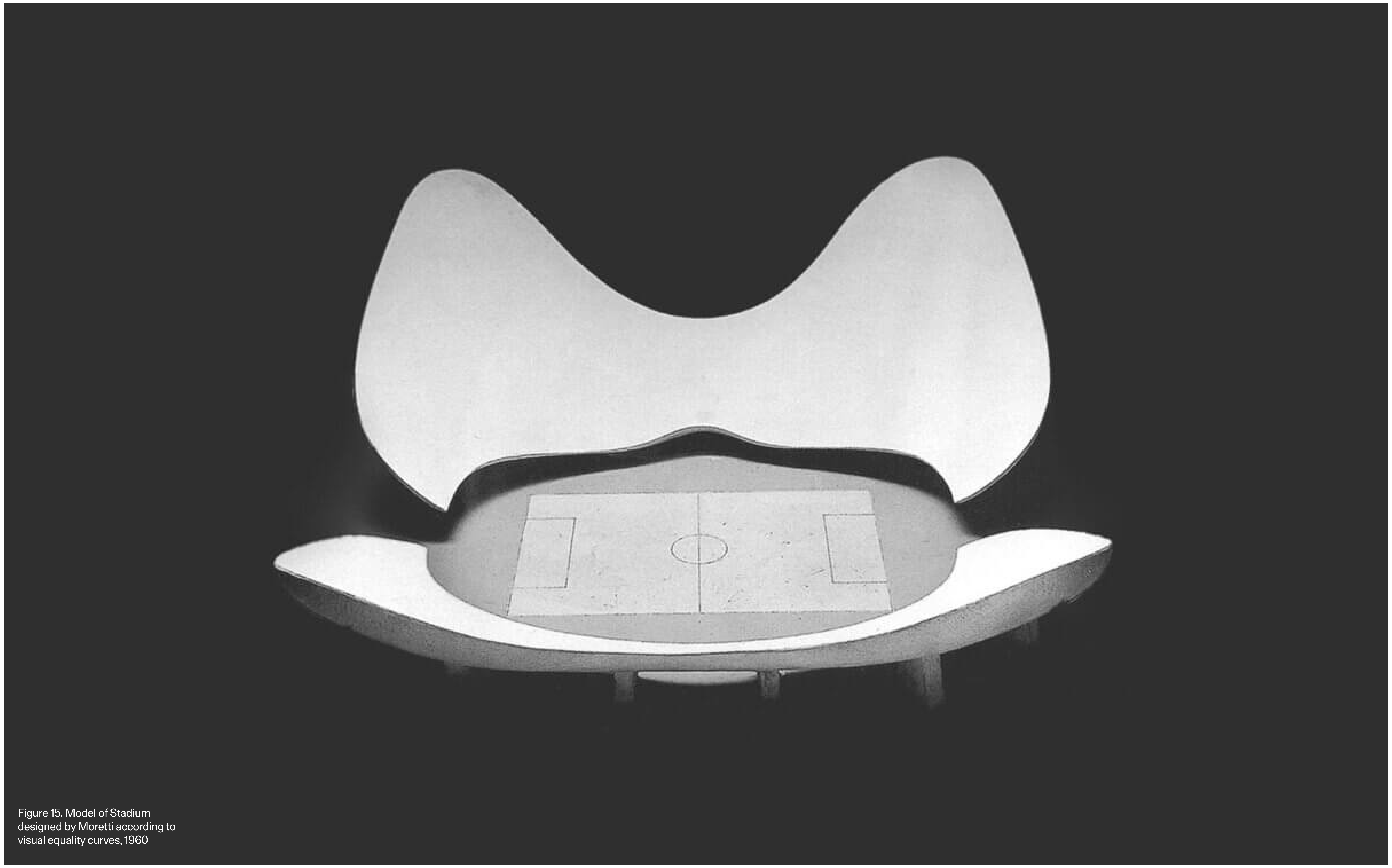


Figure 15. Model of Stadium designed by Moretti according to visual equality curves, 1960

Luigi Moretti

Luigi Walter Moretti (1906-1973) was an Italian architect. Active especially in Italy from the 1930s, he designed buildings such as the Watergate Complex in Washington DC, The Academy of Fencing and Il Girasole house, both in Rome. He was the founder of the Institute for Operations Research and Applied Mathematics Urbanism, where he developed his research on the history of architecture, and on the application of algorithmic methods to architectural design.

Although not directly linked to generative design, Luigi Moretti's pioneering work in parametric architecture played a crucial role in laying the foundations for computational design thinking, influencing the development of methodologies that explore form through data-driven parameters.

In 1971, Luigi Moretti articulated his vision of parametric architecture in Moebius, emphasizing the need to move beyond empirical design through the integration of mathematics, computational techniques, and operational research. He outlined eight principles that defined this approach, focusing on objective analysis, the quantification of design parameters, and the precise relationship between form, function, and context—foundational ideas that anticipated contemporary computational design methodologies.

Moretti's vision of parametric architecture - as both an art and a science - can be fully understood through the analysis of his projects. Stadium N embodies this approach, with its form derived from 19 parameters related to visual perception, structural efficiency, and cost. Defined through typology, parameters, and analytical descriptions, its geometry is optimized for ideal sightlines, showcasing how data-driven processes shape architectural outcomes.

“You never change things by fighting the existing reality. To change something, build a new model that makes the existing model obsolete.”

Richard Buckminster Fuller

03

Roots of Generative Design

Figure 16. Evolution chart of design by Raymond Loewy, in *Industrial Design*, 1930

The roots of generative design can be traced back to the convergence of architectural experimentation with the advent of digital technologies. While the concept of using rule-based systems to generate form has historical precedents - visible in parametric explorations like those of Luigi Moretti - the introduction of computational tools profoundly transformed the scope and potential of these methods. Digital technologies enabled architects to shift from static representations of form to dynamic processes capable of generating endless design variations through algorithmic logic.

On one hand, the emergence of computation, closely tied to the pioneering work of Alan Turing, laid the foundation for algorithmic thinking. Turing's contributions to computer science introduced the idea of machines capable of processing complex operations through logical sequences, an idea that would later become central to generative design.

On the other hand, generative design has been profoundly influenced by Darwin's theory of evolution, which introduced the concept of adaptation through variation, mutation and selection. These principles have been translated into computational methodologies, giving rise to algorithms that mimic natural processes to generate and optimize design solutions. At the intersection of these ideas, Celestino Soddu emerged as a pioneer in the late 1980s. Through developing "*Argenia*" , one of the earliest generative design software programs, he demonstrated how algorithmic processes could simulate natural morphogenesis, generating endless design variations while maintaining coherence with an initial design intent.

Figure 17. Venice reimagined by Argenia, the first generative design software, by Celestino Soddu

Alan Turing and Modern Computing

Alan Turing (1912–1954), a British mathematician, logician, and cryptographer, is widely regarded as one of the founding figures of modern computing.

In his groundbreaking 1936 paper, *On Computable Numbers*, he introduced the concept of the Turing Machine - a theoretical model that formalized the principles of algorithms and mechanical computation. This abstract device demonstrated how any problem solvable by an algorithm could be executed by a machine, laying the theoretical groundwork for digital computers. Turing's ideas were later realized in hardware through the development of stored-program architectures, most notably by John von Neumann, which enabled computers to process both data and instructions flexibly.

Beyond his pivotal role in codebreaking during World War II and his contributions to artificial intelligence - most notably through the proposal of the Turing Test - Turing's work fundamentally shaped how we conceptualize computation. His theories not only enabled the development of modern computers but also laid the foundations for algorithmic thinking, which is central to generative methodologies.

The notion of rule-based systems, iterative processes, and computational logic in generative design can be directly traced back to Turing's exploration of how machines can process and generate complex outputs through predefined instructions. Without his contributions, the evolution of generative design, with its reliance on algorithmic frameworks and automated systems, would not have been possible.

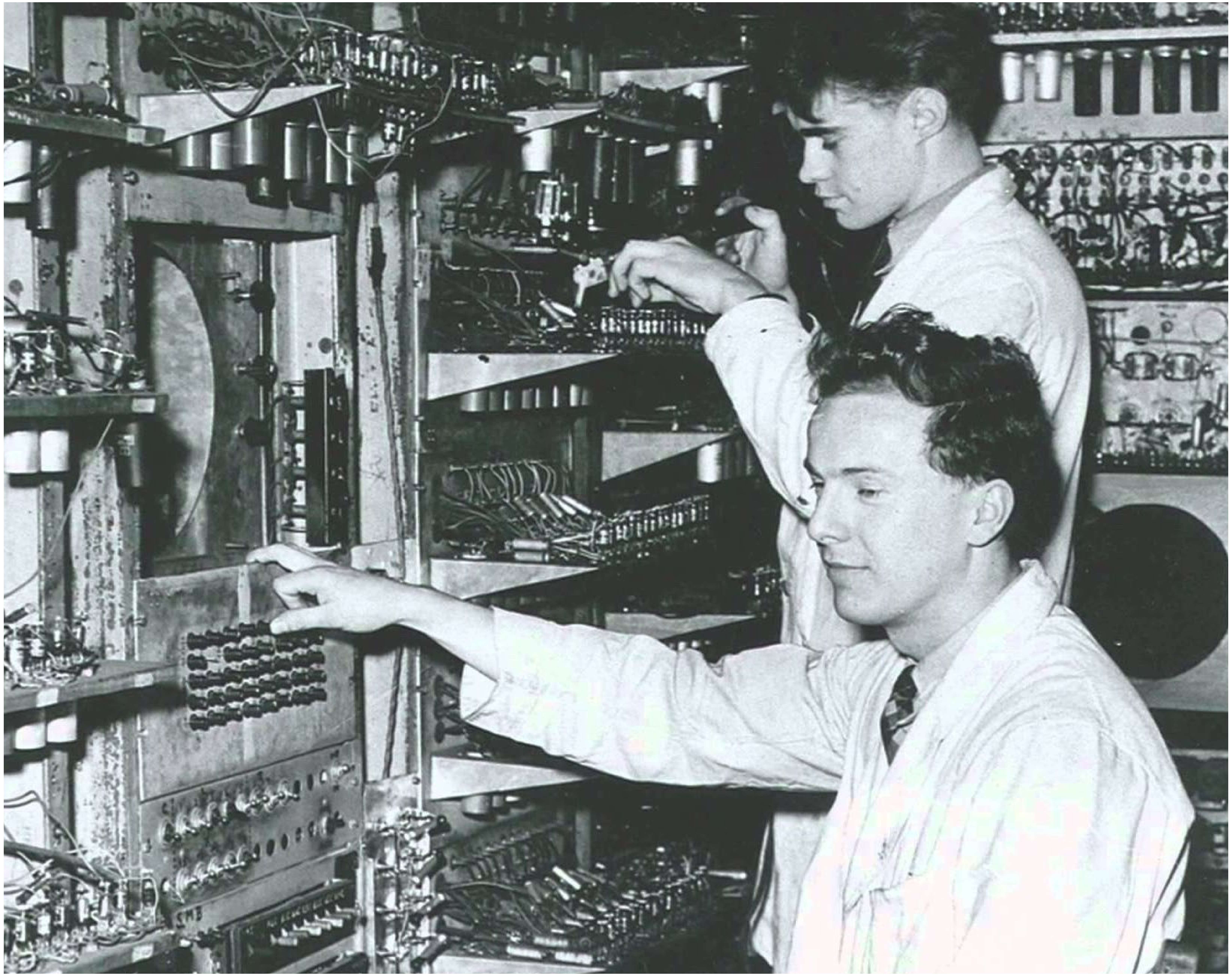


Figure 18. Alan Turing and the code-breaking machine used during WWII at Bletchey Park

The Theory of Evolution

Darwin's theory of evolution, introduced in *On the Origin of Species* (1859), profoundly shaped not only biological sciences but also influenced various disciplines, including the development of generative design principles.

At its core, the theory suggests that within any species, individuals exhibit variations - differences that, while often subtle, can provide distinct advantages in survival and reproduction. Over time, traits that enhance an organism's ability to thrive in its environment are passed on to future generations, while less advantageous characteristics gradually disappear. This continuous process of variation, selection, and adaptation leads to the dynamic evolution of species, shaping the incredible diversity of life observed in nature.

What makes Darwin's theory revolutionary is its emphasis on change as a constant force, driven not by predetermined outcomes but by interactions between organisms and their environments. It highlights the importance of systems that are flexible, adaptive, and capable of responding to shifting conditions through incremental transformations. This idea of complexity emerging from simple rules and iterative processes has resonated far beyond biology, influencing fields such as systems theory, cybernetics, and ultimately, design methodologies.

In generative design, Darwin's principles provide a framework for how solutions evolve. Focusing on variation, adaptation, and refinement, designers use rule-based systems to explore possibilities, shaping designs through iterative processes that respond to specific constraints—much like species adapting to their environments.

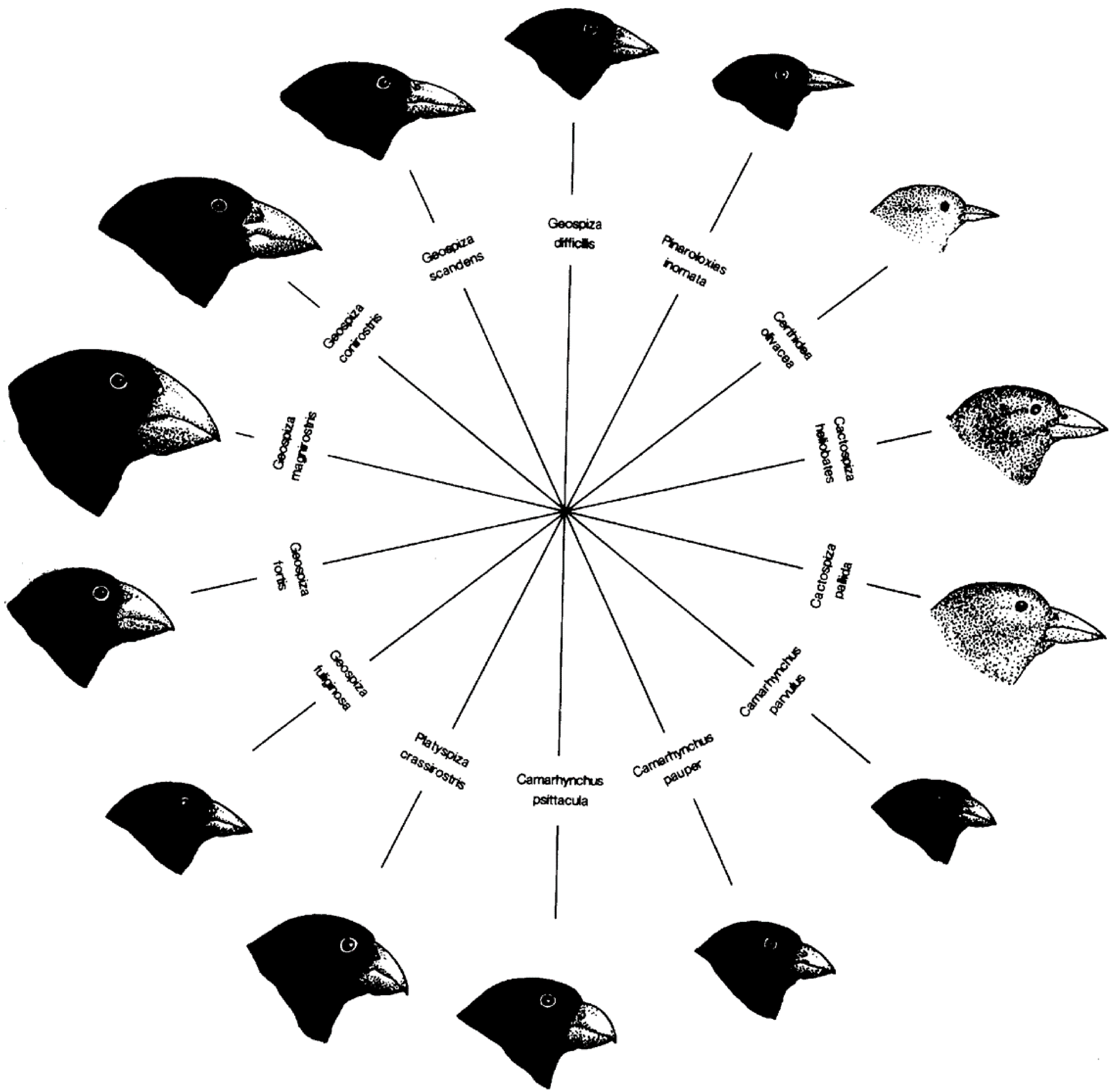


Figure 19. Darwin's finches and the theory of evolution of the species

Evolutionary Computation

Evolutionary computation emerged at the intersection of computer science and evolutionary biology, combining the algorithmic foundations laid by Alan Turing with the adaptive principles of Darwin's theory of natural selection.

While Turing's theoretical work in the 1950s hinted at the potential for machines to learn through evolutionary principles, the formal development of evolutionary computation began with researchers like John Holland, who introduced genetic algorithms in his 1975 book *Adaptation in Natural and Artificial Systems*. Holland's work provided a framework for simulating natural evolutionary processes within computational environments, allowing algorithms to evolve solutions to complex problems. Around the same time, Lawrence Fogel pioneered evolutionary programming, focusing on evolving finite-state machines to predict system behavior, particularly useful in areas like control systems and pattern recognition. Ingo Rechenberg and Hans-Paul Schwefel further advanced the field through the development of evolution strategies, applying them to engineering problems such as aerodynamic design optimization.

These contributions collectively shaped evolutionary computation into a vast research field, enabling optimization techniques used in various industries today, including engineering, finance, and architecture, to optimize designs, automate decision-making, and solve problems that are difficult for traditional algorithms.

In generative design, evolutionary models adds an extra layer that helps optimize design solutions by mimicking natural evolution. Instead of following fixed rules, designs evolve over time through variation and selection. Multiple options are generated and tested against specific goals allowing continuous improvement through feedback. Rather than aiming for a single perfect solution from the start, evolutionary computation helps uncover unexpected, innovative outcomes by allowing designs to adapt and evolve in response to specific challenges and goals.

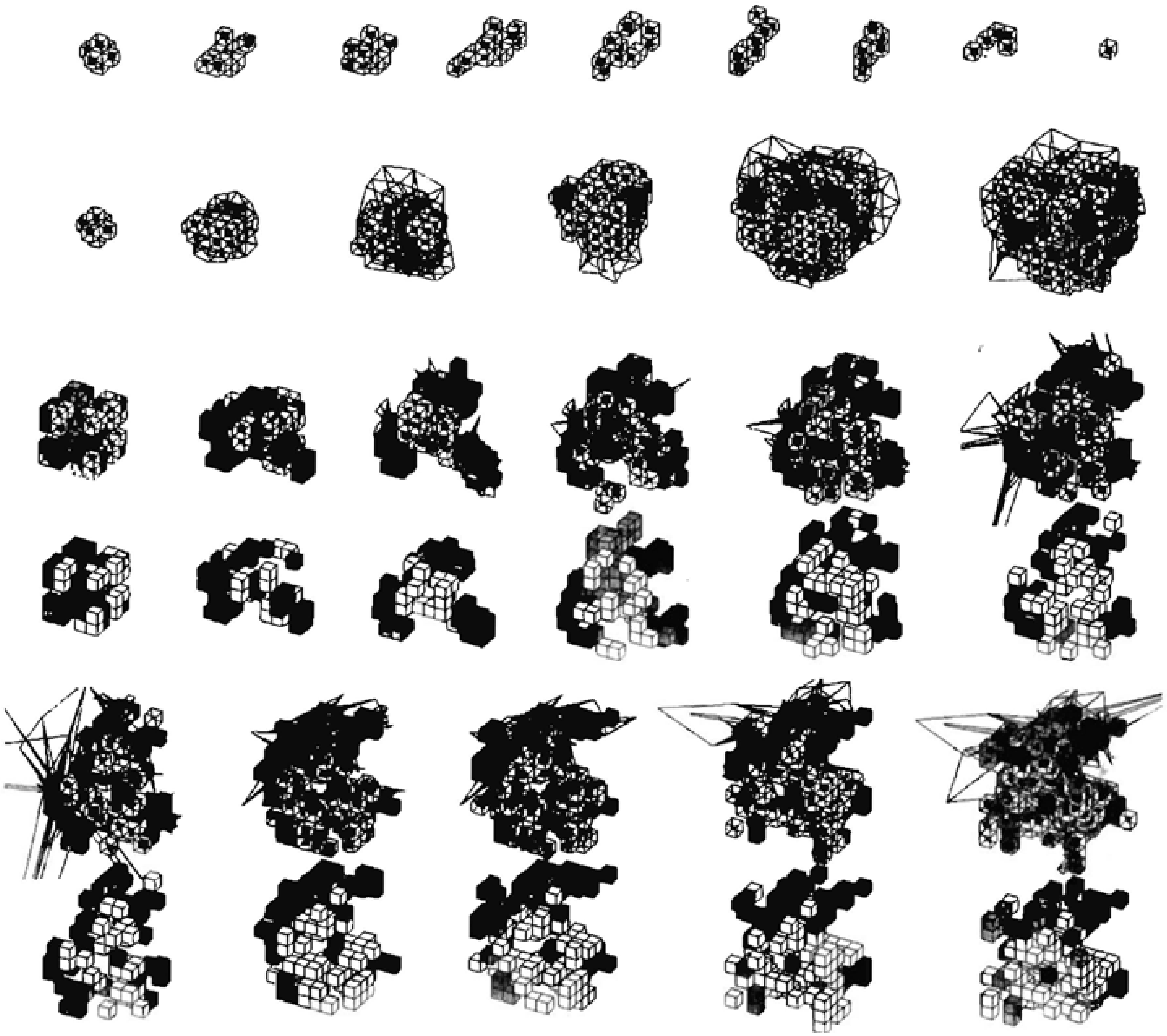


Figure 20. Geometrical aggregations shaped by parameters variation

04

Generative Design Toolkit

EVOLUTION OF DIGITAL DESIGN TECHNOLOGY

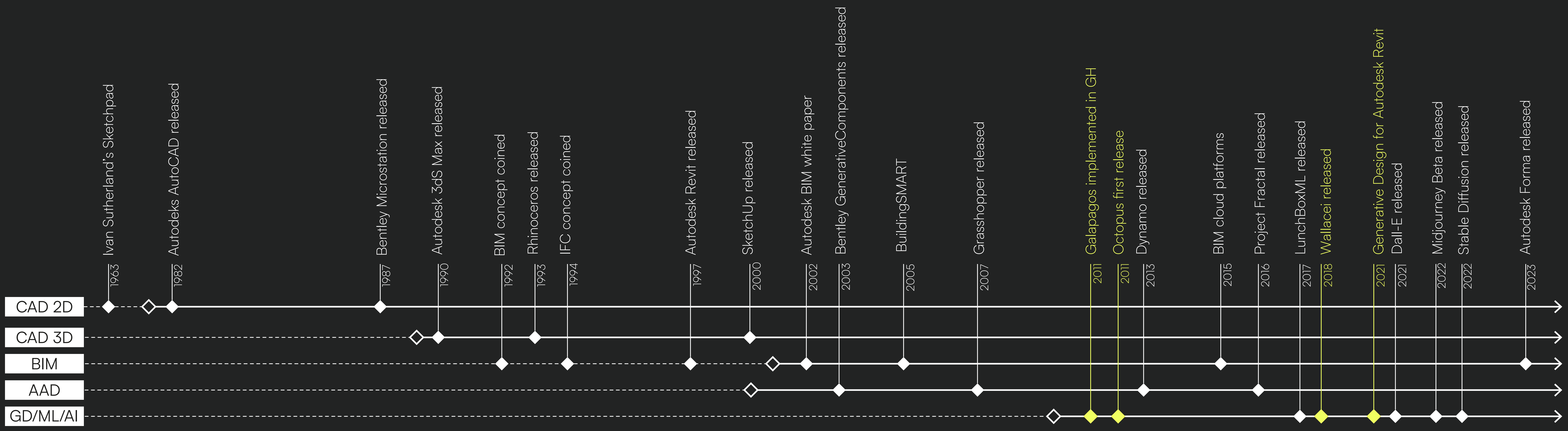


Figure 21. Evolution of digital design technology over time

Brain, Hand, Machine

The relationship between the designer's brain, hand and the machine has evolved alongside technological advancements, driven by the tools that mediate the design process.

Until the 1980s, design was an entirely manual process - drawing by hand directly shaped the final outcome, with no digital mediation.

The introduction of Computer-Aided Design (CAD) in the 1980s marked a shift, building on the pioneering work of Ivan Sutherland and his development of Sketchpad in the 1960s. CAD tools began to mediate the translation of ideas into their final representations. While still largely dependent on manual input, these tools empowered designers to craft and store large drawings in a digital environment.

By the early 2000s, the spread of Algorithm-Aided Design (AAD) transformed this relationship further, moving from direct geometric manipulation to rule-based procedural processes. Designers could now define sets of parameters and constraints, allowing parametric models to generate design variations.

In recent years, generative design has gained momentum in the AEC industry, fueled by the integration of algorithmic tools into most used modelling software.

The increased accessibility of these tools has enabled designers to explore iterative workflows and incorporate evolutionary algorithms such as genetic algorithms (GA) into their design process to develop performance-driven solutions.

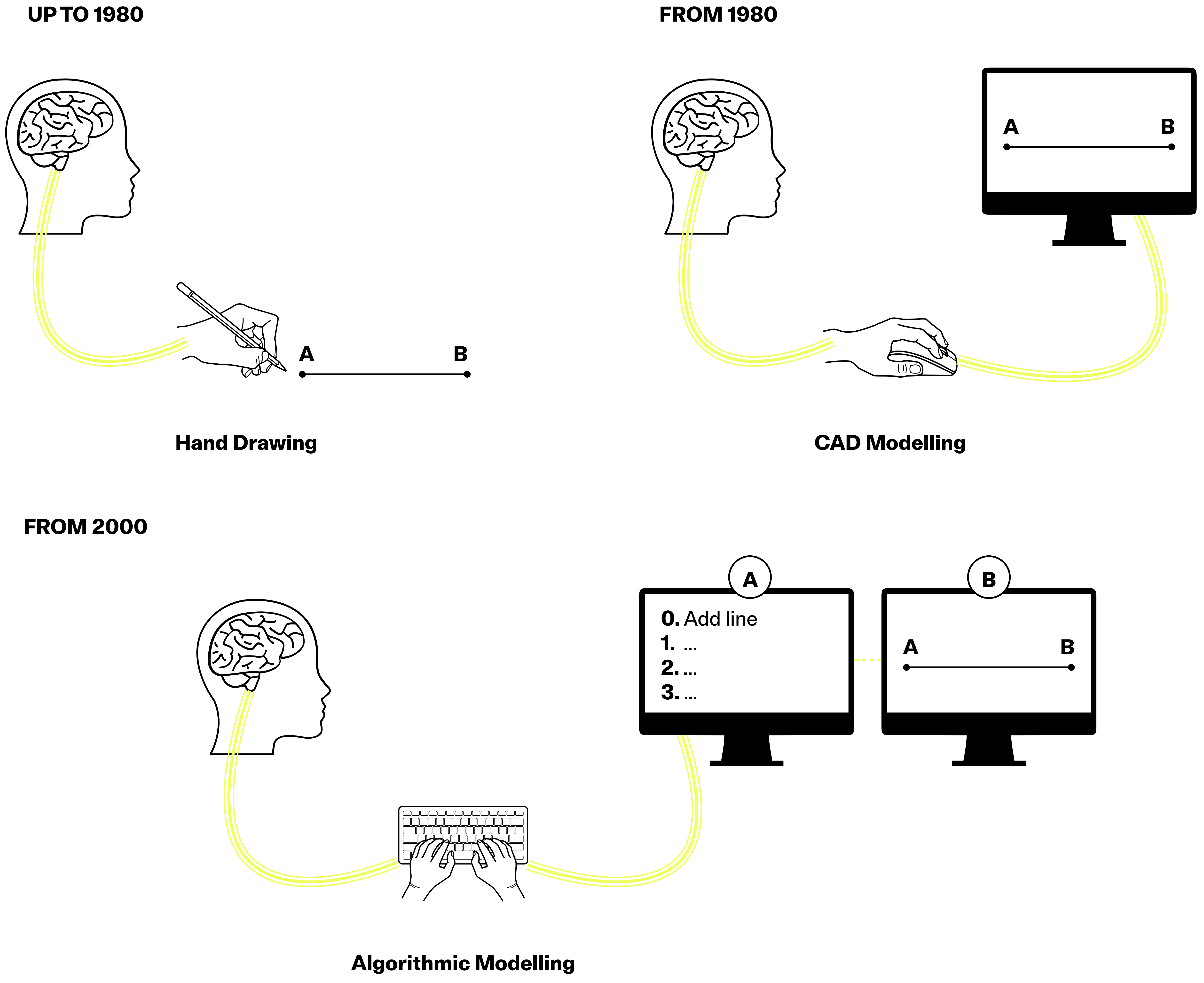


Figure 22. Brain–hand–machine interaction diagram. Adapted from Arturo Tedeschi's AAD Algorithm-Aided Design book

“Digitalization is as inevitable as the Renaissance was after the tools of perspective, as modernism was after movies and trains, as postmodernism was after cars and television.”

Lars Spuybroek,
‘The Architecture of Continuity’

Algorithmic Modelling

Algorithmic-Aided Design (AAD) represents a shift in the way designers conceptualize and generate architectural forms. Unlike traditional CAD, where geometry is directly drawn and manipulated, AAD relies on algorithms - structured sets of instructions that automate design processes. At its core, an algorithm can be defined as a structured sequence of operations designed to solve a problem or execute a specific task.

In architectural design, this translates into rule-based modelling, where defined parameters and their interdependencies guide the generation and control of geometry. Arturo Tedeschi, in his book Algorithms-Aided Design, emphasizes this approach as a mean of encoding design logic rather than merely shaping a fixed form, fostering an interactive and adaptable workflow.

Rather than manually crafting each geometric element, designers define constraints, mathematical functions, and dependencies, allowing the system to produce geometric variations parametrically.

Beyond improving efficiency, algorithmic modelling has given architects new possibilities to expand their architectural language, allowing them to move beyond Cartesian spatial constraints. By embedding rules that govern geometric relationships, this approach has enabled the exploration of fluid, complex forms that were once challenging to conceive or fabricate.

Early pioneers, such as Frank Gehry, leveraged digital tools to push the boundaries of 3D modeling, setting the stage for contemporary algorithmic design.

The algorithmic approach has laid the foundation for generative design in the AEC industry by enabling rapid exploration of design alternatives within a form-finding process. By structuring design logic through parametric rules, architects can generate multiple iterations while maintaining control over constraints and relationships.

Figure 23. MI.C parametric facade design by Park

Evolutionary Design

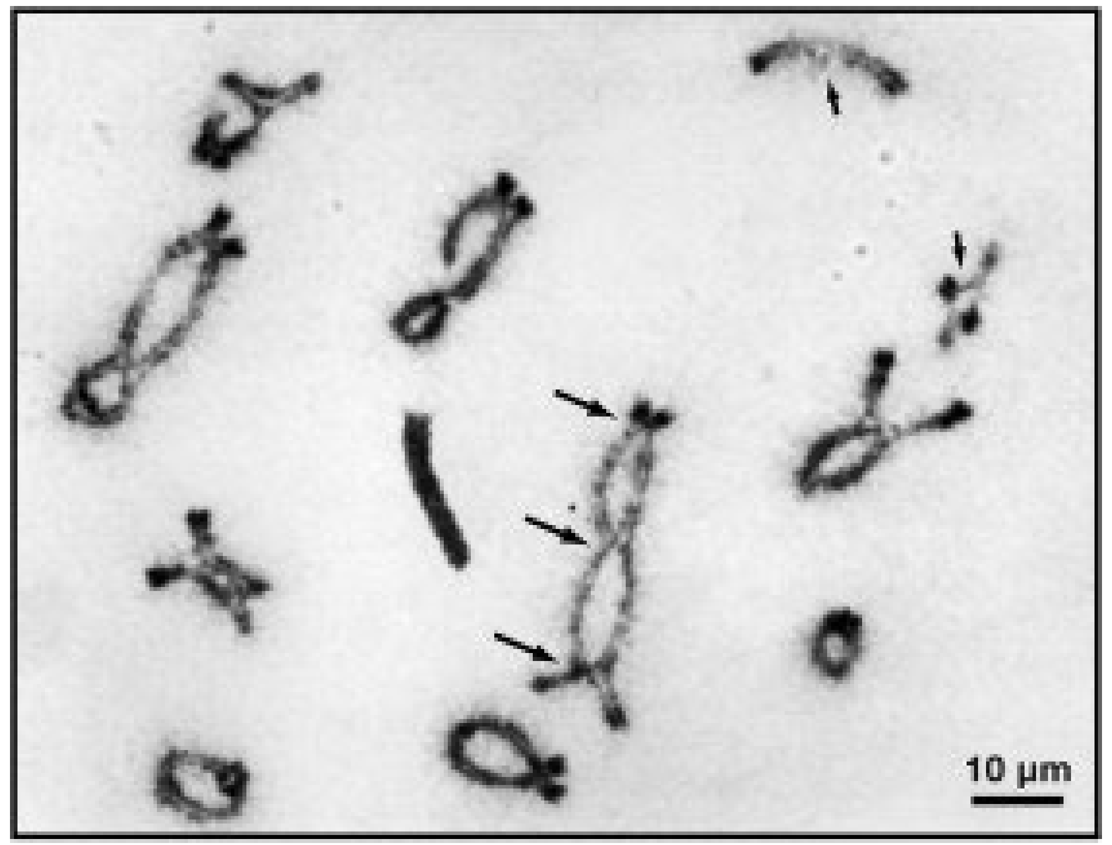
Evolutionary design represent a subset of generative design inspired by biological evolution, where solutions emerge through iterative cycles of generation, evaluation, evolution and selection. Unlike other generative design workflows that generate variations based on predefined rules, an evolutionary approach introduces a dynamic optimization process, enabling solutions to evolve towards optimal performance. Peter J. Bentley and David W. Corne, in *An Introduction to Creative Evolutionary Systems*, outline five fundamental components that define an evolutionary system:

- **Evolutionary Algorithm:** The computational engine responsible for generating solutions through specific rules such as genetic algorithms (GAs).
- **Genotype Representation:** The search space, consisting of all possible combinations of genes (design parameters), which defines the pool from which populations of solutions are generated.
- **Embryogeny:** The mapping process that translates a genotype into a phenotype, effectively constructing the design output from encoded parameters.
- **Phenotype Representation:** The resulting solution, expressed in a recognizable form, such as a geometric configuration.
- **Fitness Function:** A quantitative evaluation metric that assigns a performance score to each generated solution based on predefined design objectives.

The evolutionary process begins with generating an initial population of solutions. Each solution is represented by a genome, which encodes genes corresponding to specific design parameters. These genomes define multiple possible design alternatives, expressed as phenotypes - the outcomes of the generative process.

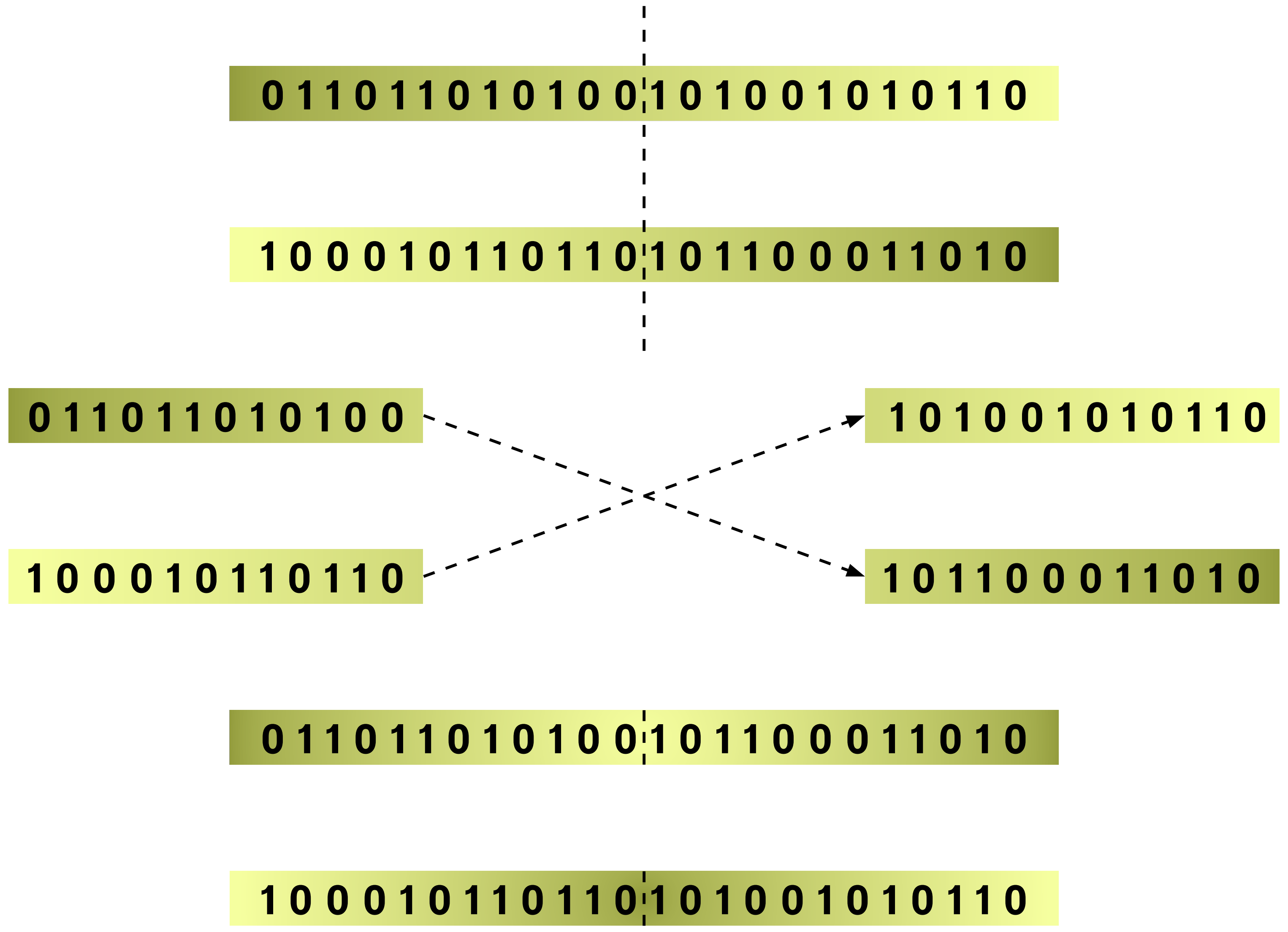
Once the initial population is generated, each phenotype is evaluated using the fitness function, which measures how well a given solution meets the predefined performance criteria. The best-performing designs are selected for reproduction, where genetic operators such as mutation (random alterations to introduce diversity) and crossover (combining elements from different solutions) refine subsequent generations. This cycle continues iteratively, allowing the design population to evolve towards increasingly optimized solutions.

By integrating evolutionary systems into generative pipelines, designers can enhance their ability to explore complex design spaces. Advancing computational power and accessible parametric software make evolutionary design increasingly accessible, providing a robust framework for solving multi-objective challenges in architecture and engineering.



Chromosomal cross-over: mechanism of recombination of genetic material.

Figure 24. Meiotic crossing-over under the microscope



The gene pool of the algorithms consists of 0 and 1 strings. The adaptive value of each string is assessed, and the best strings are paired and generate offspring through crossing over.

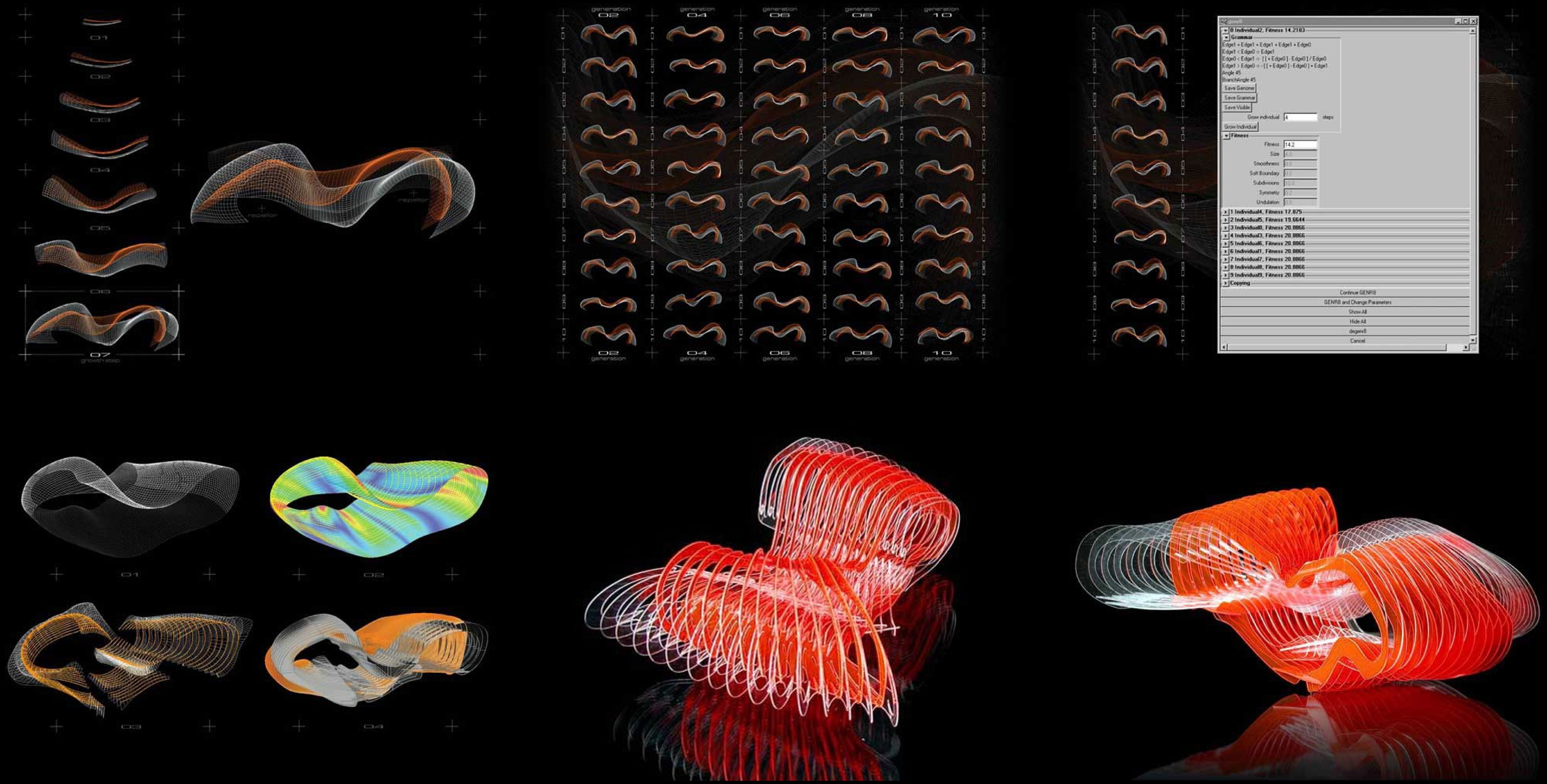


Figure 25. Morphogenetic Design
Experiment 01
by Achim Menges

GENE = VARIABLE

Elementary unit of genetic information

FITNESS = EVALUATION METRIC

Value that describes the ability of a genome to meet a given objective

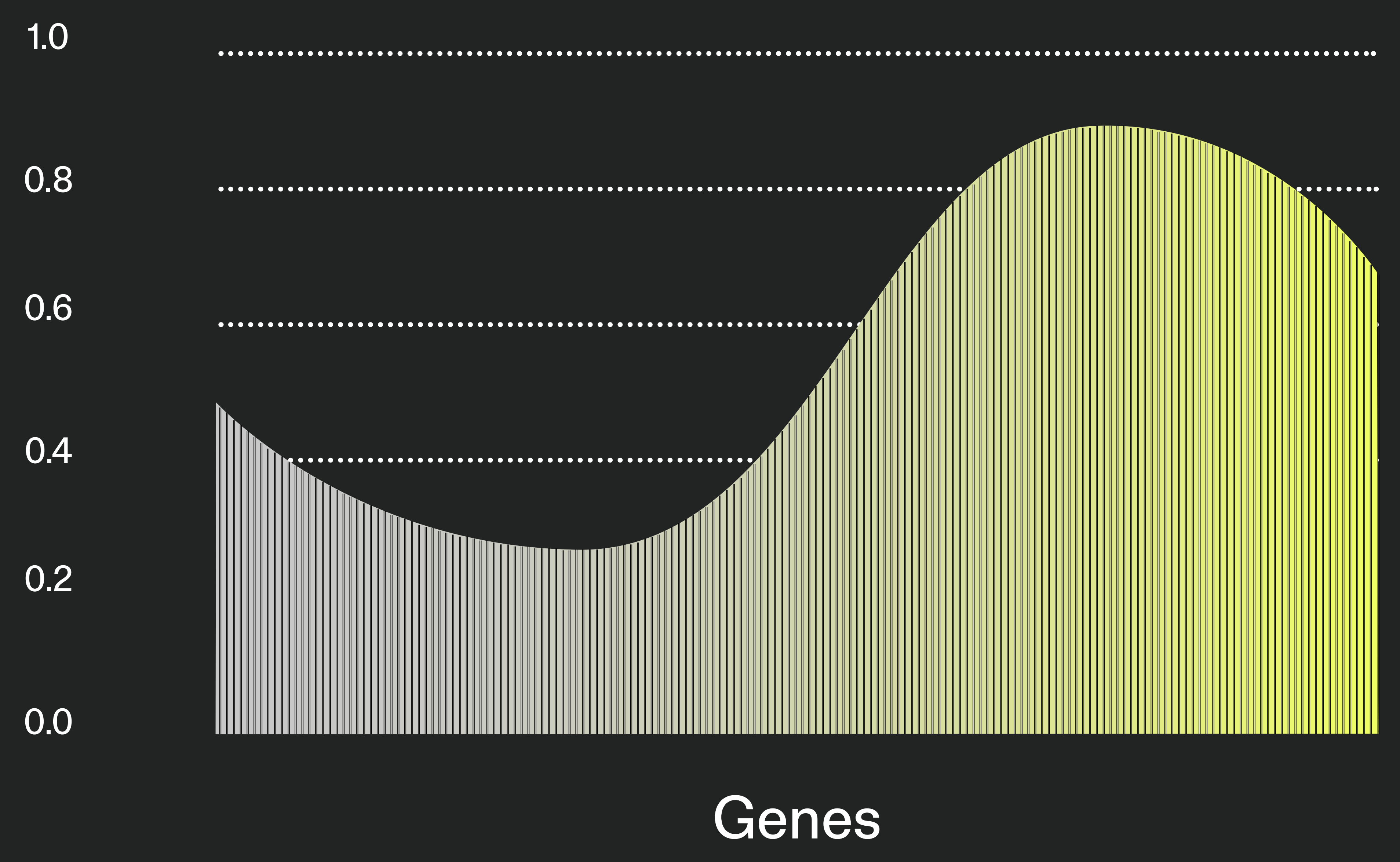
GENOME = SET OF VARIABLES

Set of all genes in an organism

PHENOTYPE = RESULTING SOLUTION

Set of characteristics of an organism, resulting from its genome and the influence of environmental factors

Fitness values



Fitness landscape (1 objective)

Figure 26. An example of fitness function with a single objective set

Generate

The process begins with generating an initial population of design solutions based on predefined parameters. These solutions, encoded as digital genomes, represent a diverse set of possibilities within a defined search space. The generation phase establishes the foundation for exploration by introducing variation in the generate configurations.

EVOLUTIONARY DESIGN PROCESS

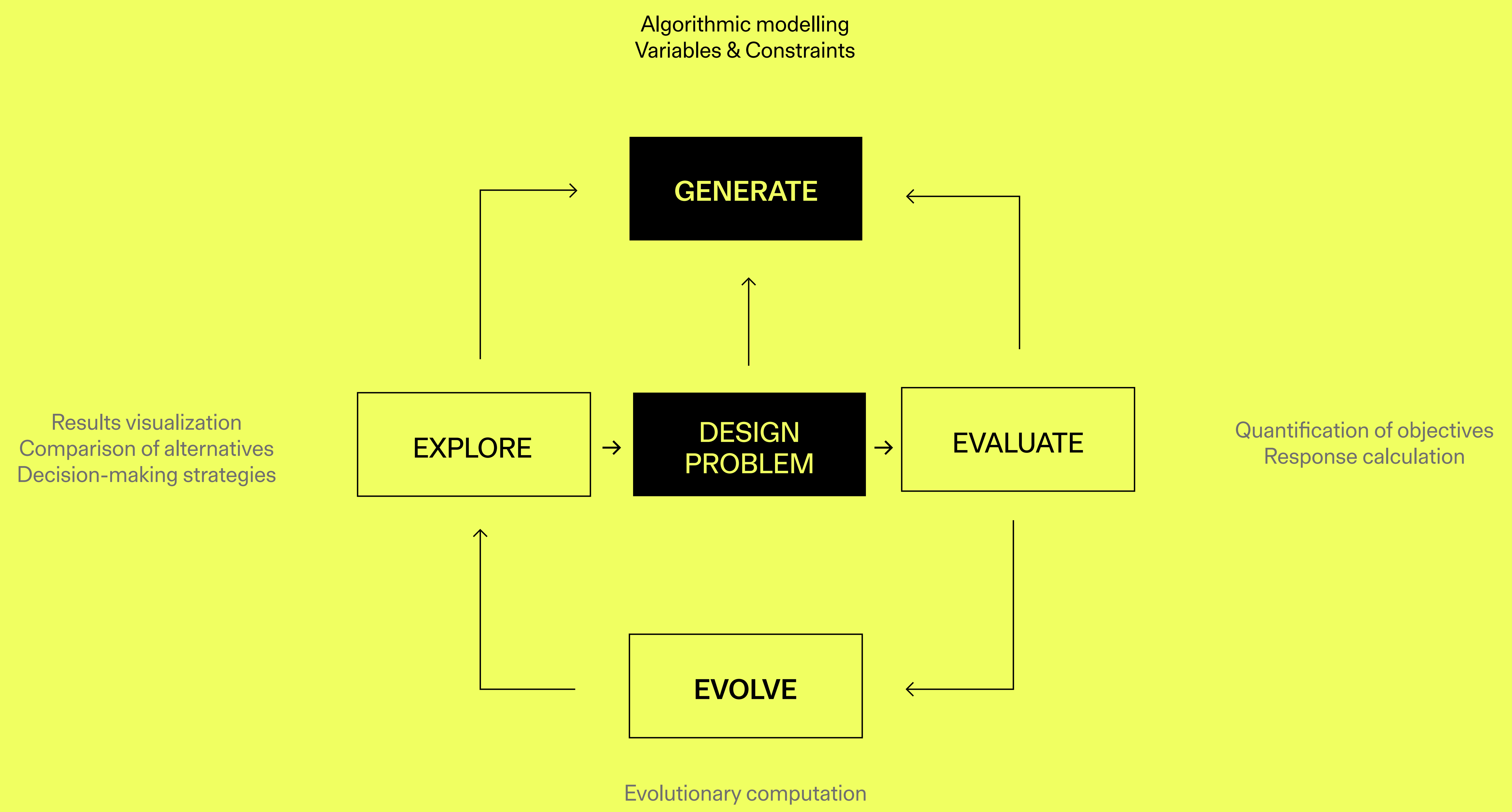


Figure 27. Evolutionary design process flowchart

Evaluate

Each generated solution undergoes evaluation using a fitness function - a computational metric that assesses performance against specific criteria. This step ensures that each iteration is quantitatively measured, allowing the system to identify solutions that best align with the design objectives.

EVOLUTIONARY DESIGN PROCESS

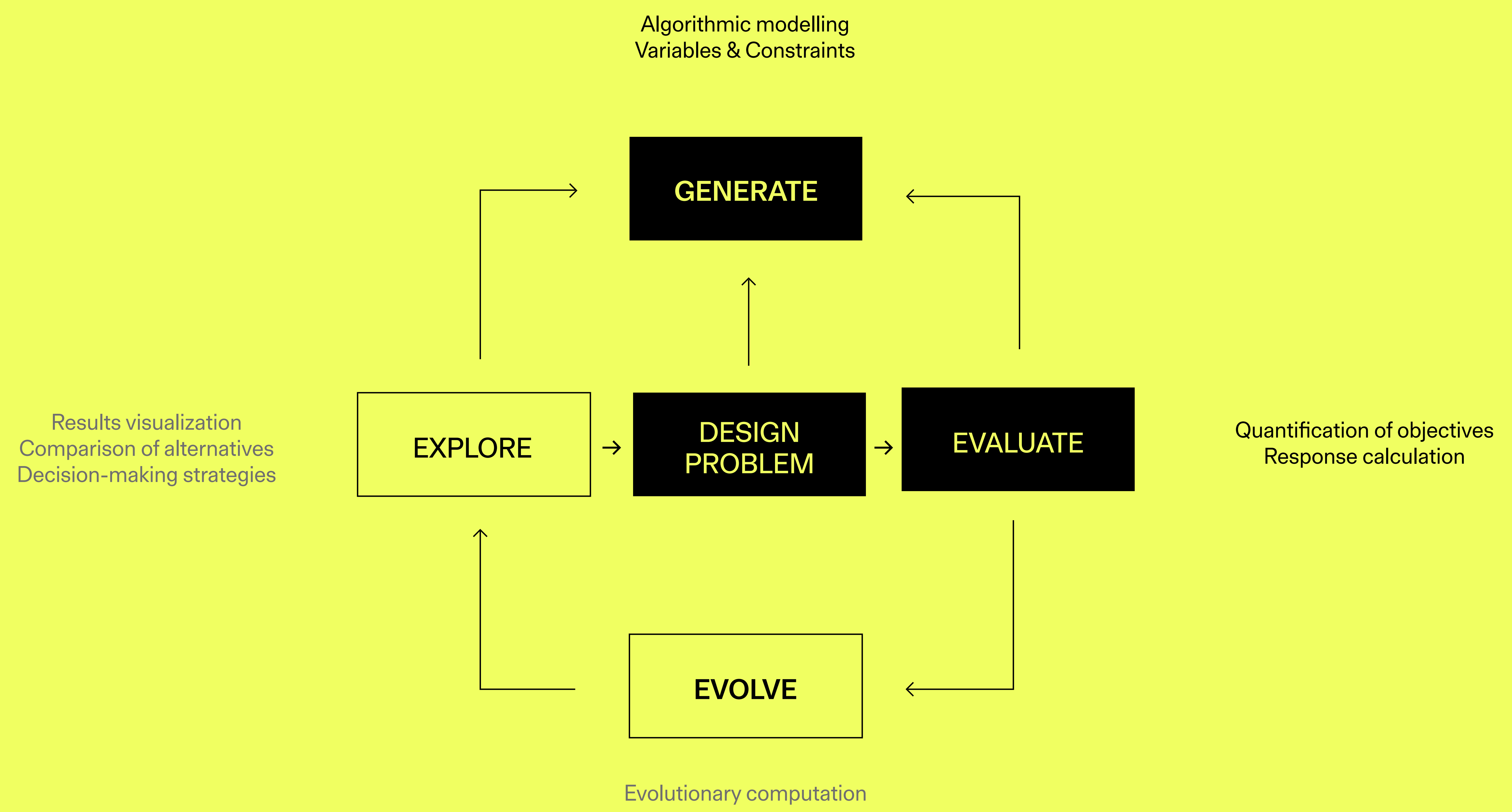


Figure 27. Evolutionary design process flowchart

Evolve

Through selection, crossover, and mutation, the best-performing solutions are combined and modified to enhance their effectiveness. This iterative refinement introduces genetic diversity while progressively optimizing results. Over successive generations, the algorithm steers solutions toward an optimal balance between the various evaluation criteria.

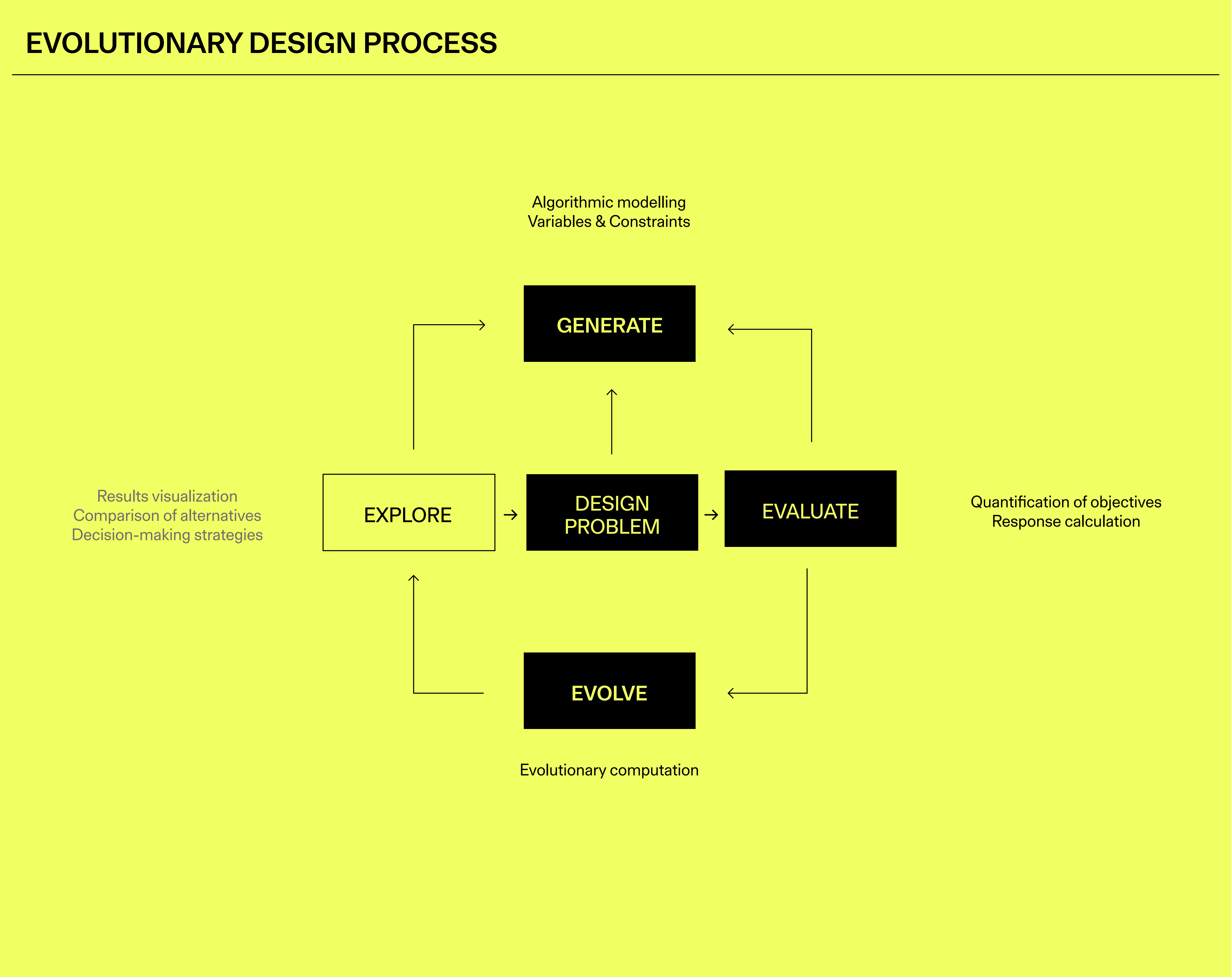


Figure 27. Evolutionary design process flowchart

Explore

Evolutionary design does not converge on a single solution but continuously generates new possibilities. This phase emphasizes the creative potential of this kind of generative workflows, encouraging designers to investigate multiple alternatives and uncover unexpected yet viable solutions that might not have emerged through conventional design approaches.

EVOLUTIONARY DESIGN PROCESS

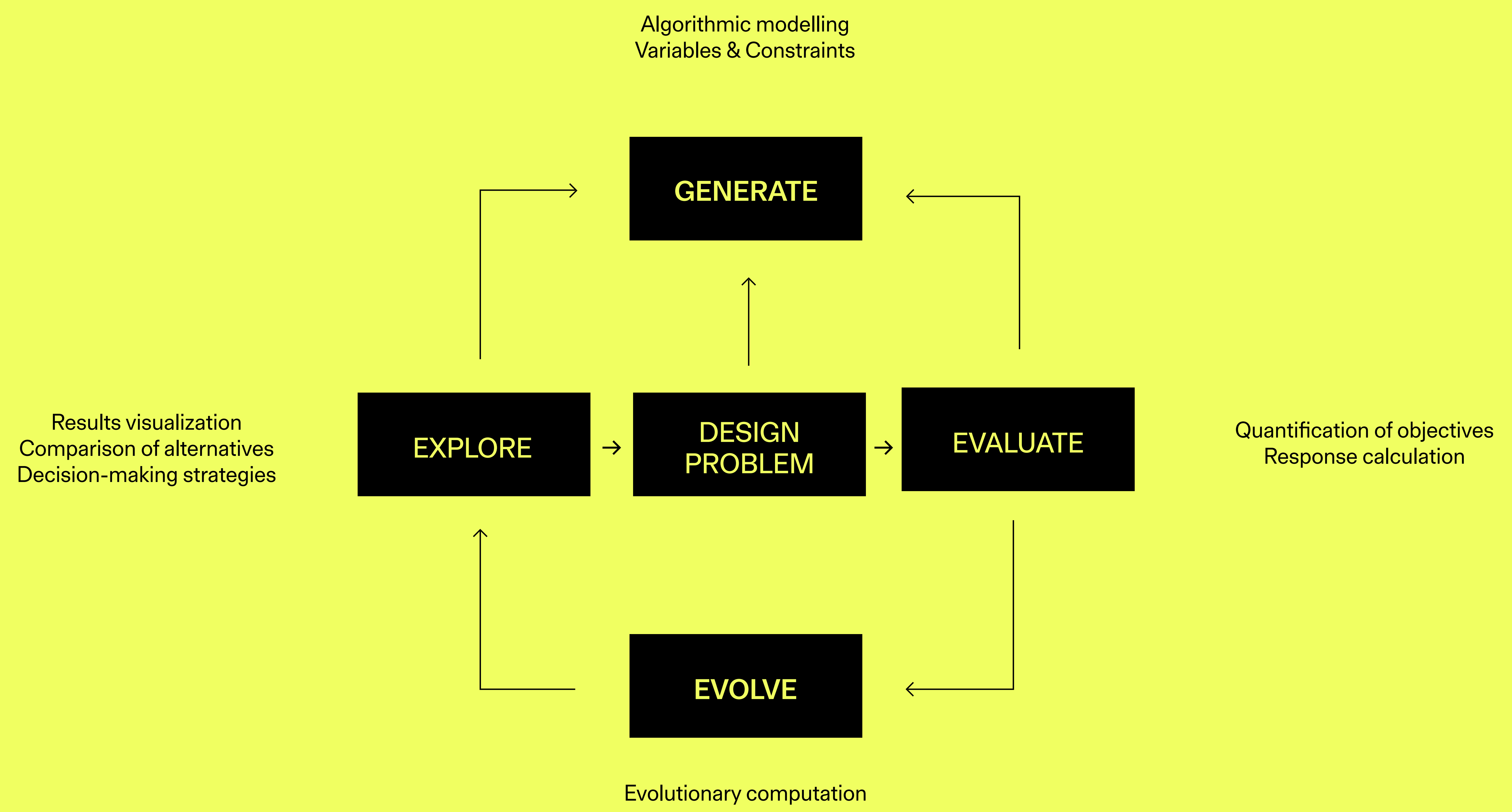
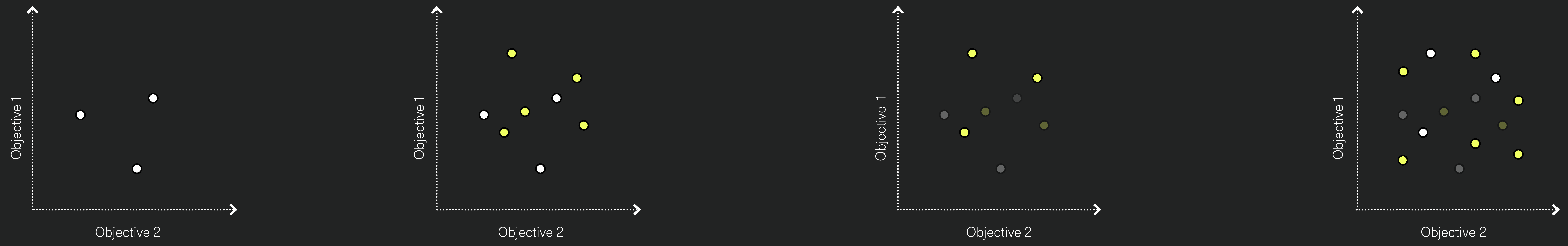


Figure 27. Evolutionary design process flowchart

STARTING POPULATION → **SELECTION BY FITNESS** → **REPRODUCTION** → **SURVIVAL** → **MUTATION**



Starting population

1° Generation

Survival

2° Generation

Figure 28. Optimization process diagram, based on Darwin's natural selection principles

“From the time of ancient Vitruvian geometric ideals to modern Corbusian regulating lines and Miesian modular grids, architecture has always been bound to (if not by) a conscious use of numbers.”

Brett Steele,

‘Weapons of the Gods’ in *The New Mathematics of Architecture by Jane Burry and Mark Burry*

05

Emerging Design Paradigms

Generative design introduces a new dimension into architectural thinking, allowing designers to move beyond traditional workflows and embed computational methods that enhance creativity, optimize performance, and dynamically respond to complex design challenges. Instead of one single methodology, generative design offers a flexible framework that redefines how form, data, and algorithmic processes relate.
This chapter examines how generative design is applied within contemporary practice and provides a framework through which to understand the varied ways in which these strategies take shape in design intent.
We have chosen to highlight three key approaches that illustrate how generative design integrates into the design process and contributes to redefining design methodologies:

- **Form-driven design** utilizes computational tools to generate and manipulate complex geometries, expanding possibilities for spatial articulation and aesthetic expression.
- **Performance-driven design** focuses on optimizing measurable criteria such as energy efficiency, structural integrity and environmental performance.
- **Nature-driven design**, inspired by biomimicry, translates biological principles into architectural strategies that foster adaptive and resilient systems. Whereas each approach prioritizes different aspects of the design process, they do not exclude one another.

Understanding their differences allows architects to strategically apply computational and generative design tools and balance creative exploration with functional optimization to enhance and support the design process.

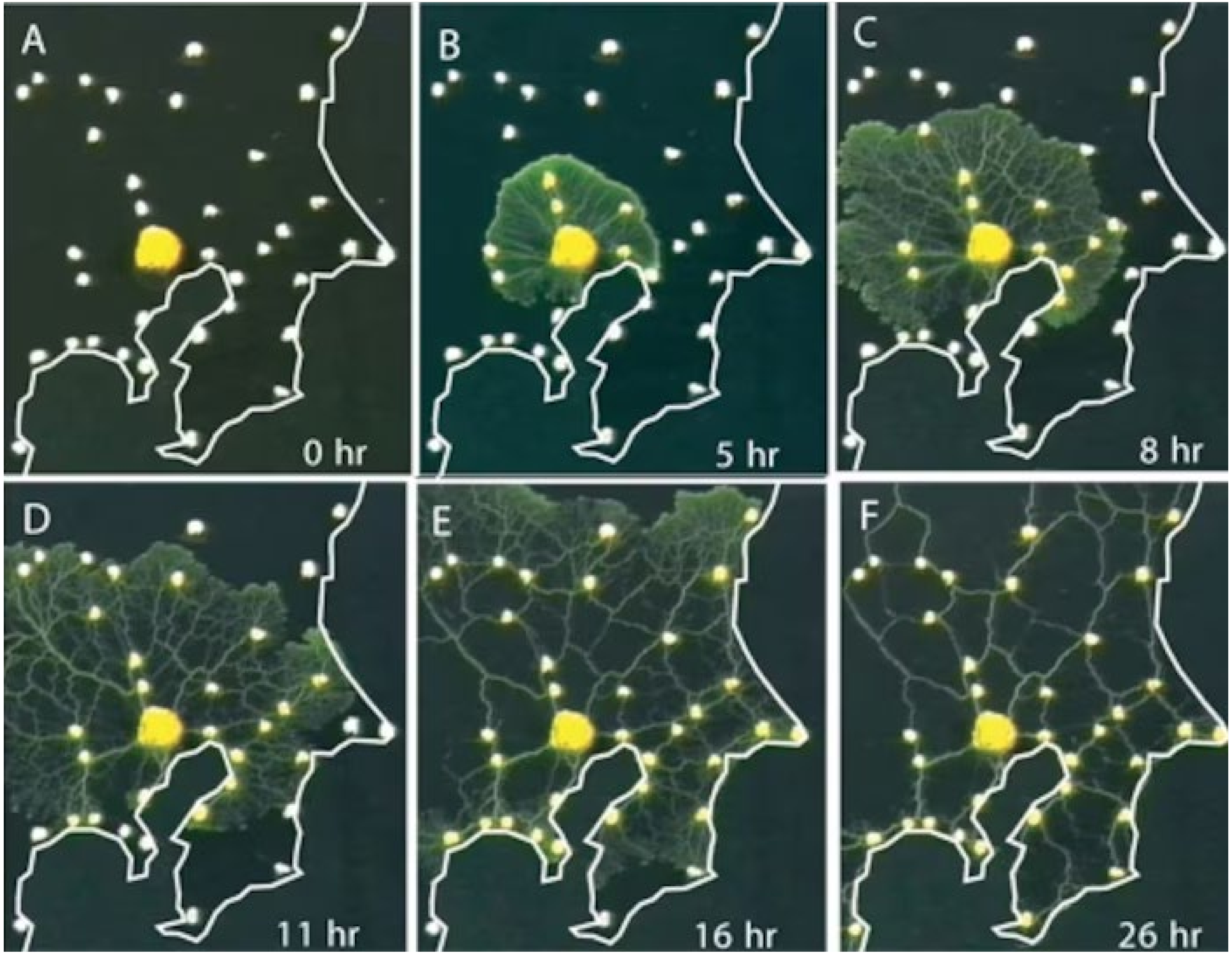


Figure 29. Slime mould mimics Tokyo's railway system

Figure 30. Roger Johnston's fractal art

Form-driven Design

Form-driven design prioritizes geometry, aesthetics, and spatial articulation, leveraging generative design to generate and manipulate formal solutions within a parametric framework that facilitates iterative exploration and refinement of complex forms.

Performance-driven Design

Nature-driven Design

A form-driven design approach focuses on geometry, aesthetics, and spatial articulation, leveraging generative design and computational strategies allows to generate and manipulate formal solutions within a parametric framework.

Unlike traditional approaches, where form emerges from intuition or typological references, generative design is based on rule-driven systems that give more priority to the definition of geometric parameters and computational rules over direct form-making.

Just as architects have traditionally defined composition rules through proportions and modular relationships to structure spatial design, generative design follows a similar logic but shifts the emphasis toward the creation of a framework and process that drive form generation rather than directly shaping the final outcome. This allows for an iterative exploration of possibilities, where compositions emerge dynamically within a structured yet flexible design system.

Generative algorithms, including Voronoi diagrams, Metaball fields, Cellular Automata, and Shape Grammars, exemplify this approach by enabling rule-based form generation and spatial organization.

Implementing Voronoi diagram algorithms, which partition space into regions based on proximity to specific points, can generate intricate, cellular patterns in façades and spatial layouts.

Networked field interactions in metaball algorithms, instead of rigid geometries, create dynamic, organic surfaces that adapt fluidly to spatial constraints.

In this approach, formal expression plays a central role, enabling designers to push the boundaries of architectural languages while maintaining coherence in geometric logic.

Despite its primary focus on aesthetics and spatial complexity, this approach does not operate in isolation; it intersects with performance and contextual considerations, ensuring that formal generation remains purposeful and adaptable.

Form-driven Design

An architectural expression of natural patterns, inspired by the intricate, interwoven structure of a bird's nest.

Figure 31. Olympic Stadium Beijing by Herzog & de Meuron

A design inspired by soap bubble geometry, reflecting self-organizing systems for structural efficiency.

Figure 32. Water Cube by PWT Architects

Features a facade inspired by cellular structures, combining aesthetic complexity with functional performance.

Figure 33. Torre de Especialidades by Elegant Embellishments

Form-driven Design

Performance-driven Design

Performance-driven design prioritizes measurable criteria, using generative processes to optimize structural, environmental, and spatial performance while balancing efficiency with design intent.

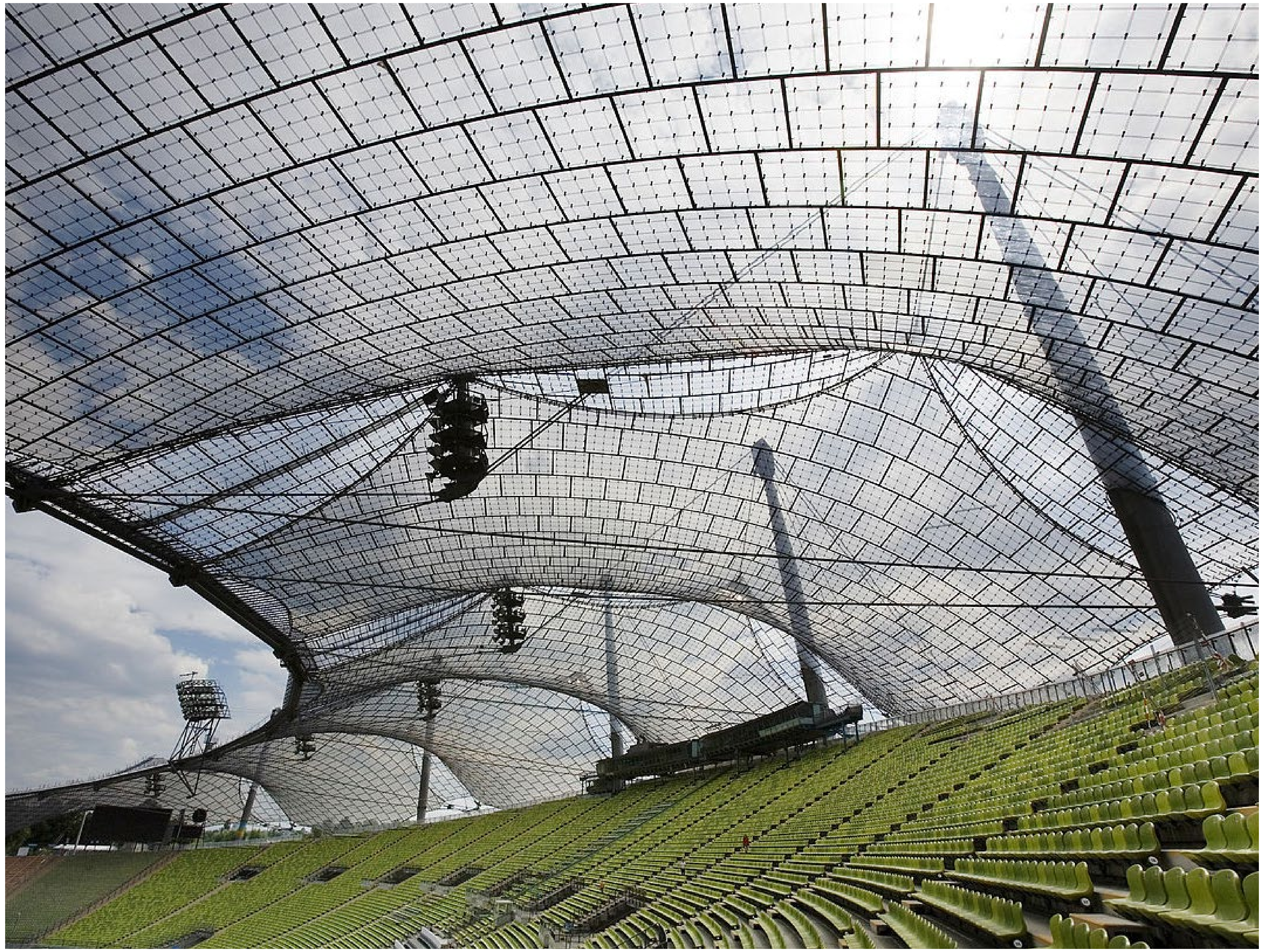
Nature-driven Design

A performance-driven design approach shifts the emphasis from purely aesthetic or spatial exploration to optimizing measurable criteria such as energy efficiency, structural integrity, and environmental performance. Generative design becomes a form-finding tool, leveraging computational techniques for evaluating and improving solutions based on quantifiable data. Rather than defining a fixed form from the outset, this approach sets performance objectives and allows algorithms to explore and evolve optimal design configurations.

Parametric models play a critical role by enabling dynamic simulations, real-time feedback, and iterative testing of multiple variations. Computational tools like structural optimization algorithms and daylighting simulations, help architects make data-driven decisions from the early design stages. Evolutionary algorithms further develop this process by simulating natural selection, iterating through generations of design variations to identify solutions that best balance competing performance criteria.

A key benefit of this approach is its ability to resolve complex, multi-variable challenges—such as maximizing natural lighting while minimizing solar heat gain or optimizing structural weight while ensuring stability. By embedding performance-based evaluation into generative workflows, architects can develop solutions that are not only visually appealing but also highly efficient, functional, and responsive to changing environmental conditions.

Performance-driven Design



The tensile membrane structures for the Munich Olympic Stadium by Frei Otto reflects an early performative design approach, rooted in physical form-finding experiments that predate the digital era yet align with contemporary generative principles.

Figure 34. Munich Olympic stadium canopy by Frei Otto

The diagrid canopy of the British Museum's Great Court by Foster and Partners was designed using optimization techniques to refine the roof's geometry and structural efficiency.

Figure 35. British Museum's Great Court by Foster and Partners

Form-driven Design

Performance-driven Design

Nature-driven Design

Nature-driven design applies biomimetic principles, using generative processes to develop innovative architectural solutions inspired by natural systems.

Nature-driven design, often referred to as biomimicry in architecture, draws inspiration from biological processes, patterns, and ecosystems to develop innovative design solutions.

Rather than imposing man-made forms on the built environment, this approach looks into how nature optimizes structures, materials, and systems to develop functional and sustainable designs. Generative design plays a crucial role in this process, enabling architects to translate biological principles into computational models that can emulate natural growth, self-organization, and material intelligence.

Parametric tools and simulation software allow designers to analyze and replicate biological strategies such as structural efficiency found in bone formations, aerodynamic shaping inspired by animal morphologies, and material adaptability seen in plant behavior. Through iterative computation, architects can generate solutions that evolve in response to environmental forces, optimizing performance while maintaining organic aesthetics.
This approach has inspired innovative architectural solutions that echo nature's intelligence—façades that adjust like a tree's canopy to provide shade, structures that mimic the strength and efficiency of cellular formations, and systems that dynamically respond to shifting environmental conditions. By embedding these principles into generative workflows, nature-driven design fosters an architecture that is not only efficient but also deeply intertwined with natural processes.

Nature-driven Design

A biomimetic design inspired by soap bubbles and geodesic patterns, adapting to an irregular quarry site while optimizing solar gain and structural efficiency.

Figure 36. The Eden project by Grimshaw

The building's distinctive lattice-like exoskeleton mimics that of the Venus's flower basket sea sponge and helps power the natural ventilation system, reducing the need for air conditioning.

Figure 37. 30 St Mary Axe by Foster and Partners

A biomimetic structure inspired by beetle wing covers, showcasing fiber-composite design, computational fabrication, and resource-efficient architecture.

Figure 38. ICD-ITKE Research Pavilion 2013-14

“Process is more important than outcome. When the outcome drives the process we will only ever go to where we’ve already been. If process drives outcome we may not know where we’re going, but we will know we want to be there.”

Bruce Mau
‘Incomplete manifesto of growth’, 1998

06

Case Studies

Design Technology at Park

In the AECO industry (Architecture, Engineering, Construction, and Operation), the term "Design Technology" refers to the research, development, and implementation of emerging digital technologies aimed at conceiving, designing, constructing, and managing a project.

The Design Technology department at Park is composed of a team of in-house BIM and Computational Design specialists that investigate and implement innovative digital design tools across the office in a collaborative manner.

Experimentation and collaboration are the two core principles driving the Design Technology department, whose main objective is to support project teams in achieving efficient, informed, and collaborative design outcomes. Leveraging its expertise, the group works alongside other departments to develop tailored digital tools and to implement innovative methodologies that enhance the design process across all project phases, from urban planning to product design. Alongside its regular activities, it carries out applied research to identify opportunities for incorporating new workflows and digital tools into design operations, while also organising training sessions to disseminate knowledge within the practice, fostering a culture of digital innovation across teams.

Figure 39. Park Design Technology workflows palette

Building Information Modelling

From the early stages of the design process, Park approaches new projects by implementing the Building Information Modeling (BIM) methodology. This ensures efficient management and coordination of project data among all parties involved in the design process, facilitating communication and collaboration. At the core of this approach is the information model, which plays a pivotal role throughout all design phases by housing all relevant project data. The Design Technology department plays a crucial role by providing daily support to project teams, ensuring adherence to quality standards and effective creation and management of information models. This ongoing assistance is key in streamlining workflows, addressing technical challenges, and ensuring alignment with project objectives.

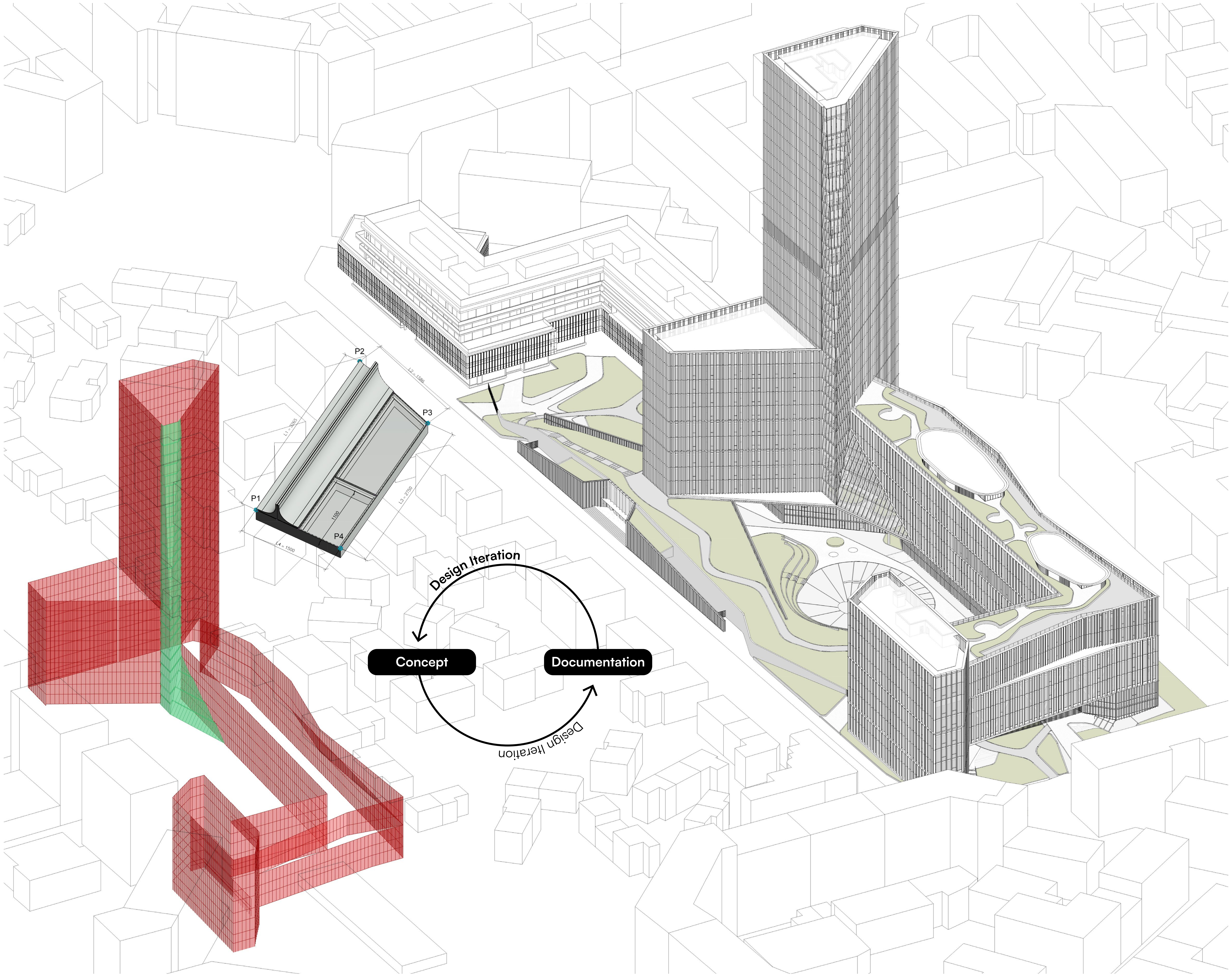


Figure 40. Interoperability approach for Palazzo Sistema's facade design by Park

Computational Design and R&D

The use of Computational Design tools applied to project workflows blends with the office ethos and design culture. The digital tools developed by the team are characterized by their innovative and technological capabilities, designed to facilitate collaborative and versatile use across design teams. Currently, these tools are being deployed for various use cases, including parametric modelling for design optioneering, workflow automation and software interoperability, as well as optimization of complex geometries and integrated environmental performance analysis. Design Technology at Park also provides a safe space for exploring and experimenting with emerging digital technologies, including innovative solutions in XR, UI design, and generative AI.

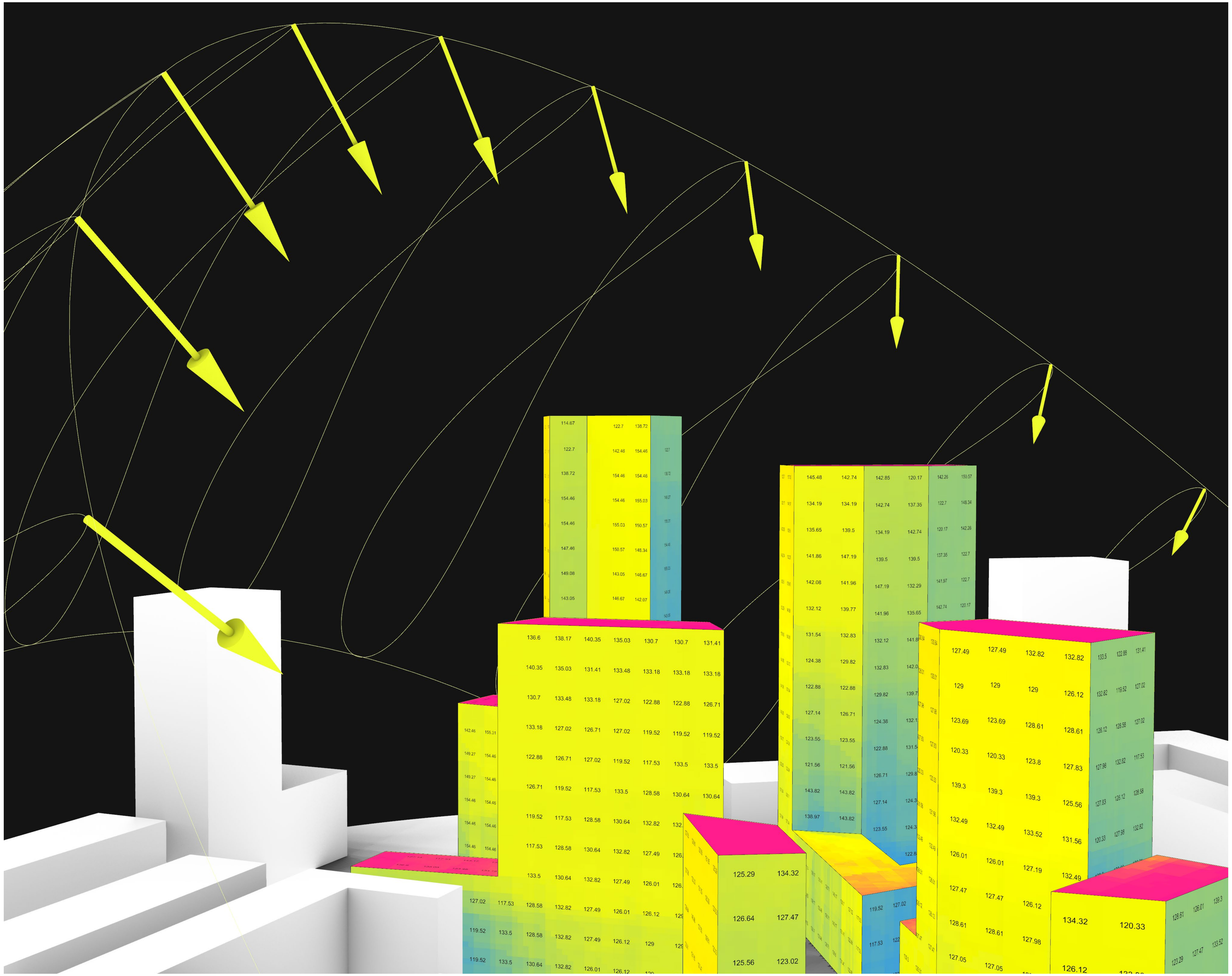


Figure 41. Environmental performance analysis of a massing scenario

Our approach to Generative Workflows

At Park, our approach to computational and generative design transcends the boundaries of formalism; it is a methodology aimed at amplifying the design intent and enhancing project outcomes. For us, these tools are not just meant to create and manage complex geometries but strategic instruments to address specific design challenges, improve efficiency and unlock creative potential.

The following case studies showcase how we at Park have leveraged generative design methodologies to address diverse architectural challenges. The two examples below not only illustrate the potential of these tools to unlock innovative solutions but also the challenges we navigated in developing tailored workflows to suit each project's unique demands.

Figure 42. Park - Milan studio

Building Massing Exploration

One of the key design challenges we face whenever we embark on the concept phase of a new building project is the articulation of the building's massing and its relationship with the surrounding context.

Defining how volumes interact with adjacent spaces, respond to environmental conditions, and integrate within the urban fabric is a complex process that often requires balancing multiple, sometimes conflicting, factors. In the example, we illustrate how, for this specific project, the use of generative design tools enabled us to conduct in-depth evaluations of how the building's massing interacts with its context and to move beyond intuitive decision-making, providing data-driven insights that informed the development of a massing strategy aligned with both design intent and contextual dynamics.

Project overview

This project is a proposal for a new mixed-use masterplan development in Bratislava, that reinterprets some of the archetypal elements drawn from historical Italian cities, integrating covered arcades and residential towers to create a dynamic urban fabric. The design emphasizes pedestrian permeability, transforming the block into a car-free zone with interconnected galleries that foster vibrant public spaces. Residential volumes are concentrated in five towers set on a three-story plinth, allowing for optimized views of Bratislava's landmarks, including the historic city, the city castle, and the Danube River. Generative design tools were employed to evaluate multiple massing scenarios, optimizing tower heights and placements to maximize panoramic views, solar exposure, and urban integration.

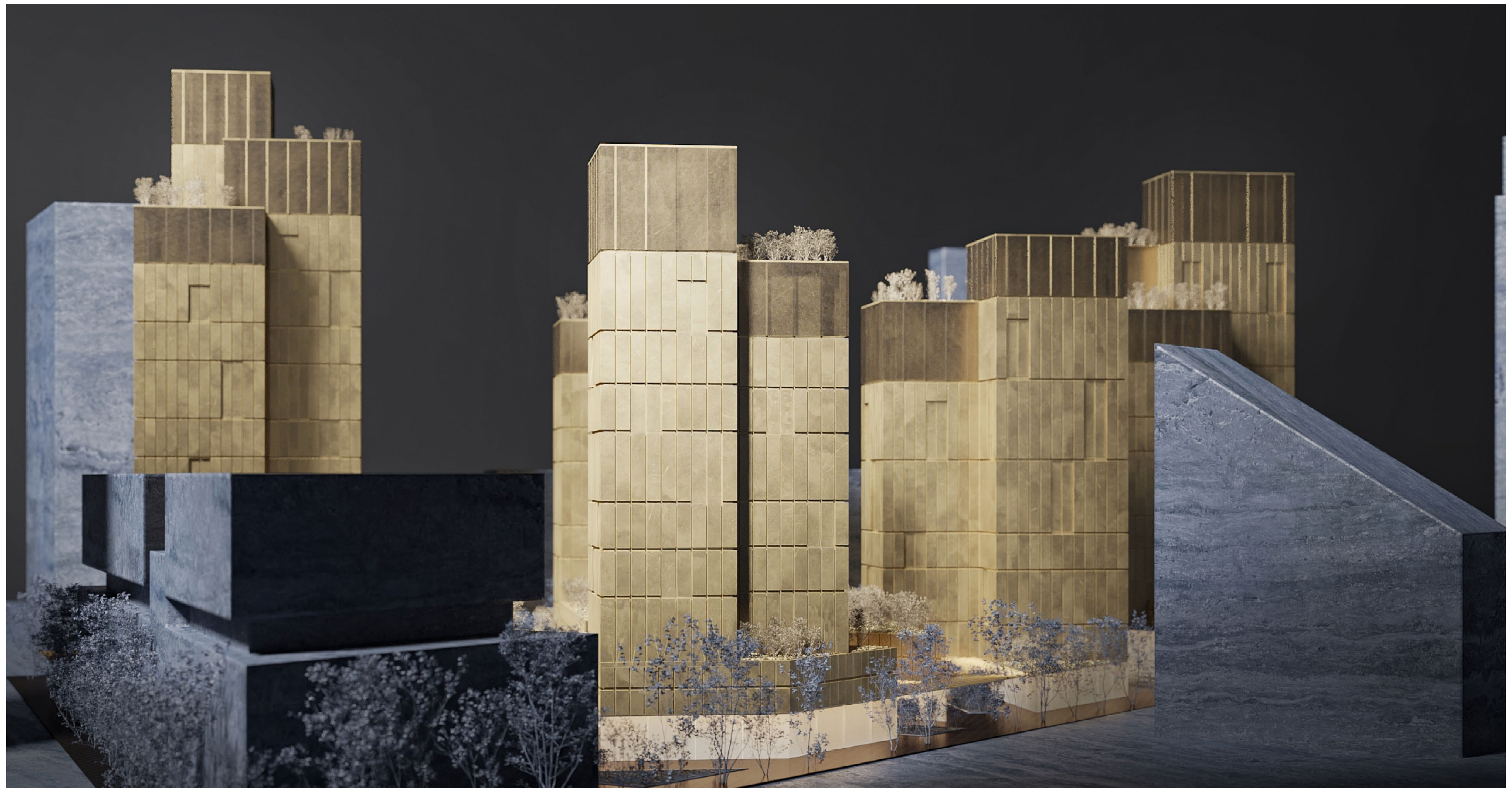
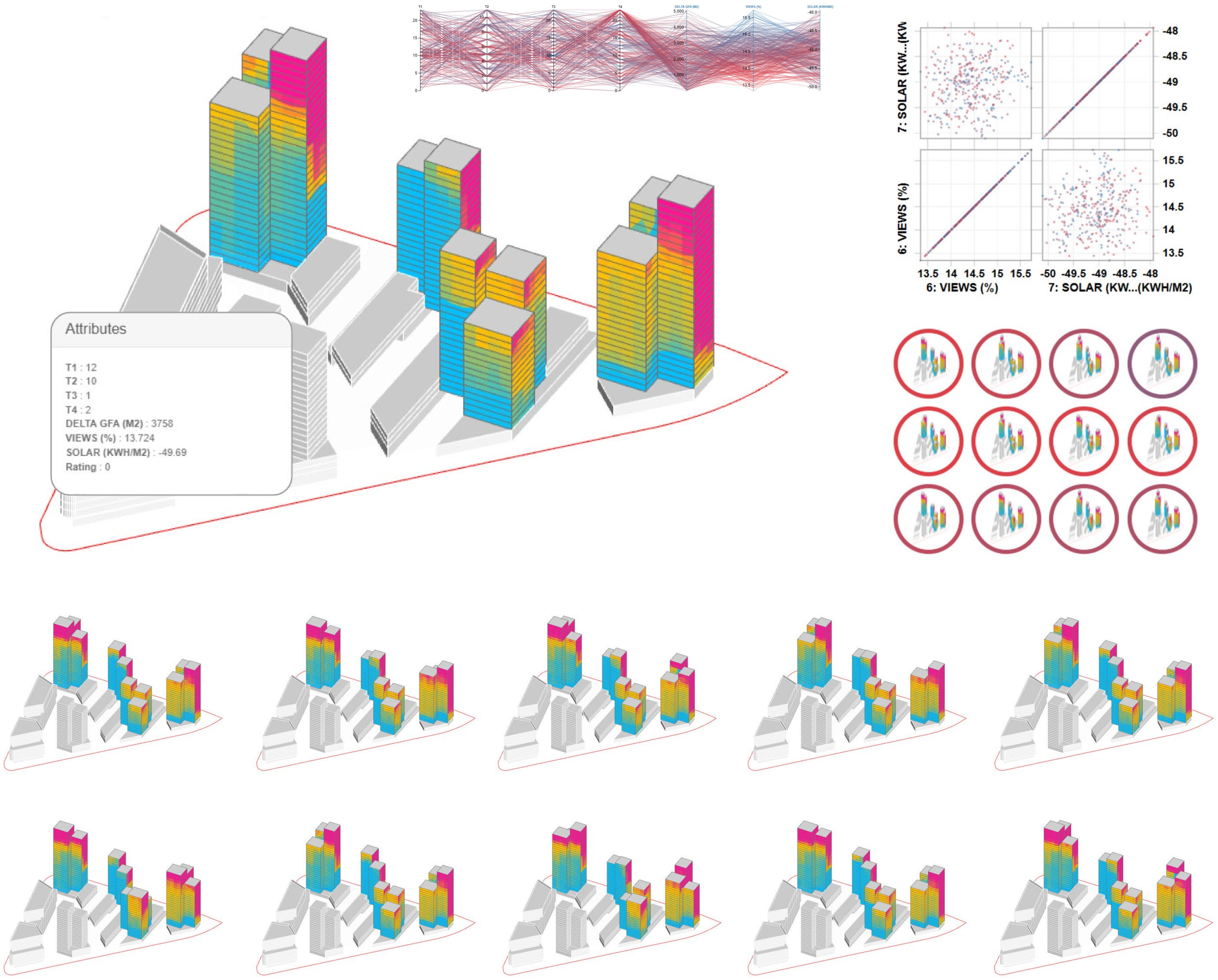


Figure 43. (above) Bratislava masterplan aerial concept view
Figure 44. (right) Bratislava masterplan concept view



Workflow

The form-finding process for this mixed-use masterplan began with the selection of an archetypal building typology, serving as the foundation for exploring massing strategies. The generative design process focused on optimizing the positioning and height distribution of the residential towers in relation to key contextual factors, including the ground-level public realm, panoramic views, and solar exposure.

The first step involved developing a parametric model designed to test alternative tower locations based on their interaction with pedestrian flows and connections to public spaces. This model enabled rapid exploration of different configurations, providing insights into how the towers could best integrate with the surrounding urban fabric while enhancing the vibrancy of the public realm.

Once the optimal positioning of the towers was identified, the focus shifted to the strategic allocation of residential Gross Floor Area (GFA) to meet the client’s brief while allowing for massing articulation that responded to the site’s environmental and urban context. The objective was to achieve a balance between maximizing views, optimizing solar performance, and maintaining urban coherence. Three key parameters guided the generative process:

1. Maximizing Exposure to Quality Panoramic Views
2. Minimizing Incident Solar Radiation
3. Meeting GFA Targets

By integrating these parameters into a generative framework, multiple massing scenarios were generated and evaluated. This iterative process enabled the design team to identify the most efficient and contextually responsive configuration to further develop.

Figure 45. Matrix of the selected generative massing solutions

Facade Geometry Exploration

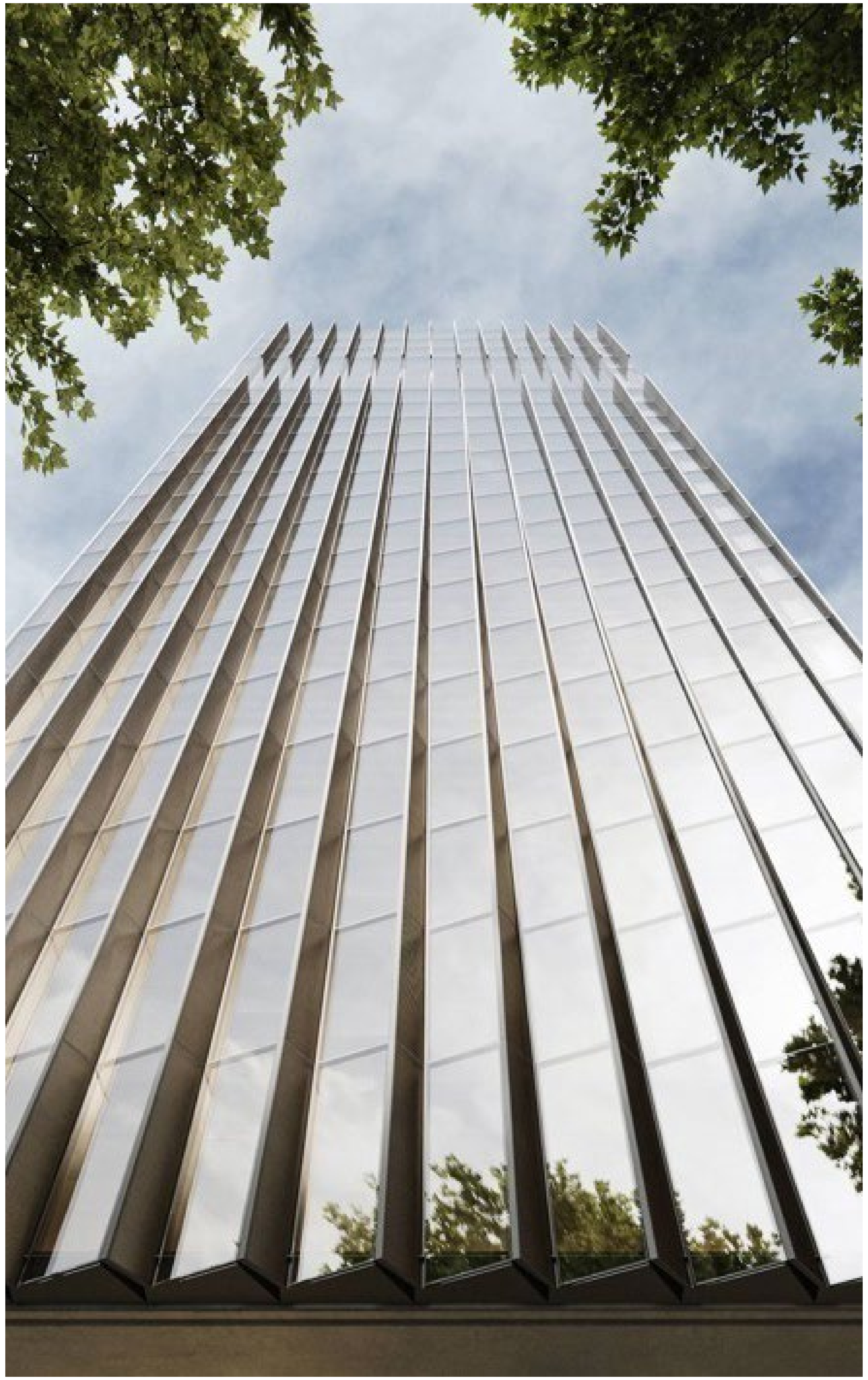
The following example focuses on the application of a generative design process to the development of a building façade, tracing the integration of these tools from the earliest concept design stages. In this case, the design process aimed to create a façade that not only responded to functional and aesthetic goals but also engaged in a dialogue with its surrounding context. By incorporating data-driven insights, generative design supported the exploration of solutions that reflect and respond to environmental and spatial conditions, helping to shape a façade that is both dynamic and contextually aware.

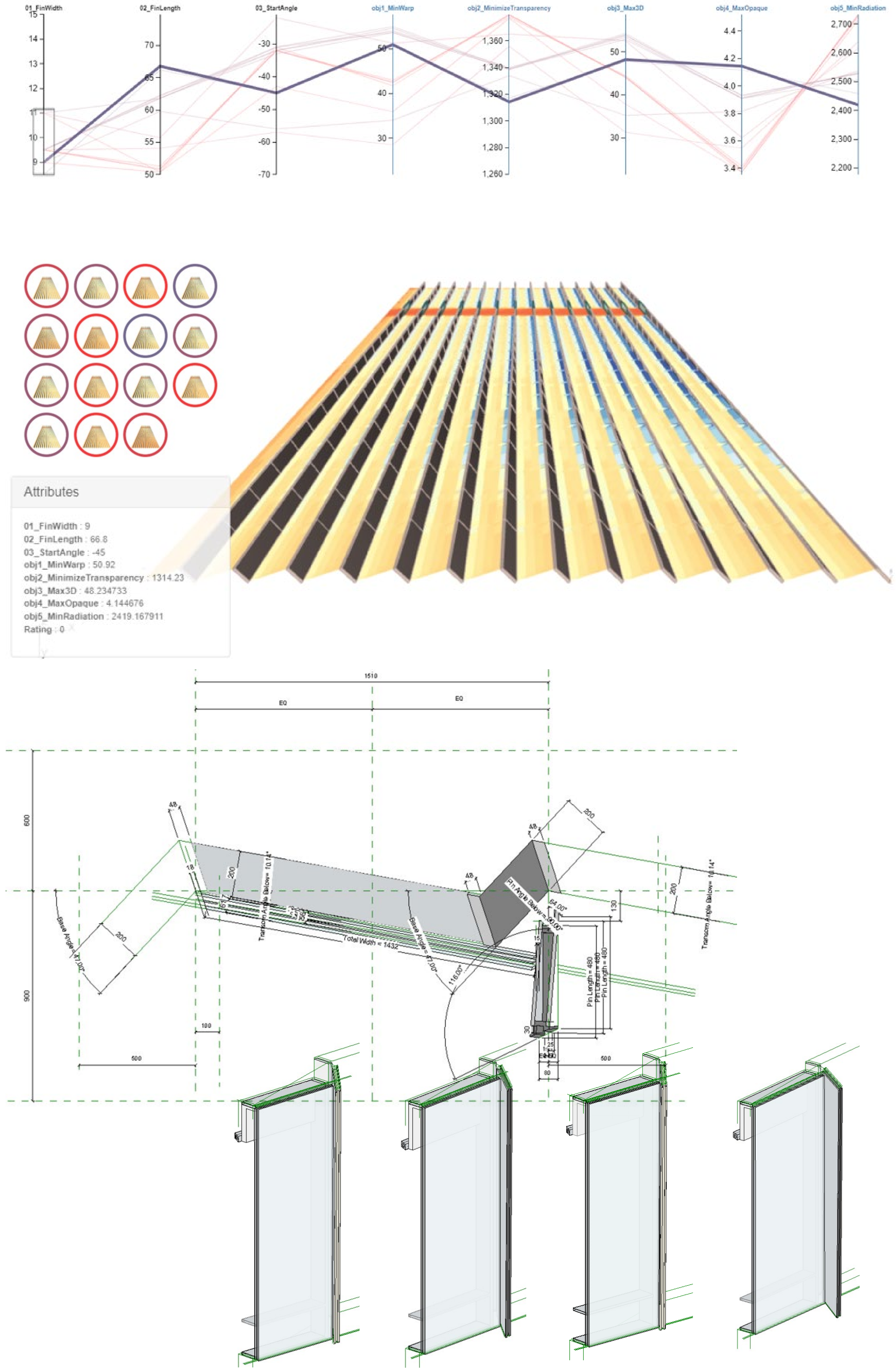
Project overview

MI.C is an architectural design project aimed at regenerating Milan's Stazione Centrale area, transforming the iconic Hotel Michelangelo into a vibrant new architectural complex.

The design features two adjacent towers seamlessly blending with Milan's urban landscape. The façade of the towers, which is the most dynamic element of the project, adapts and changes in accordance with the building's daily life. As well as being a motif of architectural design, glass plays a starring role thanks to cusp-shaped elements – some opaque, other partially transparent – that change their inclination vertically. This vertical variation is carefully designed to respond to the dual objectives of maximizing openings towards the surrounding views and minimizing heat gain and solar radiation. At higher levels, where panoramic views are most desirable, the façade becomes more open and transparent, framing vistas of the cityscape. Conversely, at lower levels, where privacy and shading are more critical, the façade incorporates increased opacity and tighter angles to shield interiors from excessive sunlight.

Figure 46. MI.C by Park





Workflow

The initial concept sought to create a building with a façade that acts as an dynamic element, responding to both the environmental conditions and building programme. This resulted in the development of a geometric language capable of adapting and varying its configuration to meet changing needs as the height of the towers increases.

A parametric model was developed to facilitate control over the façade's geometry, enabling the testing of various design alternatives. This model was developed around three key variables—width, length, and rotation angle of the fins—which directly interact with the glazed elements.

The next step involved employing a generative design methodology, driven by the need to control and optimize specific metrics to achieve geometrical outputs aligned with the broader project goals:

1. Minimizing glazed surfaces
2. Minimizing Solar Heat Gain
3. Optimizing façade depth and glass warp

These metrics were integrated into the same parametric model, allowing for the automated generation and evaluation of multiple design variations. A genetic algorithm (GA) was employed to simultaneously optimize these criteria, enabling an iterative optioneering process where each solution was assessed and ranked.

The preferred design solution was selected by the designers, based on the algorithm's evaluation of the trade-offs between the various objectives, ultimately shaping the façade geometry.

Figure 47. (left) MI.C facade design outcome
Figure 48. (right) MI.C facade solutions assessment, selection and development

Key Takeaways and Beyond

The case studies presented highlight both the potential and the challenges of implementing generative design principles in real-world projects.

One of the most significant hurdles is the resource-intensive nature of this approach. Developing project-specific generative workflows demands significant time and computational power to set-up and deploy which can strain tight project schedules. Additionally, the lack of holistic models limits the ability to address multiple design challenges simultaneously, as current tools focus on specific, easily quantifiable parameters.

Furthermore, the iterative nature of generative design adds other complexity—each change in geometric concepts or design objectives triggers new solution cycles.

To optimize this process and maintain coherence throughout project development, it is essential to define key design objectives and embed meaningful parameters relevant to the project's specific goals right from the beginning of the design process. By focusing on these from the start, generative models can produce design outputs that stay aligned with the original intent while addressing practical constraints, such as constructability, as the design evolves.

When thoughtfully integrated, generative design tools serve as a powerful asset within the design process. They not only support the pursuit of design intent through data-driven insights but also enhance the efficiency of workflows. Moreover, they facilitate the exploration of design possibilities that may remain undiscovered through traditional design methods.

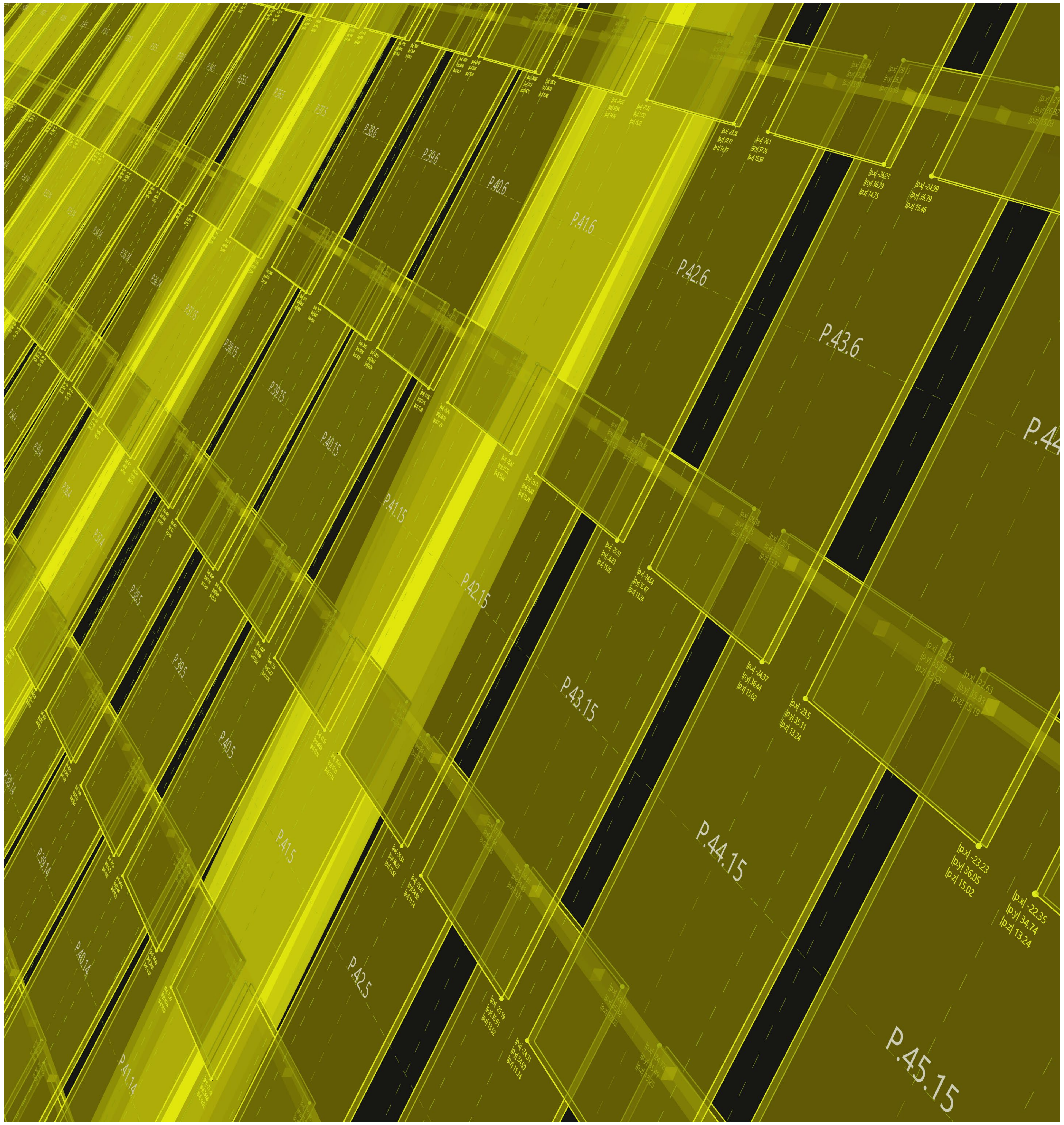


Figure 49. Geometry rationalization of a double-curvature canopy

Looking ahead, the advent of Machine Learning and the more recent emergence of genAI tools in the AEC field offer exciting opportunities while raising important questions about their integration into more established computational and generative design workflows.

Beyond the current hype, AI holds the potential to manage complex large-scale data and support predictive design and analysis models, that can further enhance the design process. However, its adoption requires a critical and ethical approach to ensure it truly enhances design processes—a complex challenge that is still unfolding and requires further exploration.

Figure 50. Park Hub 3D point cloud model

Method over style.

“Style is accustomed to aging, but method endures. We prioritize method as the foundation for innovation, drawing from the past to create transformative and enduring environments that meet tomorrow’s challenges.”

Park

07

Further Readings

Further Readings

Agkathidis, Asterios. Generative Design: Form-Finding Techniques in Architecture. Laurence King Publishing, 2016.

Bucci, Federico, and Marco Mulazzani. Luigi Moretti: Works and Writings. Translated by Marina deConciliis, Princeton Architectural Press, 2002

Carpo, Mario. The Alphabet and the Algorithm. The MIT Press, 2011.

Cocco, Pio Lorenzo, and Cognoli Roberto. Module 5.5_Material Behaviors Parametric Analysis, Circul_Ar, International Master II Level, University of Camerino, 2021.

Cogdell, Christina. Toward a Living Architecture? Complexism and Biology in Generative Design. University of Minnesota Press, 2019.

Darwin, Charles. The Origin of Species, by Means of Natural Selection or the Preservation of Favoured Races in the Struggle for Life. John Murray, 1859.

Fogel, Lawrence J. Artificial Intelligence Through Simulated Evolution. Biophysics and Cybernetic Science Symposium, 1966.

Frazer, John. An Evolutionary Architecture. Architectural Association Publications, 1995.

Holland, John H. Adaptation in Natural and Artificial Systems. University of Michigan Press, Ann Arbor, 1975.

Holland, John H. Complex Adaptive Systems. Daedalus, vol. 121, no. 1, 1992, pp. 17–30.

Jabi, Wassim. Parametric Design for Architecture. Laurence King Publishing, 2013.

Kolarevic, Branko (Ed.). Architecture in the Digital Age: Design and Manufacturing. Taylor & Francis, 2003.

Leach, Neil, and Philip F. Yuan (Eds.). Computational Design. Tongji University Press, 2018.

Negroponte, Nicholas. Toward a Theory of Architecture Machines. Journal of Architectural Education (1947–1974), vol. 23, no. 2, March 1969.

Pawlyn, Michael. Biomimicry in Architecture. RIBA Publishing, 2011.

Sabin, Jenny E., Peter Lloyd Jones, and Ferda Kolatan. LabStudio: Design Research Between Architecture and Biology. Routledge, 2017.

Schwefel, Hans-Paul. Evolution Strategies: A Comprehensive Introduction. Natural Computing, vol. 1, no. 1, 2002, pp. 3–52.

Tedeschi, Arturo. AAD Algorithms-Aided Design: Parametric Strategies Using Grasshopper. Le Penseur Publisher, 2014.

Terzidis, Kostas. Algorithmic Architecture. Architectural Press, 2006.

Turing, Alan M. Computing Machinery and Intelligence. Mind, vol. 59, no. 236, 1950, pp. 433–460.

Turrin, Michela, Peter von Buelow, and Rudi Stouffs (Eds.). Computational Design Modeling: Proceedings of the Design Modeling Symposium Berlin 2011. Springer, 2012.

List of Figures

List of Figures

List of Figures

PLUS

Work with us

info@parkassociati.com
www.parkassociati.com