

Genetic algorithms in architecture

HISTORY AND RELEVANCE

PEDRO DE AZAMBUJA VARELA

FAUP and ISCTE-IUL, Portugal # pedrovarela@rucativa.com

ABSTRACT:

This work has its goal in clarifying hypothesis in using search computational power in the form of genetic algorithms to find geometric solutions while designing architecture. We will start by roughly explaining the fundamentals of this type of algorithms and its mechanics and ensue by exploring some uses of this technology in architecture. A clear separation between the traditional self-builder and the architect as a profession, in their ways of designing, cannot be denied. Christopher Alexander has a clear view on the mechanics behind these processes, in which repetition over time drives perfection and self-awareness in error detection. Some years before Alexander, computation brought technological possibilities of brute force solution finding, only to be “evolved” into evolutionary strategies – as predicted by Turing. Bringing biology and genetics concepts into the fields of computation, Holland creates genetic algorithms, a simulation of the laws of Mendel and Darwin applied to solutions finding. Architecture design, in its pursuit of finding good project solutions, started using these computing strategies recently. Over the last years, the complexity of design levels in which these algorithms have been applied has increased steadily, giving us a glimpse of the future of computing architecture.

KEYWORDS:

Genetic Algorithms; Evolutionary; Bottom-up; Computation; History.

1. BOTTOM-UP DESIGN

The built world as we know it is composed of a multitude of forms. Forms as in shapes, and forms as in ways of forming shapes; final designs and processes. It is easy to distinguish between what may be called simple and complex civilizations (Alexander, 1964) through an analysis of these two realities. Designs of simpler civilizations tend to be the result of aggregation of parts in the process of their formation; on the other hand, complex civilizations produce complex master plans in which the smaller fractions are put inside somehow. The tidier and less wasteful this organization of fractions inside the whole is, the better the design is. On the other hand, a design feature that is easy to grasp within simple cultures is the organicity of its shapes which might be well contrasted with the shapes produced from the Industrial Revolution till the apogee of the International Style. These latter macro shapes have a close symbiotic relation to the micro shapes of their building materials: industrial standardized mass production lend a huxleyan equality between these building blocks, opposed to the spontaneous fabrication of a Mousgoum hut in which the labourer-contractor-client-user will pick its materials and get them together for that single-use.

These two approaches are very distant from each other, having antipodal characteristics: while in the Mousgoum hut all materials are used without waste, serialized pieces of modern societies have low efficiency when meeting odd edges or corners, for example. This problem and others are often addressed in what we tend to call good (or better) designs. The designer, after becoming aware of the geometric problems of his first approach, tries to tackle one and other situation, through specially designed pieces, or with general affine or homologic operations. This might work in some situations, until more parameters might come into play. Structural metric requirements, optics considerations, or other performance-driven requirements, might create an intricate network of correlations very hard to grasp all at once (Alexander, 1964).

More recently, a new approach in which we try to mix the adaptive power of time-spanning cultures with today's computing possibilities has surfaced. Instead of creating a design that, only after being finished proves to be efficient at these various levels, we may make these various levels of input be the agents of design creation, so that fitness (or its approximation) is guaranteed from the first stages of design. This process of building up a design from its constraints while unaware of its final shape is called bottom-up; as opposed to top-down, which is the process of choosing a final shape to only afterwards test it or improve it to meet the constraints.

In the last century, many tries at bottom-up design have been made, in what seems to be a design response to various discoveries in the fields of physics and biology. Gaudi used hanging ropes to find the very efficient catenary and use it in his designs. Frei Otto tested soap bubbles. Heinz Isler experimented with ice shells. All of these are examples of forms whose shape is driven by laws of physics, but they all have one feature in common: responding to forces in such a way that spends the least amount of energy.

However, there is an ensuing interesting thought by Roudavski: "While these structures might be optimal to their physical conditions, they are not characterized by fitness for any purpose, behavior or habitat. Their tectonic efficiency, while formally pure, (...) their transformations are dictated simply by the physical attributes of the components involved and not by any feedback from how well the form is adapted to function, because there is no function" (Roudavski, 2009).

Maybe this has driven these same architects to also study biological phenomena; Frei Otto analyzed spider webs, Gaudi favored natural tilings found in honeycombs. Though products of life and not of physics, the logic behind these forms is the same: the most efficient in order to gain fitness. Through the understanding of the mechanics behind the rules that drive these systems to their optimal solutions, we start to delve in its bottom-up logic. Or in other words, we get hold of the algorithm responsible for the solution. For a bottom-up approach to exist, there needs to be a generator algorithm. This set

of rules will create, through one or more iterations, a design - be it a shape, a collection of points, whatsoever. The resulting solution might - most likely will - be emergent. Emergency as in something that surfaces from the invisible into our vision, something that takes us by surprise: a solution we didn't think of. That happens because the bottom-up approach does not take in account our prejudice: it only works its way to follow our given rules.

2. GENETIC ALGORITHMS

From our very beginning we humans try to understand the world around us. As much as we have learned over millennia, knowledge about our universe and its mechanics never stops to develop. In fact, it grows exponentially bigger everyday. It is therefore no surprise, that scientists use every means available to use as a tool for investigation and grow knowledge. Arguably, the most revolutionary tool of this century is the computer (Mitchell, 1996). If we go back in time when computers started to work for us, we get an interesting sight of these pioneers thoughts: "By the time the world's first computer ran the first stored computer program, he [Turing] had already moved on and was proposing the notion of artificial intelligence" (Frazer, 1995). Before artificial intelligence (AI) was implemented in problem solving, other approaches directly inherited from mathematics were used in computer programs, such as root-finding algorithms, polynomial equations and such. But by mid 20th century Norbert Wiener argues that "the only significant difference between controlling anti-aircraft fire and biological systems was the degree of complexity" (Weinstock, 2004) - describing a problem of "predicting" things. Wiener is the father of modern cybernetics, defining it as "the scientific study of control and communication in the animal and the machine" (Wiener, 1948).

This idea of getting the computer to replicate our world soon got momentum, only possible because of the "power to compress evolutionary time and space so that results can be achieved more realistically in our life-times" (Frazer, 1995). Soon there were productive approaches to this idea of getting the computer to act as a fast paced world giving solutions to problems. In 1963 Rechenberg and Schwefel developed "evolution strategies" while working in the optimization of wing aerodynamics (Heitkötter & Beasley, 2001). In response to the weak results they were getting in experimental optimization through brute force strategies (such as Gauss-Seidel and gradients), they tried a different approach closer to their interests: cybernetics and bionics. As they experimented with changing multiple parallel variables (offspring mutation) at the same time, within a small range, results were positive, showing that the idea of replicating the multiplicities of many individuals analyzed by a same amount of minds was a successful approach (Heitkötter & Beasley, 2001). The overall idea was to mutate and select in each iteration, resulting in a randomly generated self-adapting system. In parallel, Fogel, Owens and Walsh developed "evolutionary programming," (Mitchell, 1996) with the release of the book "Artificial Intelligence Through Simulated Evolution" (Holland, 1975). This work includes Markov predicting in its processes, as it was a recognition that prediction is a keystone to intelligent behavior. Evolution of a part of a specific species is the strategy devised by Fogel, Owens and Walsh, in which each parent generates an offspring. This vertical lineage will be accessed and the best breeds are allowed to survive.

John Holland had a different approach in the University of Michigan. Although also using the computer, he was not trying to solve a specific problem, but trying to understand adaptation in nature (Mitchell, 1996) by replicating it. For this, he implements a system based in Darwinian concepts: "the "rules" of evolution are remarkably simple: species evolve by means of random variation (via mutation, recombination, and other operators), followed by natural selection in which the fittest tend to survive and reproduce" (Mitchell, 1996).

Together with the notion that "evolution is a massively parallel search method" (Mitchell, 1996), it is clear why the concept of evolution applied to the fast computer calculus was appealing. In a way, man would be able to play the role of Nature in the creation of highly fit solutions, bypassing millions of years of evolution refining.

the allele is the number contained in the variable. This genome - a single gene genome - will produce each one a square whose area will be evaluated and compared to that of a circle with fixed radius, by means of a division. The more similar these areas are, the closer to 1 will be the result for this division. This is our fitness value: a mathematical/logic way of evaluating solutions by means of an arithmetic subtraction of 1 minus value. This subtracted values are then ordered creating a sorted of chromosomes sorted by fitness. Selection now easily takes place, discarding the last n-chromosomes of the list. The remainder chromosomes would then be subject to Crossover if consisted of more than one gene. As it is not the case, there is a direct duplication of results to achieve a total population again, followed by a simple Mutation of the single gene through variation of the radius allele, and the process starts over.

In each iteration, or generation, a population of 50 chromosomes was generated (pun intended) by Galapagos, a genetic algorithm component built-in Grasshopper. After 50 generations, a result with an error margin of 0.0007% was achieved. Comparing to the result given by the equation fore-mentioned, this is a worse result but some considerations are worth noting: it is a remarkable result for someone who does not know anything about geometric relations; and the margin of error is really small for an engine which was not optimized at all.

Bottom-up approach, such as a genetic algorithm, doesn't give us a glimpse of the final solution. This solution turns out to be a surprise, emergent (Menges, 2010): "emergence is said to be the properties of a system that cannot be deduced from its components, something more than the sum of its parts" (Weinstock, 2004). This same emergency is found throughout nature, according to the idea that biological solutions, in the form of beings, have come to a phenotype which was not to be guessed in the beginning of the process, but a phenotype with its genes deeply rooted in the original genotype with the key genes.

Alfred Whitehead advocated that "process rather than substance was the fundamental constituent of the world, and that nature consists of patterns of activity interacting with each other" (Weinstock, 2004). This view of biological processes as an algorithmic solution to problems is even more stressed by Roudavski, who says that "biological organizations are different from the purely physical systems because they result from and participate in evolution. They have to be sufficiently flexible to remain open to modification by natural selection. Purely physical morphogenesis tends to produce inflexible structures such as crystals" (Roudavski, 2009). From this kind of input, genetic algorithms naturally surface as an appropriate paradigm in which solutions cross-breed in search of the fittest offspring.

But the opposite thinking is also true. The natural processes that allow all living things to strive for life is understood as cybernetics by Norbert Wiener; the same cybernetics associated with non deterministic prediction of the future but "merely the distribution of possible futures of a system." (Wiener, 1948). Futures only possible because of the adaptability of virtual machines, since they can be fed different input while giving an always fittest output.

"No one has yet invented a scale for unhappiness or discomfort or uneasiness, and it is therefore not possible to set up performance standards for them. Yet these misfits are among the most critical which occur in design problems. The importance of these non quantifiable variables is sometimes lost in the effort to be "scientific." (...) A design problem is not an optimization problem" (Alexander, 1964).

In the verge of using this bottom-up approach to create designs for architecture it is important to understand the sheer differences between a mathematical problem - squaring the circle - and a design problem. A design problem is composed of many problems (Alexander, 1964), some more easily quantifiable than others. This quantification has been the subject of investigation in the form of various disciplines in the realm of engineering, architecture or sociology. One of the core benefits of these disciplines is allowing for a non-human analysis system evaluate fitness. Some examples are finite element method (FEM) structural analysis, spatial syntax, shape grammars or building energy use

simulation software. Through these tools, and through other custom made algorithms depending on the project, we are getting nearer to the possibility of transforming (part of) a design problem into an optimization problem. What doesn't seem negligible is that one can't go totally bottom-up. In the very definition of a design problem, there is a top-down approach, and even in the choice of solution analysis, or choice of fitness values, top-down is always present. Each design problem must be looked upon as a contained universe, in which this universe's laws correspond to our top-down contributions to the circumvention of our problem.

3. ARCHITECTURE DESIGN DRIVEN BY GENETIC ALGORITHMS

3.1 BEGINNINGS - JOHN FRAZER

Evolutionary architecture has its bases in the work of John Frazer. Coming from pure electronics, Frazer is well acquainted with the inner mechanics of calculus and logic.

The Reptile structural system is a seminal work started in 1966 in using evolutionary algorithms to create an architecture work. The code was created as to drive the formative process instead of explain the form. Two folded-plate structural elements were arranged in combinations whose details were chosen by the algorithm. "The seed was written in compressed form so that the complete details of the location of unit, its orientation and type could be contained in just one word of Titan machine-code, thus establishing the genetic coding analogy" (Frazer, 1995).

Another experiment using genetic algorithms is with Tuscan columns. He sets himself the goal of finding emergent columns such as they fit James Gibbs 18th century system of proportion which characterizes the Tuscan order (Frazer, 1995). As a first example, it is worth noting the importance of parametrization while designing within a numeric environment. The genetic algorithm always tends to create a genotype whose fitness is closest to the value intended. For this value to be reached, other numbers must come into play; these numbers may be attained by genotype analysis, or directly by the parameters that inform the genotype: either way, the algorithm must have a way to inform new genotypes. Genotypes must be coded into strings, and the way to transliterate string into graphic shapes is through parameterization.

A study in 1993 infers form from a sole agent: solar radiation. The model starts out as a circular form while elements of the surface have the ability to grow or move in order to protect the surface from solar radiation. After this first try, other criteria ensued, all of them to be put to the evolution test at the same time. This example is quite explanatory as how a bottom-up process may replace a top-down one, in which, step by step, more complexity and deeper questions are addressed. It is also important to note the parallel nature of the process, as opposed to a serial one in a top-down process. As Christopher Alexander puts it, a design problem is composed of many sub-problems, which have to be addressed one at a time. Another work by Frazer worth noting is the Universal Interactor, in which three-dimensional cellular automata are controlled by genetic algorithms.

3.2 GENETIC EXPLORATION

Celestino Soddu pioneered the use of evolutionary techniques to explore different emerging phenotypes. His work in the generation of medieval cities (1988) is of seminal interest as a stress is put in the power of multiple solutions, all obeying the same generator rules (Soddu, 1998). Paul Coates uses genetic algorithms to explore spatial configurations and designs, while investigating the possibilities of generative modeling as a teaching aid (Coates 1993).

3.3 MATURED STRUCTURE - SHEA, SASAKI

In the end of the 20th century we see new applications of GAs in the formulation of architecture - or in other words, architecture players.

Mutsuro Sasaki, a structural engineer, through the “organic inspiration of seaweed transformed digitally into structure” (Meredith & Sasaki, 2008) which he experimented in the 1998 Sendai Mediatheque project by Toyo Ito, had him turn to more organic solutions: organic in form, but also organic in process. Since the year 2000 that Sasaki had been researching at the university Sensitivity Analysis which is an optimization method of shape analysis for the generation of free-curved shells. Strains in structure are valued by the Sensitivity Coefficient which, in turn, can be used as a fitness value. This method was used in the Nation Grand Theatre competition in Beijing and the Kitagata Community Centre, both projects by Arata Isozaki.

According to the author, 10-15 minutes of computer calculation were all it took to optimize each of these structures. In 2003 a new paradigm in this structural engineer workflow emerged: instead of analyzing the structural efficiency of a shape produced by the architect, he himself was to produce a shape which would, from the first moment, be designed with efficiency in mind. In a wildly organic manner, the banyan tree is referenced as a main source of in-spiration in this design, whose form is completely performative. Carrying out extensive matrix operations using the three dimensional Finite Element Analysis (FEM), the algorithm starts to evolve into a static congruent shape, giving an optimal solution as an emergent project: both arches and catenaries were born out of this process in what looks as a very plant-like structure.

Still in the field of structural design, Kristina Shea's end of century EifForm is a turning point software in stochastic optimization algorithmic design. It is one of the first examples of effective use of GA's in the design of different projects with different inputs. Being a generalized approach, the process cannot start without first defining key variables. These are of geometric, performance and generative nature. Geometric constraints are given as initial configurations, or bounds for the generated design. Performance here has to do specifically with mechanical and statics engineering, such as materials, loads, and other forces. Generative input is characterized by a sort of shape grammar with numerical and geometrical constraints, so that the algorithm uses a bounded vocabulary of n-type words and m-total words. In each iteration, the overall structure is analyzed and reformulated using a simulated annealing algorithm together with a shape grammar.

A paradigmatic example of the application of EifForm is the design of the roof truss for the aquatic area in Carnegie Mellon University. As we can see in the pictures, the solutions devised by EifForm are quite emergent, easily distinguishable in its form to the top-down solutions generated by humans.

3.4 A SOFTWARE FOR STARTING ARCHITECTURE

Following precedent advances in shape annealing - Cagan and Mitchell, preceding EifForm - and the possibilities devised by the EifForm project, Chouchoulas creates Shape Evolution: a design tool “for supporting the initial stages of architectural design” (Chouchoulas, 2003). The main aspect of this work resides in the close link between genetic algorithms and shape grammars, putting the architect as a direct initiator as he can introduce his own shapes as generators, which, in their turn, result in own genotypes. The study-case presented by Chouchoulas is the design of an apartment building in which individual dwellings are associated with a circulation block. Shape grammars are used to devise possible associations of these elements, while the genetic algorithm generates multiple solution that conform to the conditions specified by the architect: in this case the ratio between circulation and dwelling blocks, or balconies views regulated the process.

3.5 DIVERSITY THROUGH OPTIMIZATION

Somewhat on the heels of structure optimization techniques used by Mutsuro Sasaki and others, Luisa Caldas works in a tool for the analysis of light and heat in buildings, creating meaningful suggestions in the process. Through an advanced thermal and light simulation tool - DOE 2.1 E -, the project is analyzed and, through GA, solutions are iterated until the fitness value comes to a rest. This procedure is well exemplified in the analysis of one of the blocks of FAUP school of architecture at Porto,

showing key differences or similarities in window sizes and placements between Álvaro Siza's project and the thermal/light optimized solution by GENE_ARCH.

An interesting offspring on the GENE_ARCH investigation is a recent work by Caldas in which this analysis tool is entwined with Marrakech edina's shape grammar (Duarte & Rocha & Ducla-Soares & Caldas, 2006) giving rise to multiple emergent solutions that, while conforming to a shape grammar that identifies a culture and way of living, also optimizes its energy performance. This is clearly a case of evolutionary architecture, in such as keys aspects of the project are developed by the computer algorithm, conforming to pre-established laws - these are of physical nature, and of culture pre-existence.

3.6 GEOMETRIC SOLUTION

How can we layout 100 randomly orientated lines in a 2d space in such a way that voids created between them are neither larger neither smaller than a certain range, and the angles produced in intersections fall within certain values? What starts as appearing to be an herculean task turns out to be a perfect target for a solver with an evolutionary engine. This study carried out by Ludger Hovestadt's team in cooperation with Herzog & de Meuron in 2003 translated the iconic stadium's beams into lines which by turn were read by the algorithm as genes. Through the evolutionary process, solutions were mutated, selected, multiplied, mutated(...), and after 600 iterations the solution worked flawlessly within the established limits. "You could say that we had 'shaken' the complicated roof construction for so long, letting the individual members fall where they may, that we finally arrived at the correct solution" (Meredith & Sasaki. 2008).

A "forest of columns" (Meredith & Sasaki. 2008) has in its own words what it takes to resemble a natural process of evolution. This forest, one of the key concepts in this 2003 project by KCAP, was to be materialized as supporting columns of a big continuous slab. These columns ought to have its inclination, diameter and placement decided in a random fashion so as to resemble the organicity of a forest. In order to have a thorough budget without unnecessary over-structured concrete, it was necessary for all these variables to play well together. If the distance between grows, so should its diameter grow; if the diameter exceeds a certain value, two columns should exist instead of one. The problem of "too many balls up in the air at the same time" (Hovestadt, 2010) was addressed through the implementation of a genetic algorithm by design to production. Moreover, a software with GUI was created so that the architects could play with different inputs and, in little time and effort - the effort was upstream, in the programming - a pleasing solution was chosen. The final project was created by single-agents in a bottom-up process defined by simple rules in which objects self-organizes themselves.

3.7 PROJECT AS A WHOLE

Following these advances, it is not difficult to imagine the evolutionary role of algorithms getting more and more entwined with architecture projects themselves. Ludger Hovestadt gives us account of Hardturm - a typical housing project commissioned in 2007 in which his team put computers to find optimal plan solutions as to accommodate the maximum number of habitable area while managing the highest levels of quality - comfort and salubrity. This approach combines two works, in which Hovestadt is also involved: Kaisersrot and Architectural Google. Kaisersrot is a software which employs stochastic methods in evolutionary algorithms for finding optimal geometric solutions. Its first uses were urban, such as in dividing a terrain in plots and streets while keeping a high level of fitness, or creating an urban project taking in account more complex variables such as historicity or multiple levels of circulation - Schuytgraaf in 2001, and later Heerhugowaard in 2005.

Architectural Google is a project coordinated with Matthias Castorph which aims to build a searchable database of floor plans. "it is not necessary to re-invent successful ground plans, since they already exist in sufficient numbers and with sufficient variety" (Hovestadt, 2010). In this manner, a library of well-conceived workable ground plans, and dwelling layouts was produced so that through parametrization

this plans could be applied in a variety of inputs. The overall result is a evolutionary project of a 300 apartment block, in which the floor plan of a single apartment is an input in itself. Worth noting the facade was also subject of parametrization and subsequent emergent variation.

Monte Rosa is the second-highest mountain in western Europe. A mountain hut shelters climbers, with a capacity ranging from 25 beds in 1895 to 160 in 1984. In 2006 a new 120 bed building project was initiated to create an ex-tremely optimized solution at multiple levels. As put by Hovestadt himself: "this project is a consolidation of all the previous projects, technologies, and, above all, the various approaches to architectural problem-solving presented previously". Many levels of fitness were played together in a real algorithmic mix, in which the wind and snow load resistance, facade inclination for solar exposure, bed numbers and most importantly the total mass of building elements were driven either to maximum, minimum or optimal goals. Optimization governed the emergence of a final shape, cut like a diamond, a performative diamond: "the elements (...) can break free of the grid and organize them-selves freely. Their size, their proportion, or the number of their neighbors can be continually renegotiated without losing the integrity of the overall system." (Hovestadt, 2010). What was projected was not a building but a whole process in which the key elements dialogue with one another, in what tended to be and resulted in a meaningful and symbiotic conversation.

"This optimization (...) carried out using an evolutionary algorithm that worked through an arbitrary number of iterations in order to reach an optimized solution within the defined framework parameters" (Hovestadt, 2010) resulted in a weight reduction of 40%, which means a lot when costs are calculated for helicopter transporting. The whole planning and prefabrication resulted in less than 60 hours for the construction on site. This kind of smooth productivity can only be attained with a finely tuned sequence of digital processes: "A gulf opens up between design and reality when the gaps in the digital chain are not fully plugged" (Hovestadt, 2010).

4. CONCLUSION

Other works (Chouchoulas, 2003) are worth investigating, as it was not adequate to fill this paper with all subjects encountered. It seems interesting, however, to note that throughout this brief overview of the use of evolutionary algorithms and more precisely genetic algorithms, the use of these techniques has been applied each time to more specific problems and to a more real world. An example not shown above is Makoto Sei Watanabe's Iidibashi Station project in Tokyo (2000), which he claims to be the first fully evolutionary architecture project. Using these techniques for design inside design is also valid, as one can see in Milos Dimcic's work on subdividing surfaces highly fit.

The key role played by fitness must be looked upon carefully; after all, this is the measure of purpose that the algorithm will self-govern in order to take design decisions yielding a final result. On the other hand, fitness itself is totally dependent of the algorithm which takes place before evaluation. This algorithm is the designers tool while using genetic algorithms. It is a tool crafted for each design, being a self-contained design in itself. As any tool, it must be used in coherence with the overall design, having less or more impact in the big picture. Brunelleschi designed Santo Spirito church in Florence using a strict shape grammar as a tool, to the point of causing conflict with its client. The church plan consisted of geometric module which would cause a direct consequence in the outer facade. The main nave was composed of two of these modules, creating two events in the main facade for which two symmetric doors were proposed, instead of one central door as expected by the client. Today we can visit the church and we will note the unique central door on the facade. Whether this is a sign of a design fail or not is beyond the scope of this work, but it is worth noting that the tool is not the design. We face an extremely flexible tool whose possibilities of finding answers through evolution merely depends on asking the right question.

REFERENCES

- Meredith, Michael, and Mutsuro Sasaki. *From control to design: parametric/algorithmic architecture. Actar-D, 2008.*
- Heitkötter, Jörg and Beasley, David, eds. (2001) "The Hitch-Hiker's Guide to Evolutionary Computation: A list of Frequently Asked Questions (FAQ)"
- Roudavski, Stanislav. "Towards morphogenesis in architecture." *International Journal of Architectural Computing* 7.3 (2009): 368.
- Alexander, C., *Notes on the Synthesis of Form*, 1964
- Melanie, Mitchell. "An introduction to genetic algorithms." Cambridge, Massachusetts London, England, Fifth printing (1999).
- Frazer, John. *An evolutionary architecture.* (1995).
- Hovestadt, Ludger, *Beyond the Grid: Architecture and Information Technology Applications of Digital Architectonic*, Birkhäuser, 2010
- de Boer, M., F. Fracchia, and Przemyslaw Prusinkiewicz. "A model for cellular development in morphogenetic fields." (1992): 351-370.
- Watson J.D. and Crick F.H.C. (1953). "A Structure for Deoxyribose Nucleic Acid" (PDF). *Nature* 171 (4356): 737-738.
- Wiener, Norbert (1948). *Cybernetics, or Communication and Control in the Animal and the Machine.* Cambridge: MIT Press.
- Weinstock, Michael (2004). *Morphogenesis and the mathematics of emergence.* *Architectural design* 3: 10-17.
- Hensel, Michael, Achim Menges, and Michael Weinstock (2004). *Emergence: morphogenetic design strategies.* Chichester: Wiley-Academy,
- Dimic, Milos. "Structural optimization of grid shells based on genetic algorithms." (2011).
- Chouchoulas, Orestes (2003). *Shape evolution: an algorithmic method for conceptual architectural design combining shape grammars and genetic algorithms.* Diss. University of Bath.
- Holland, John H. (1975). "Adaptation in natural and artificial systems, University of Michigan press." Ann Arbor, MI 1.97: 5
- Soddu, Celestino & Colabella, Enrica, (1992) "Il progetto ambientale di morfogenesi: il DNA dell'artificiale", Esculapio Publisher
- Coates, Paul (2010). *Programming architecture.* Routledge.