



HAL
open science

Evolutionary Tools and the Practice of Architecture: from Appropriated Typology to Becoming the Black Boxes of CAAD

Nadja Gaudillière

► **To cite this version:**

Nadja Gaudillière. Evolutionary Tools and the Practice of Architecture: from Appropriated Typology to Becoming the Black Boxes of CAAD. BIODIG 2020 - 4th International Conference for Biodigital Architecture & Genetics, Alberto T. Estevez, Jun 2020, Barcelona, Spain. hal-04588642

HAL Id: hal-04588642

<https://hal.science/hal-04588642>

Submitted on 27 May 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/342658352>

Evolutionary Tools and the Practice of Architecture: from Appropriated Typology to Becoming the Black Boxes of CAAD

Conference Paper · June 2020

CITATIONS

3

READS

223

1 author:



[Nadja Gaudillière-Jami](#)

Technische Universität Darmstadt

19 PUBLICATIONS 943 CITATIONS

SEE PROFILE

Evolutionary Tools and the Practice of Architecture: from Appropriated Typology to Becoming the Black Boxes of CAAD

Nadja Gaudillière¹

¹ *Laboratoire GSA, ENSA Paris-Malaquais, 14 rue Bonaparte, 75006 Paris, France*

¹ *nadja.gaudilliere@paris-malaquais.archi.fr*

Abstract. The present research proposes a comprehensive methodology of investigation of algorithmic typologies currently in use in the computational field in architecture, focusing on three elements : practitioners, tools and projects. The methodology has been applied to evolutionary design tools in order to investigate the technical and epistemological optimization bias they entail. The research shows the existence of two approaches to handling the tension between formalization required by algorithms and tacit knowledge required by architectural practice. Whereas one of the approaches displays an intertwinement of explicit and tacit, of qualitative and quantitative, the other prioritizes a rational, performance-based approach. Given the conditions of development of evolutionary tools shown, this latter conception of architectural practice has spread while recent tools enabled a democratization of the resort to evolutionary techniques. By transforming evolutionary algorithms into ready-made tools, or black boxes of CAAD, this process of development prevents the implementation of tacit knowledge within them, potentially impoverishing architectural design.

Keywords. Evolutionary tools; tacit knowledge; socio-technical network; technical biases; optimization; architectural practice.

Introduction

Since the rise of digital tools, and in particular of algorithmic design tools relying on programming skills, a tension has established between the formalization required by the use of programming languages and the expertise and mobilisation of tacit knowledge architectural practice relies on. In opposition to explicit knowledge, tacit forms of knowledge cannot be formalized : their transmission by written or oral communication entails a partial loss and they can only be acquired by the experience of practice (Collins 2010)¹. Practices such as architecture rely on an equilibrium between the resort to tacit knowledge and the resort to explicit knowledge provided by a corpus of scientific methodologies. This specificity renders the automation of such practices in computational models a difficult but rich exercise. Studies on the computerization of medical practices have nevertheless shown that refusing to acknowledge the tacit dimension of such disciplines leads to conflictual relations to digital technologies for the practitioners (Berg 1997). The tension between computational formalization and tacit knowledge therefore constitutes an essential node of our understanding of the use of digital tools for architectural design. In the last six decades, several means of resolving this tension have been investigated in the digital experimentations led in architecture, both in theory and in practice.

Evolutionary tools in particular have been migrating during the history of their use in architecture from one mode of appropriation to another and therefore constitute an

¹ Sometimes referred to as "empirical decision making" in the field of computational, it can also be described in other fields as intuition or creativity - although each of these terms has specificities, they refer to the same general notion.

emblematic model of how practitioners tackle this tension. These typologies of algorithms enclose a variety of computational methods designed to mimic natural phenomena such as plant growth, fire spread or darwinian selection. As many of these natural phenomena involve reaching an optimal state in a given environment, they have been identified as potent models of optimization and are nowadays mainly used to that end. While scarcely observed in existing studies, biases inherent to such design tools are instrumental to be understood for us to grasp the influence of the digital paradigm on architecture. Evolutionary tools, in particular recent ones, hold an optimization bias in their very structure, a technical specificity that has been accompanied by an epistemological bias entertaining the predominance of a performance-based understanding of architectural practice. By studying the historical development throughout the past thirty years of evolutionary tools in architectural practice, as well as the structure of their interfaces, the present papers analyze the technical and epistemological optimization bias they encompass. It thus offers an illustration of the two main approaches to tackling the tension between formalization and tacit knowledge, through the investigation of the use of evolutionary tools for architectural design.

Socio-technical networks of computation in architecture

The socio-technical network of evolutionary tools

The computational field in architecture has been a small academic community for several decades, and before the recent mass democratization of digital tools, experimentations on their use for architectural design was conducted in a handful of research groups. These research groups and the tools and the projects they have developed constitute what Latour (1991) refers to as a socio-technical network. This core network further expanded over the years beyond the initial microcosm and towards various players of the AEC industry. This socio-technical network of computation in architecture yet remains to be described as such.

In existing literature, some aspects of this sociotechnical network have however been studied. In particular, the work of Mario Carpo (2011, 2018) must be mentioned as one of the major historical and global overview of the computational movements in architecture. It must nevertheless be underlined that in recent works, in particular on optimization tools, parts of the dynamics that have led to the almost systematic recourse to these tools are not examined, lacking somewhat critical approaches to these recent. If numerous archaeologies of the digital in architecture already exist as well, such as (Lynn 2013), they consist in their majority of an assembly of emblematic projects short presentations and texts by the architects and designers of the computational field. Only a few studies of the development of specific digital tools exist, such as a history of Form*Z (Serraino 2002) or a study of the development of digital tools used in Frank Gehry's practice (Smith 2017), focusing mainly on the period of the 1990s and early 2000s and on the development of 3D modelling.

While algorithmic design and the computational field form a global socio-technical network, tracking the development of major algorithmic typologies in use reveals local chains of connections. The present research focuses on the network formed by evolutionary tools, whose core is depicted in figure 1. The lines indicate connections at large between institutions, practitioners and evolutionary tools. Light blue circles and the constellation they are at the center of represent clusters of evolutionary algorithms development and use. These clusters include the early research at the AA, both between 1989-1996 at Diploma

Unit 11 while John Frazer led it and in the early 2000s with the work of Michael Hensel, Michael Weinstock and Achim Menges. It also includes the first applications for built projects by Bollinger + Grohmann, in association with a few prominent architecture practices from 2007 on, as well as major software clusters such as McNeel or Autodesk. Compared to other sub-networks such as the multi-agent systems network (Gaudillière 2020), constituted of academics or small-scale offices and open-source software, this network involves a greater number of larger players of the AEC industry as well as of the software industry. This greater variety of players in the clusters implies a greater diversity in users and in uses, to be assessed in the present study. Furthermore, while the multi-agent systems network features few tools travelling from one cluster to another, the evolutionary tools system features few clusters generating several tools each.

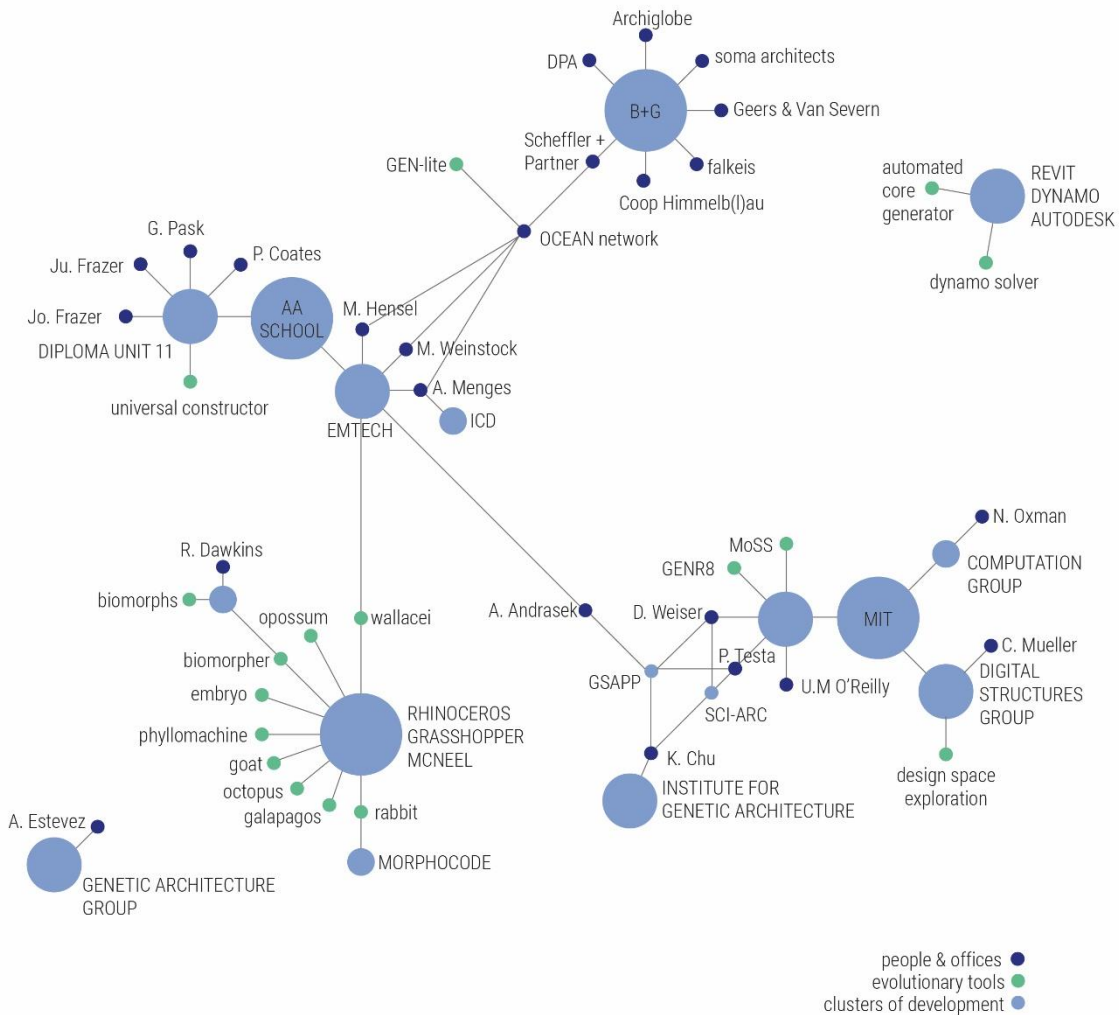


Figure 1
Clusters of evolutionary tool development in computational architecture

Methodology

The present research has been conducted on two algorithms families: growth systems such as L-systems and genetic algorithms, following four steps. First, in order to identify the clusters, AD issues and the Cumincad ACADIA database have been searched² to establish a list of projects and practitioners resorting to evolutionary tools, with a focus on the use of scripting to produce architectural objects or information relevant to the production of an architectural object. The second step focuses on the definition of a taxonomy of users, by looking at the practitioners identified and assessing their education (architecture, engineering, computer science or else), professional environment (academic, practice or both), programming skills (ability to use a ready-made algorithm, to develop a custom algorithm, to develop a tool). Third, projects³ resorting to evolutionary tools have been assessed examining four areas: algorithmic tools/method, computational set-up, organizational chart and architectural design⁴, in order to determine which technical aim was favored when using evolutionary algorithms (formal mimicry or optimization). Finally, released evolutionary design tools have been studied, focusing on the conditions of their development and on how their interface enables the manipulation of the algorithmic typologies they are designed for.

Evolutionary tools in architecture

Evolutionary tools: from biologists to architects

First developed by biologists, geneticists and biophysicists in the 1950s and 1960s - Nils Aall Barricelli (1954), Alex Fraser (1957) or John Holland (1975) can be mentioned as pioneers - simulations of natural selection soon became a full-fledged field of study, meeting point of biology and computer science. Evolutionary tools, genetic algorithms in particular, were then adopted by engineers to solve complex optimization problems in a variety of applications such as encrypting data, designing motor parts, ranking players, etc. In many domains, genetic algorithms have become popular optimization tools, due to their ability to take multiple criteria into account and to introduce mutations in order to avoid local minima or maxima, thus enhancing their performance. Furthermore, their ability to explore a vast design space made genetic algorithms prized not only by engineers, but also by architects who would in the 1990s increasingly appropriate them, alongside morphogenetic growth systems also popular for their formal potential. Historically though, exploring the latter has first been predominant among architectural experimentation, while genetic algorithms would then gradually take a hegemonic position.

² Following keywords were used when searching the database : “evolutionary tools”, “growth system”, “genetic algorithm”, “morphogenetic”.

³ Both paper / research projects and built projects indiscriminately

⁴ Data has been obtained by conducting interviews with practitioners and consulting projects drawings, programming files and publications.

More broadly, what many architects earlier on referred to as morphogenetic processes⁵ enclose a large variety of projects. Despite the vocabulary used, not all of them resort to what is nowadays known as evolutionary tools. Those which resort to evolutionary tools employ various typologies, or families, of algorithms. Algorithmic typologies are identified by the rules of formalization they rely on, different typologies having different programming structures and therefore different types of mandatory inputs. In the case of evolutionary tools, observing the typologies' characteristics allows for an analysis of how the technical constraints linked to the optimization bias are implemented. Two are studied in the present paper : L-systems and genetic algorithms. L-systems, or branching systems, are mathematical structures dating back to the beginning of the XXth century, developed by several mathematician until their current form, proposed by Lindenmayer in 1968 (Prusinkiewicz & Lindenmayer 1990). Used in particular to simulate plant growth, their rule typology includes : a) the initiator shape, b) the generator shape, c) seed placement, d) the branching rules (angle, length, number), e) the number of iterations (figure 2.a). As for genetic algorithms, despite various kinds of computation methods exist, they are all based on the same principle of darwinian selection (Calixto & Celani 2014). Populations of individuals are generated and evaluated, and the better performing individuals' characteristics are retained to develop a new generation, a process repeated until a series of individuals as performing as possible are obtained. The rule typology for genetic algorithms includes a) selecting the computation method, b) defining the gene pool (characteristics varying in each individual), c) defining the fitness function (the criterias on which each individual is evaluated), d) selecting an individual amongst the last, most performing generation (figure 2.b).

Taxonomy of users

The taxonomy of users developed in the present research results in three major categories, presented in table 1, also summing up the characteristics of each user category. First, two categories of skilled practitioners, characterized by their high programming competences, are identified : in architecture - with early adopters such as Karl Chu or John Frazer, but also a few contemporary users - and in engineering-based practice, with both academics such as the Digital Structures research group and professional practitioners such as Bollinger + Grohmann or Ramboll. A third group gathers mass users of the contemporary period, having accessed evolutionary algorithms after the release of tools with robust, appropriable interfaces. While the two first groups reflect the clusters of the socio-technical network shown previously, this third group embodies the turning point of evolutionary tools exiting the core network and expanding into common practice. Finally, a fourth group is also highlighted: while software developers only produce little research on the use of these programs for architectural design, they remain an important player given the key role their tools play in the democratization of evolutionary algorithms.

⁵ As can be seen in several AD issues such as *Emergence: Morphogenetic Design Strategies* (2004) or *Techniques and Technologies in Morphogenetic Design* (2006).

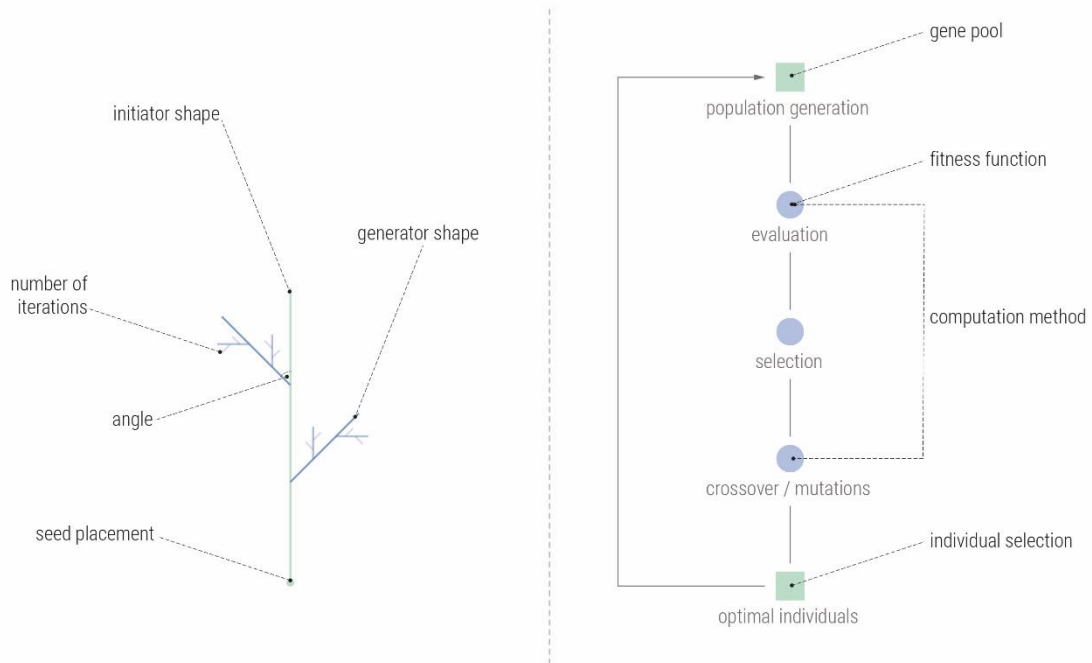


Figure 2
 a. L-system typology rules; b. Genetic algorithm typology rules.

	01 - skilled practitioners (arch.)	02 - skilled practitioners (eng.)	03 - mass users	04 - software developers
education	architecture (for many also a technical background in engineering / computer science)	engineering	architecture	computer science architecture
professional environment	academic (occasional projects as professional offices structures)	academic professional	academic professional	professional
programming skills	custom algorithms (also tool development but with little spread beyond their own institutional network)	tool development custom algorithms	ready-made algorithms	tool development
use of evolutionary tools	formal research optimization (mainly paper projects and academic research) (regular collaborations with computer scientists)	optimization (including for built projects in collaboration with large architecture offices)	formal research optimization (predominance of optimization)	/

Table 1
 Taxonomy of users

Appropriating evolutionary tools in the practice of architecture

Mastering programming tools, successfully implementing the tacit

Among the established corpus of those mentioning morphogenetic processes at large, identifying the projects placing evolutionary tools at their core is necessary. Those drawing inspiration from natural growth phenomena (Roussel 2017) must be differentiated from those using programming tools simulating such phenomena - even though some practitioners have mobilized the two together. Whether the evolutionary tool resorted to is the typology at the forefront of the design process or not must also be determined, since evolutionary tools, in particular genetic algorithms, are often combined with other algorithmic typologies. Despite the large corpus of projects mentioning evolutionary tools, only few match those two criterias and attest of a complete architectural appropriation. Among those, some resort to evolutionary tools mainly for formal explorations, such as the Tote (Serie Architects, 2009) or the New Czech National Library (Scheffler + Partner, 2006), other push further the entanglement of formal, structural and computational requirements - such as the Seroussi Pavilion (EZCT, 2006) or the Jyvaskyla Art and Music Center (OCEAN, 2004). The practitioners, by relying either on collaboration with computer scientists or on a solid knowledge of programming languages, enable themselves to design complex, custom algorithms, and the resort to evolutionary tools provides in these case the evidence of the practice of scripting as a craft. Beyond the combination of requirements, it is the transformation of tacit knowledge on spatial design directly into the computational typologies rules rather than into visual representations that is at play, attesting of an appropriation of evolutionary tools as any other technical tool would be appropriated into a classical architectural practice.

Morphogenetic liberties and the heritage of cybernetics

Discourse on the use of evolutionary tools in architecture is illustrated by a diversity of written explanations of the benefits of resorting to such techniques - a discourse that has been strongly criticized (Cogdell 2019). These critics underline the inaccuracy of the use made of evolution simulations, consequence of the absence of understanding in the architectural community of the details of biological phenomena studied and of the peculiarities of simulation models adopted. While the lack of precision on technical, computational and biological aspects in the vocabulary generally used by architects to discuss morphogenetic processes in their projects is indeed questionable, liberties taken with what evolutionary tools stand for in the biology and computer science communities is demonstrating precisely how the tension between tacit knowledge and formalization has been tackled through appropriation. Acknowledging this altogether classical dimension of part of the architectural production mobilizing digital tools means acknowledging the role played by tacit in architectural practice, which cannot - yet - be formalized and performed by a computer. In the early days of the computational field, practitioners such as Nicholas Negroponte (1969) or Nigel Cross (2006), and the cybernetic movement at large, have confronted this issue of automating the tacit. While a form of inheritance of these reflections is tangible in the practices of the contemporary computational field, the theoretical position assumed by these early researchers is nevertheless much less present

in discussions nowadays⁶. In particular for evolutionary algorithms, it is a line of thought that has disappeared while the use of tools progressed, making way for a preponderance of the notion of rationalisation.

Becoming black boxes of CAAD

Interfaces for democratization

In the past ten years, the computational field has grown through the democratization of its tools, including evolutionary algorithms, seeing both many new users and a multitude of dedicated software and plugins appear. Evolutionary tools have since the seminal work of Richard Dawkins (1986) and his Biomorphs evolved into both professionalized resources such as the Evolver software and open-source plugins such as the ones available on Grasshopper. Figure 3 displays a chronology of appearance of those various resources, all having in common the typological rules described previously. These various tools also feature another common trait: the presence of a user interface, contrariwise to custom algorithm developed in the first period of use of evolutionary processes in architecture. Those interfaces and their ease of use have been instrumental to the spread of those methods, despite the crucial simplification they entail. This simplification is stressed by many of the tool developers as necessary for a majority of architects to seize these potent design tools, sometimes at the expense of enabling a comprehensive apprehension of their structure, as can be seen in the description of some of those tools : “*the aim is for users (of all degrees of expertise) to better understand their evolutionary simulations, gain a thorough understanding of the outputted numeric values, and seamlessly (...); all within one user interface.*” (Wallacei plugin description), “*helping to explore the wide combinatorial space of parametric models without always knowing where you are headed.*” (Biomorpher plugin description). The result of this approach is, as expected, an increase in the number of users⁷, but also a shift in user skills : while earlier users were skilled both in computer science and in their own domain (here architecture), mass users are untrained in scripting while maintaining their own field of competence. In parallel, a shift in the background of evolutionary tools developers is also witnessed, with a large predominance in professionals stemming from engineering, AEC and software industry (table 1).

The consequence of both the reduction of complexity entailed by a more easily grasped interface and of the growing distance between the developer and the user sets of skills results in a thickening of the interface of evolutionary design tools. The thickness of an interface is a notion developed by Anthony Masure (2019), transcribing the distance between what a user can perceive and manipulate of a program through its interface and the actual structure of the program. Figure 4 displays a comparison of this distance in the case of a custom evolutionary algorithm and in the case of a ready-made evolutionary tool⁸,

⁶ Beyond scarce publications and the almost inevitable mention of “empirical decision-making” in paper introductions.

⁷ While the number of users of evolutionary tools has been in the present research be estimated to 35 000, significantly more than the few hundreds of the core network, the potential pool of users gathered by McNeel and Autodesk could reach up to 2 millions.

⁸ For the custom algorithm, the 2007 Seroussi Pavilion algorithm by EZCT (Guenoun 2007) was taken as example, and for the ready-made tool, Galapagos was selected.

by showing the layers of structure of each program. The specific rules of the typology and where to access them in this structure are also shown. A common goal was fixed for the two algorithms in order to compare the ease of access to each rule of the typology: performing an optimization in order to obtain a light repartition as uniform as possible in a given space. While the custom algorithm contains five straight layers, three programming languages and twelve software⁹, with the selection of rules regrouped in two layers, the ready-made tool contains seven layers including three double-layers, six programming language and three software, with the selection of rules scattered across five layers. It is furthermore interesting to note that in most Grasshopper plugins, individual selection is a sub-layer of the first interface, as pictured here, whereas in most other, individual selection is the first layer, before accessing fitness function and individual selection.

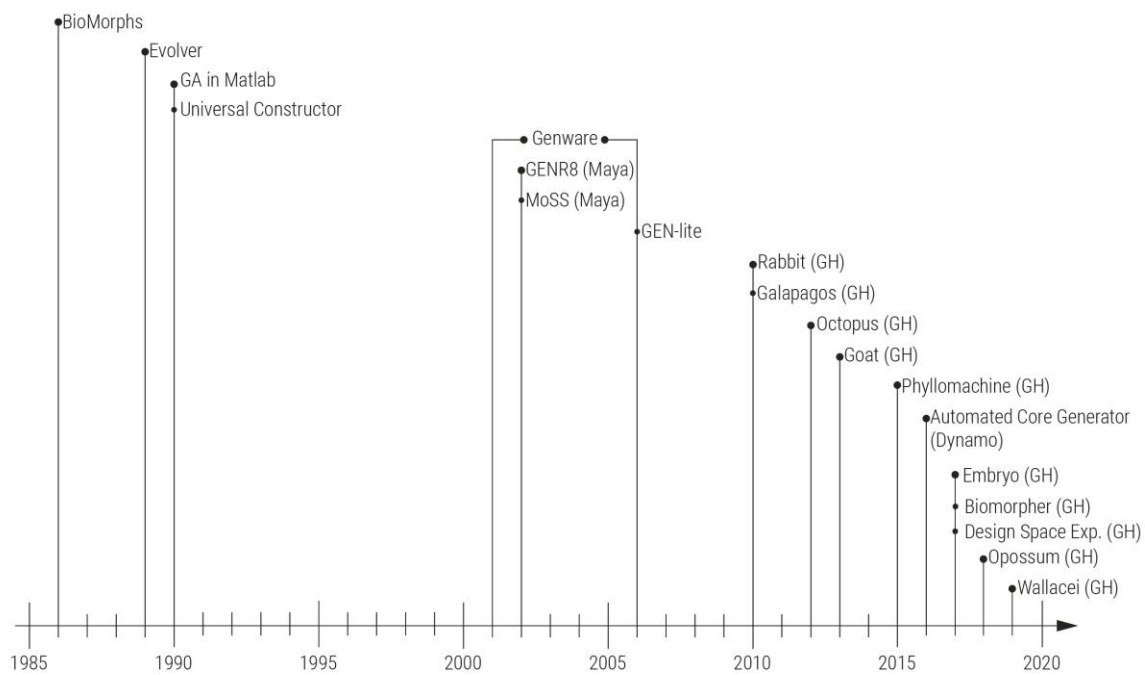


Figure 3
Chronology of appearance of evolutionary tools for CAAD

While crucial to democratization, these characteristics lead to a lack of taking into consideration the possibility to implement the tacit knowledge characterizing the skill set of architecture users and to a reduction of the flexibility of evolutionary tools. This also fuels an enduring lack of computational skills of untrained architects, weakening here again the possibility of appropriation through tacit knowledge implementation. Although it is also the consequence of a long-term diminishing foothold of architects into sciences for the built environment, from the apparition of the modern architect and engineer profession to

⁹ It is nevertheless an algorithm dating back to 2007, and a more recent custom algorithm might be able to reduce this large number of softwares.

contemporary architects (Picon 1989, Carpo 2011), the current structure of evolutionary tools further sustains it, as well as promoting a performance based approach of design.

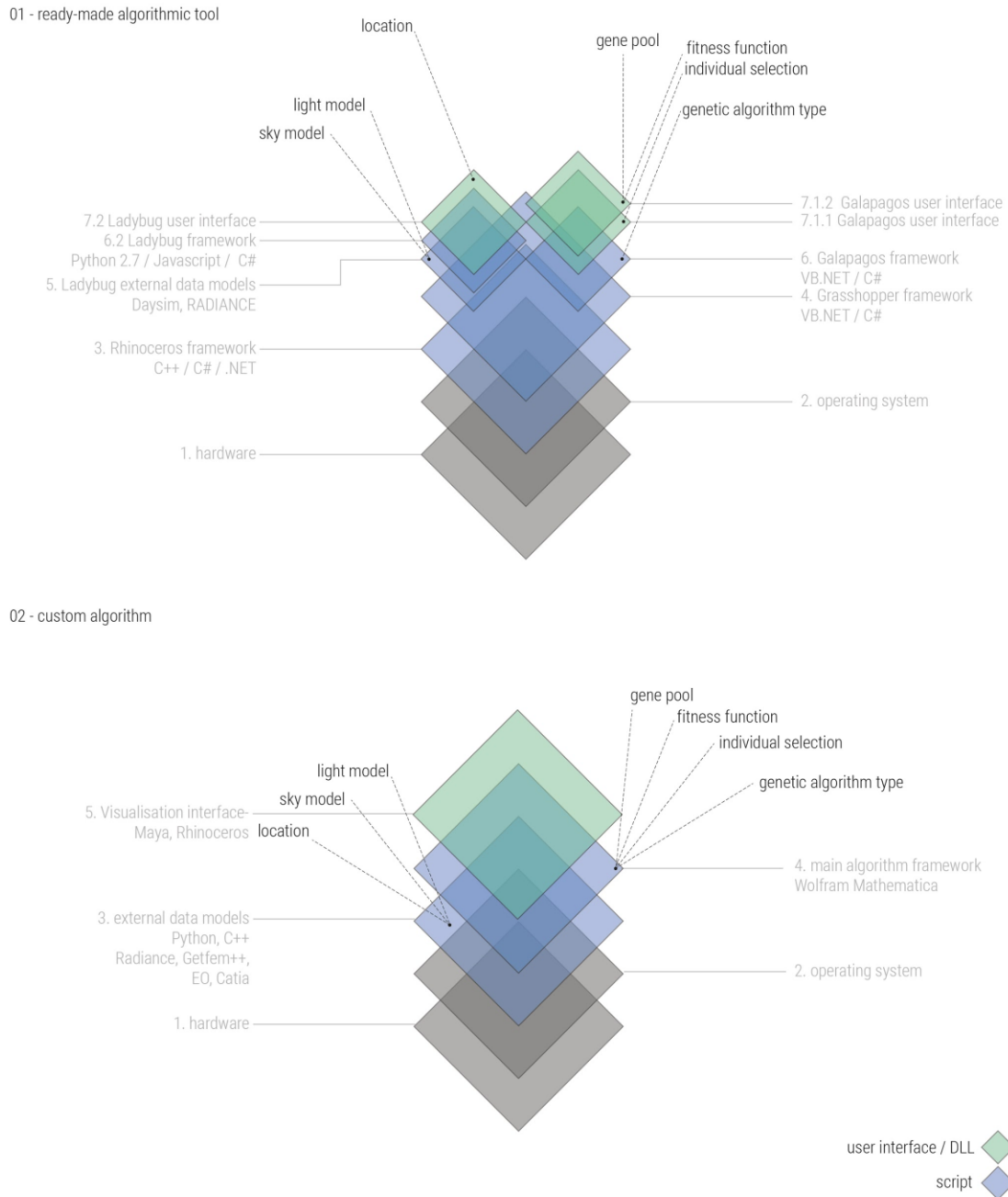


Figure 4
Interface thickness for a ready-made tool and a custom algorithm

Promoting performance: rationalization at all costs?

Performance-based design (Hensel 2013) is guided by the evaluation of rationally measurable performances, examining quantifiable physical phenomena affecting the built environment through verified mathematical models and assessing designs for them to withstand best those phenomena. Mechanical, thermal, acoustic, light performances, but also financial or environmental costs are thus taken into account during the design process. The interest in performance-based design in the computational field in architecture is not only found in link to evolutionary tools, as it is a global trend, but given that optimization is evidently at the core of it, links between both are strong.

The recent spike in the promotion of performance-based approaches is built on two major arguments. First, since the resulting designs have been designed resorting to optimization tools - often genetic algorithms - they can only be better. In recently released tools for Dynamo and Revit for example, Autodesk developers thus impose design decisions and fitness functions upon their users, arguing that those decisions ensure optimized solutions regarding building norms and therefore a rational approach to architectural design (Das & al 2016, Nagy & al. 2017). One could nevertheless ask what exactly an optimized architecture is, the very nature of the discipline being tirelessly interpreted and discussed by many in the profession. In the same spirit, Stanislas Chaillou (2020), in a recent talk, is going as far as claiming to know what any architect would ask from a computer, an impossible question to answer given the diversity of architectural practices.

The second main argument resorted to by proponents of performance-based design is the vastness of choice proposed to the architect, brought back to his tool user status, a vastness s.he is free to explore given that algorithms, developed and guided by other, are here to take care of tedious constraints such as mechanical or thermal characteristics. Evolutionary design tools such as those proposed by the Digital Structures group (Brown & Mueller 2017), where the user is expected to select his/her favorite individuals, are illustrations of this approach. Such rethoric and approach however replace the freedom given by mastery of tools by an illusion of choice and reduce the tacit dimension of architecture to aesthetic taste. As performance-based design is promoted by the same practitioners developing evolutionary tools, the epistemological and the technical optimization bias mutually strengthen, aligning with the general tendency to quantification and rationalization despite the many critics that have been made to this modern model (Latour 1991).

Conclusion

The present research proposes a comprehensive methodology of investigation of algorithmic typologies currently in use in the computational field in architecture, focusing on three elements : practitioners, tools and projects. The methodology has been applied to evolutionary design tools in order to investigate the technical and epistemological optimization bias they entail. The research shows the existence of two approaches to handling the tension between formalization required by algorithms and tacit knowledge required by architectural practice. Whereas one of the approaches displays an intertwinement of explicit and tacit, of qualitative and quantitative, the other prioritizes a rational, performance-based approach. Given the conditions of development of

evolutionary tools shown, this latter conception of architectural practice has spread while recent tools enabled a democratization of the resort to evolutionary techniques. By transforming evolutionary algorithms into ready-made tools, or black boxes of CAAD, this process of development prevents the implementation of tacit knowledge within them, potentially impoverishing architectural design.

References

- Barricelli, N.A., “Esempi numerici di processi di evoluzione”, *Methodos 1954*, p. 45–68, 1954.
- Berg, M., *Rationalizing Medical Work : Decision-Support Techniques and Medical Practice*, The MIT Press, 1997.
- Brown, N. & Mueller, C., “Designing with data: moving beyond the design space catalog”, in Nagakura, T., Tibbits, S., Mueller, C. (eds.), *Disciplines & Disruption : ACADIA 2017 Conference Proceedings*, pp. 154-163, 2017.
- Calixto, V. & Celani, G., « A literature review for space planning optimization using an evolutionary algorithm approach : 1992-2014 », in *Project Information for Interaction : Proceedings of the 19th SIGRADI Conference*, Universidade Federal de Santa Catarina, 2015.
- Carmo, M., *The alphabet and the algorithm*, The MIT Press, 2011.
- Carmo, M., *The Second Digital Turn : Design Beyond Intelligence*, The MIT Press, 2017.
- Chaillou, S., “IA & Architecture”, talk at the Pavillon de l’Arsenal, 27 February 2020.
- Cogdell, C., *Toward a Living Architecture?: Complexism and Biology in Generative Design*, University of Minnesota Press, 2019.
- Collins, H., *Tacit and Explicit Knowledge*, University of Chicago Press, 2010.
- Cross, N., *Designerly Ways of Knowing*, Springer, 2006.
- Das, S., Day, C., Dewberry, M.J., Toulkeridou, V., Hauck, A., “Automated Service Core Generator in Autodesk Dynamo : Embedded design intelligence aiding rapid generation of design options”, in Herneoja, A., Österlund, T., Markkanen, P. (eds.), *Complexity & Simplicity : eCAADe 2016 Conference Proceedings*, University of Oulu, pp. 217-226, 2016.
- Daston, L. & Galison, P., *Objectivity*, The MIT Press, 2007.
- Dawkins, R., *The Blind Watchmaker*, Norton, 1986
- Fraser, A., “Simulation of Genetic Systems by Automatic Digital Computers I. Introduction”, *Australian Journal of Biological Sciences*, Vol. 10, num. 4, p484 1957
- Frazer, J., *An Evolutionary Architecture*, Architectural Association, 1995.
- Gaudillière, N., “Towards an History of Computational Tools in Automated Architectural Design - The Seroussi Pavilion Competition as a Case Study”, in M. Haeusler, M. A. Schnabel, T. Fukuda (eds.), *Intelligent & Informed - Proceedings of the 24th CAADRIA Conference - Volume 2*, Victoria University of Wellington, Wellington, New Zealand, 15-18 April 2019, pp. 581-590, 2019.
- Gaudillière, N., “Computational Tools in Architecture and Their Genesis : The Development of Agent-Based Models in Spatial Design”, in *CAADRIA 2020 Conference Proceedings*, to be published.
- Guenoun, E. (ed.), *Pavillon Seroussi : biothing, DORA (Design Office for Research and Architecture)*, EZCT Architecture & Design Research, Gramazio & Kohler, IJP – George L. Legendre, Xefirotarch, HYX, 2007.

- Hensel, M. (ed.), *Performance-Oriented Architecture: Rethinking Architectural Design and the Built Environment*, AD Special Issue, John Wiley & Sons, 2013.
- Holland, J.H., *Adaptation in Natural and Artificial Systems*, The MIT Press, 1975.
- Kuhn, T., *The Structure of Scientific Revolutions*, University of Chicago Press, 1962.
- Latour, B., *Science in Action*, Harvard University Press, 1987.
- Latour, B., *Nous n'avons jamais été modernes*, La Découverte, 1991.
- Lynn, G. (ed.), *Archaeology of the Digital*, Sternberg Press, 2013.
- Masure, A., *De l'influence des interfaces*, Emilieu, 2019
- Nagy, D., Lau, D., Locke, J., Stoddart, J., Villaggi, L., Wang, R., Zhao, D., Benjamin, D., "Project Discover : An application of generative design for architectural space planning", in Turrin, M., Peters, B., O'Brien, W., Stouffs, R., Dogan, T. (eds.), *2017 Proceedings of the Symposium on Simulation for Architecture and Urban Design*, University of Toronto, 2017.
- Negroponte, N., "Towards a theory of architecture machine", *Journal of Architectural Education*, Vol. 23, num. 2, pp. 9-12, 1969.
- Picon, A., "Les ingénieurs et la mathématisation. L'exemple du génie civil et de la construction", *Revue d'histoire des sciences*, vol. 42, num. 1-2, pp. 155-172, 1989.
- Prusinkiewicz, P. & Lindenmayer, A., "Graphical modeling using L-Systems", in Prusinkiewicz, P. & Lindenmayer, A., *The Algorithmic Beauty of Plants*, pp. 1-50, Springer, 1990.
- Roussel, M., "De la cybernétique à l'architecture numérique : retour sur un demi-siècle de théories, pratiques et projets expérimentaux", *Canadian Journal of Communication*, Vol. 42, pp. 515–533, 2017.
- Serraino, P., *History of Form*Z*, Birkhauser, 2002.
- Smith, R., *Fabricating the Frank Gehry Legacy: The Story of the Evolution of Digital Practice in Frank Gehry's office*, CreateSpace, 2017.