AUTOMATING THE ARCHITECTURAL DESIGN OF APARTMENT POINT BLOCK

IGOR JANSEN[1], ŁUKASZ PIĄTEK[2], MARTA CYGAN, JAKUB HLEBOWICZ, BARTOSZ KRAJEWSKI, AGATA MISZEWICZ, AGNIESZKA NOWACKA, BARTŁOMIEJ ROGUSKI, and MARTA SZABELEWSKA
[1,2]*Warsaw University of Technology*
[1]*igor.jansen.dokt@pw.edu.pl, 0000-0002-1648-5458*
[2]*lukasz.piatek@pw.edu.pl, 0000-0002-0302-0723*

Abstract. Can architectural design be automated? The development of CAAD tools led to the point where this question can be relevant. To prove that some branches of architectural design can be automatized to a substantial extent we aimed at building a digital tool able to perform automated preliminary design of residential building in the typology of a point block (tower). The algorithmic process was split into three consecutive stages corresponding to the traditional design method: massing, floorplans layouts and apartments design. The workflow is linear, with some form of feedback-loop-driven optimization. The result of the study is an algorithmic tool in the Rhino + Grasshopper environment automatically proposing a point block of optimal size, position and orientation on a given plot, dividing the floorplans into flats of optimal size structure and solving the flat layouts. The system proved to be working on different plots where the existing buildings were used as a reference for performance evaluation.

Keywords. Automation, MOO, Optimization, Algorithmic Design, Apartment Block Design, Architectural Design, Massing

1. Introduction

"Not all things architectural can be arrived at through drawing" (Evans, 1986) wrote Robin Evans pointing on the inevitable separation architects encounter between drawing, the traditional medium of design, and building, the final outcome of their work (Iwamoto, 2009). Design methods based on software and scripting that emerged and became popular among architects (Mitchell, 1970; Alexander, 1967; Stiny & Gips, 1971) aim at bridging this gap. Currently, Computational Design (CD) challenges previous architectural design conventions and praxis which were heavily based on manual drafting (Rocker, 2006). CD approaches are increasingly important in architecture (Duarte, 2005; Knecht & Koenig, 2010; Kwieciński & Pasternak, 2016; Dogan, et al., 2015; Guo & Li, 2017; Hamacher & Kjølsrud, 2014), and Algorithmic Design (AD), which is based on algorithms (Terzidis, K., 2004) to generate designs, is one of the most promising. Even though not everything in architecture is quantifiable,

HUMAN-CENTRIC, Proceedings of the 28th International Conference of the Association for Computer-Aided Architectural Design Research in Asia (CAADRIA) 2023, Volume 2, 491-500.

geometric and interdependent (Meredith, Sasaki, 2008), the AD is the key to automation which, besides enhancing design exploration, helps to deal with recurrent changes, to obtain results quicker, and supports optimization (Canestrino, G. 2021).

Considering the complexity and diversity of the design process, apartment buildings represent a promising field for automation because of the large number of quantitative factors that must be included, e.g.: efficiency, minimum insolation, view obstruction (screening) and strong interdependencies like parts-to-whole and whole-to-parts relation and repetitiveness of units. They all are being addressed at the beginning of the design process - Massing Study (MS) that aims at the optimal distribution of built form on the plot. The outcomes of MS are 3D building envelope and data on areas, types and number of apartments in it. The process is extremely time-consuming as it requires considering both the complex geometric relationships as well as the parameters defined in building code and zoning acts like coverage, density, heights, setbacks, daylighting and others.

Recently, several tools for MS appeared such as Autodesk Spacemaker or Digital Blue Foam (DBF). They enable very simplified generation of building blocks and their approximate comparative analyses, which do not, however, ensure that the designed solutions are the best ones. The generative methods implemented in both projects require manual input with strong assumptions from the designer to present interesting results. They are rather tools designed to simplify the study of urban compositions equipped with the preliminary analysis of results, than programs for optimization or automation of architectural design. Some other tools developed to automate the residential architecture design such as PRISM, works of Hersey and Freedman (1992), Duarte (2005), are based on shape grammars. Some works incorporate ML with voxel sampling – Koh (2023). Another approach is evolutionary simulations in concept creation augmentation such as works of Dogan, Saratsis and Reinhart (2015), Nahara and Terzidis (2006), Showkatbakhsh & Makki(2022), a modular building performance assessment and optimization tool (Kiss, Szalay, 2022) or MOT - a script generating, analysing, and optimizing a 3D model of an apartment building (Jansen, Piątek, 2020). None of them covers the conceptual design process comprehensively, from site location and geometry and desired flat structure to apartment design.

2. Methods

This project aims at demonstrating that for relevant architectural conceptual tasks we can already effectively use AD running on standard PCs. To simplify the problem, we limited the task to MS for single buildings and focused on the typology of a point block (tower). Despite this simplification the range of the task ("from site to apartments") is still comparable to a real-life situation in architectural practice. Our Design Algorithmic Tool (DAT) was developed in Rhinoceros (ver. 6 and 7), Grasshopper (v.1.0.0007), Wallacei optimization add-on (v.2.7), LunchBox (v.2020.11.2.0) and EleFront add-on (v.4.3.0 and 5.0.0). Computer specs: 8 core @4200 MHz AMD Ryzen 7 5800X CPU with 2 x 16 GB RAM @1600 MHz, Windows 11 Pro.

DAT works in three main stages – Optimum Massing, Floor Layouts & Apartment layouts. This division corresponds to the traditional design performed by human. To search for the best solution DAT operator must change the input parameters (such as the expected number and size of flats) and repeat the process.

2.1. STAGE 1 – MASSING

This stage is focused on shaping and placing the building as efficiently as possible, according to zoning and building code. To do so design is optimized with several factors that are crucial to follow to provide a healthy and habitable space.

Firstly, the 3D maximum envelope is created, which represents the space possible to be filled by the building without casting shadows on the existing buildings. In the beginning the plot boundary is inset according to the required setbacks. The remaining plot area is extruded to the maximum allowed height. Then Ladybug sunlight analysis is used to get proper sun vectors hitting the windows of nearby buildings (according to Polish Building Code at least 3 hours of sunlight at equinox is required). The vectors that collide with 3D plot boundary volume are transformed into slicing planes which cut off the top of the 3D extrusion to get the final volume where the building can be placed. Then a schematic building (simplified to cuboid) is placed on the plot (Fig. 1).

Secondly, we use the multi-objective evolutionary algorithm NSGA-II through Wallacei, which after translating the factors into numbers enabled to easily minimize all negative impacts that can occur during the massing stage. All parameters of building such as position, side lengths, number of floors and height are controlled by the algorithm and can be optimized. Several objectives are evaluated. The first objective is responsible for controlling if the building fits inside the defined boundary volume. The surface of the building is divided into hundreds of points. The algorithm minimizes the number of points outside the volume. Second objective is responsible for maximizing the gross floor area to achieve maximal residential space simultaneously minimizing the footprint of the building. The third objective is responsible for providing at least 3 hours of sunlight to all residential flats of the designed building. To do so, the number of vectors colliding with nearby buildings is minimized. The fourth objective is view obstruction defined according to the Polish Building Code. The algorithm minimizes the number of windows not meeting the view obstruction rule. The fifth objective is responsible for adjusting the building to the urban composition of the site. It analyses each of the nearby buildings in a specified distance range and selects the most repetitive and dominating facade direction and adjusts the rotation of the building to it.

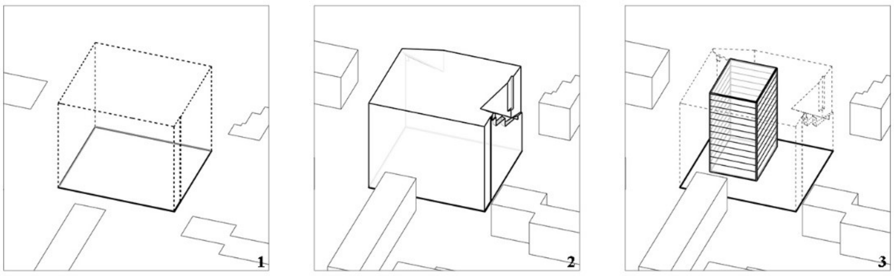

Figure 1. Massing optimization steps: 1) Creating a volume according to the plot. 2) Reducing the volume by splitting form to comply local law and sun accessibility. 3) Matching the best possible solution for the building inside the volume.

2.2. STAGE 2 – FLOOR DIVISIONS

	Number of rooms	Min – max area [m^2]	Min – max depth [m]	Min – max length [m]	Possible aspect
M1	1	25 - 37	5,93 – 7,12	5,12 – 6,02	Single
M2	2	37 - 55		6,02 – 8,97	Single or corner
M3	3	55- 70	5,93 – 8,12	7,42 – 10,42	Corner
M4	4	70 - 100		10,45 – 14,72	Corner or triple
M5	5	100 - 150		11,00 – 18,00	

Table 1. Selected parameters used as rules for apartment evaluation.

The goal of this stage is setting the circulation core and apartment outlines on every floor in a way that the MS meets the desired apartment schedule, containing desired fraction of building area taken by each apartment type. The algorithm based on exhaustive search method tests many floorplan variants generated automatically.

The process of creating a floor division begins with placing a core in the centre of the floor. Dividing the floor into apartments starts from creating 4-apartment layouts, using 6 (Fig. 2) schemes based on Polish building code and common practice in residential architecture design. Secondary divisions are then added, with the constraint of not creating single-aspect apartments facing north. By picking sets of primary and secondary apartments divisions, layouts with 4, 5, 6, 7 and 8 apartments are generated. Apartment labels are then assigned based on predefined areas. Floorplans with apartment areas and width outside accepted range are eliminated. The apartments which exceed the depth limit are narrowed (Table 1). After creating all possible floorplans for the initial core position, a recursive algorithm moves the core by a fixed value in 8 directions and repeat the process to provide a greater variety of versions.

Next, pairs of floorplans are created by matching a layout and its subdivision to guarantee a logical building structure. It was proven through trial and error that using two different floorplan schemes allows to fulfil any apartment schedule to a satisfying extent. Variations are then created, consisting of given two floorplans distributed differently amongst all available stories. E.g. if the building consists of 5 floors in total, there could be 0 floorplans A and 5 floorplans B, 1 floorplan A and 4 floorplans B, etc. For each pair of layouts, percentage of total building area taken by each apartment type is calculated. Pair evaluation is based on calculating the difference between the obtained and the desired percentage for each type and finding the biggest difference for each pair. The best solution is the one in which that value is the smallest.

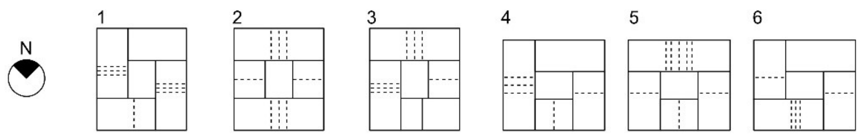

Figure 2. Six basic floorplan layouts with their secondary division variants (shown in dotted lines). Mirrored variants not presented.

2.3. STAGE 3 – APARTMENTS

Basing on an assumption that there is a finite number of acceptable floorplans that fit into a standardized rectangular shape, it is possible to create a database of verified apartment layouts chosen and adapted to a given boundary. In DAT 5 typical apartment layouts were predefined (M1-M5). Each plan is described by parametric, geometric variables that can be modified within a defined scope (Figure 3).

Figure 3. Selected typical layouts and their min. and max. boundaries (dotted lines).

Upon the algorithm's start, each apartment set in stage 2 is assigned as one of the floorplan types, depending on its area (Table 1). For the simplification of the algorithm, the area ranges are programmed to be disjunctive and continuous, hence there is always one best match to be chosen. As the numerical data comes from the previous step, alongside the guaranteed proportions, no further analysis is needed. The flat is then divided into day and night zones – their proportions are adjusted depending on the overall size of the flat. Further divisions follow several shape-grammar rules that allow for adaptive non-uniform scaling. One of them is a maximum and a minimum width of each room type based on the functional analysis of possible furniture and window placements. When the outline is ready, the doors are placed. To prove to correctness of the layout the kitchen top and water fixtures in the bathrooms are marked.

3. Results

The whole workflow starting with massing optimization and ending with apartment layout design was tested in real locations in Warsaw, Poland, on the plots with existing single point blocks and zoning parameters regarding maximum height and maximum density available. For the test purposes the existing point blocks were eliminated from the site 3D model and the DAT's goal was to design a building in this place (Figure 4). The difference between existing building and the proposal generated automatically was used for evaluating DAT's performance (Table 2).

DAT takes up to 25 minutes to complete. Average calculation times per task are: Plot analysis: 20sec; Massing optimization: 16 min (@Generation Count = 100, Generation Size = 25); Floor Divisions: 6min; Apartments Layout: 2 sec. The process can be shortened by reducing generation count or size during the evolutionary optimization.

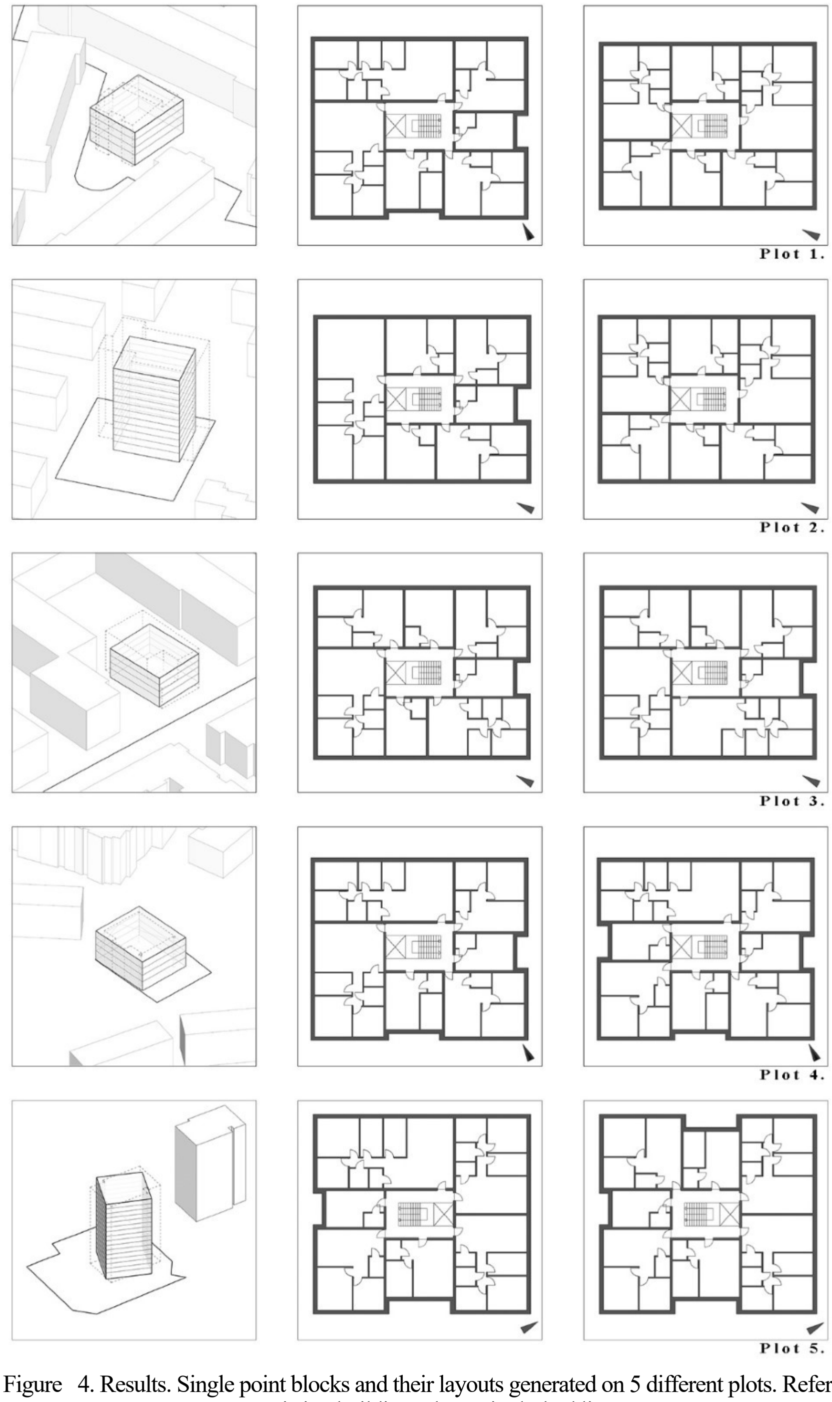

Figure 4. Results. Single point blocks and their layouts generated on 5 different plots. Referenced existing buildings shown in dashed lines.

		Plot 1	**Plot 2**	**Plot 3**	**Plot 4**	**Plot 5**
Location in Warsaw		Rzeczypos-politej 33	Lasek Brzozowy 12	Branickiego 9	Elbląska 55	Szolc-Rogo-zińskiego 17
GFA [m^2]		1 221	4 989	1 411	1 467	8 604
ref. GFA [m^2]		1 613	8 303	2 368	1 213	9 565
GFA/ ref GFA [%]		76	60	60	121	90
No of floors		3	11	3	3	15
Building height [m]		13	4	13	13	50
Max height [m]		14.5	40	14.5	15	50
Apartment schedule goal/ achieved [%]	M1	20 / 17	20 / 22	30 / 36	10 / 21	10 / 15
	M2	20 /17	20 / 17	20 / 9	10 / 16	30 / 25
	M3	35 / 43	35 / 35	30 / 32	40 / 37	30 / 25
	M4	20 / 17	20 / 20	15 / 22	20 / 11	20 / 30
	M5	5 / 6	5 / 6	5 / 1	20 / 15	10 / 5

Table 2. Results comparison.

Results are analyzed by comparison to values known from existing buildings, which are assumed to be valid. The ratio of achieved GFA to the GFA of the referenced building is used to evaluate Stage 1 – Massing. Based on five presented cases GFA ratio remains between 60% and 121%. The achieved results are better if the referenced building plan is approximate to a rectangle, which is a logical result of our simplification in applying the cuboid as the only possible shape.

Stage 2 is evaluated by comparing the proposed and desired apartment. Similar ratios are desirable. There is a noticeable correlation between a higher number of building floors and better results. Maximum difference between the goal and the result is 11 percent points for M2 in 3-storey building (Plot 3). Minimum difference is 0 percent points for M3 and M4 in 11-storey building (Plot 2). Because only two different layouts are assigned for the whole building - more floors allow for better apartment structure fine-tuning.

Stage 3, the apartment layouts generation should be evaluated as non-numeric data regarding known residential architecture design principles. Figure 4 presents that all plans are drafted correctly and that they match the schematic floor layout.

4. Discussion

Despite the positive results presented in the work, in the authors' opinion, it should be clearly emphasized that at the current stage the tool development has a number of serious limitations, which include, above all:

- Limiting the number of buildings to one: For the further development of the project, it would be particularly important to enrich the functionality by generating more than one building on one plot - such an analysis is more complex but it would make the algorithm applicable in much wider group of design cases.

- Limited typology: The functionality of the algorithm should also be extended to generate buildings other than point blocks. Shape radically simplified to cuboids does not allow for using the maximum available space for the building. Improvements like podiums or recessed floors would be easy to implement and interesting.
- Flat topography assumption: At the current stage the tool assumes that any terrain provided is flat and can be freely built-up.
- Limited performance: DAT's computation time varies depending on the machine used, as well as the complexity of a given problem, but some optimizations such as replacing exhaustive search algorithms in stage 2 could be implemented. Number of nearby buildings, and size of the analyzed plot makes crucial difference in tool workflow. In case of multiple spaces where building can be generated the evolutionary part of the algorithm should process adequately larger generation size in goal of broader spectrum of results at early generations.
- No parking analysis: A significant improvement in the work of the algorithm would be the ability to analyze the zoning requirements for parking and the possibilities of the plot in this respect.
- Human interaction is limited to setting the input data. Manual form-finding can be done by manipulating the shape of the plot that impacts the footprint.

5. Conclusions

DAT aimed at verifying the thesis that architectural design process can be effectively automated with algorithmic tools at any stage, with particular emphasis on the creative conceptual stage. This thesis was proved by using the algorithm in designing the initial urban and architectural concept of a multi-family residential building. The tool defined both the optimized massing of the building as well as the apartment layout taking into consideration the desired apartment schedule. The project is developmental in nature and leaves a lot of room for expansion of the implemented functions.

The difference between DAT and tools such as DBF or Spacemaker is that the presented approach is fully automatic and contextual. It aids in the conceptual phase automatically producing viable solutions while the other tools don't approach the conceptual design process holistically starting with the preliminary massing and ending with the apartment design, as well as universally to the point that the automation could be performed on any plot meeting the basic criteria (such as maximum size).

We have presented a new simple optimization tool proving that the conceptual phase of architectural design is prone to automation. The Tool was verified on 5 reference building plots in Warsaw, Poland. The results delivered by DAT show that the system quickly produces reasonable solutions within the initial limitations proposing solutions in the range of 60% to 121% referenced GFA. The use of both multi-objective optimization and exhaustive search algorithms within one pipeline proved to be effective and fast compared to traditional design process. The results presented show that DAT should be developed further in order to eliminate its shortcomings and optimize its performance.

Acknowledgements

This research was initiated in 2022 during the Experimental Design Studio at the Faculty of Architecture, Warsaw University of Technology, when the first prototype of DAT was built. Therefore, we would like to acknowledge the co-autorship of the tool to the students: Bayrak Sule, Bléas Arnaud, Cygan Marta, Galant Krzysztof, Golod Darya, Grzybowska Weronika, Hlebowicz Jakub, Kot Maria, Krajewski Bartosz, Krawczyk Julia, Makarevych Anastasiia, Miszewicz Agata, Morina Tringa, Mościcki Mateusz, Nowacka Agnieszka, Nowobilski Marzena, Platonava Vera, Roguski Bartłomiej, Skowronska Alicja, Skurzewski Michał, Szabelewska Marta and tutors: Piątek Łukasz Ph.D, and Jansen Igor, Ph.D. candidate.

References

Alexander, C. (1964). Notes on the Synthesis of Form (Vol. 5). *Harvard University Press.*

Canestrino, G. (2021). Considerations on Optimization as an Architectural Design Tool, *Nexus Network*, Journal,23, 4, (919-931). https://doi.org/10.1007/s00004-021-00563-y

DBF. (2022). Digital Blue Foam. Retrieved November 25, 2022, Retrieved from https://www.digitalbluefoam.com.

Deb K., Pratap, A. Agarwal S. and Meyarivan T. (2002). 'A fast and elitist multiobjective genetic algorithm: NSGA-II', IEEE *transactions on evolutionary computation*, 6(2), pp. 182-197.

Dinçer, A. E., Çağdaş, G., & Tong, H. (2014). A computational model for mass housing design as a decision-support tool. *Procedia Environmental Sciences*, 22, 270-279. https://doi.org/10.1016/j.proenv.2014.11.026

Dogan T., Saratsis E., and Reinhart C. (2015). 'The optimization potential of floor-plan typologies in early design energy modeling', *Proceedings of BS* 2015.

Duarte, J.P. (2005). 'Towards the mass customization of housing: the grammar of Siza's houses at Malagueira',*Environment and planning B: Planning and Design*,32(3), pp.347-380.

ELEFRONT. (2022). EleFront v5.0.0. Retrieved November 26, 2022, from https://www.food4rhino.com/en/app/elefront.

Evans R. (1986). *Translations from Drawing to Building and Other Essays* (The MIT Press, Cambridge: 1986) p. 159.

Iwamoto l. (2009). Digital Fabrication, Architectural and material techniques (Princeton Architectural Press, New York,) p4.

Grasshopper (2022). Grasshopper. Retrieved November 26, 2022, from https://www.grasshopper3d.com.

Guo, Z., & Li, B (2017). 'Evolutionary approach for spatial architecture layout designenhanced by an agent-based topology finding system',*Frontiers of Architectural Research*, 6(1), pp. 53-62. https://doi.org/10.1016/j.foar.2016.11.003

Hamacher, H., & Kjolsrud, E (2014). 'Growing floor plans and buildings -generation and optimization through algorithms and synthetic evolution', *Proceedings of 2nd International Conference of Biodigital Architecture & Genetics*

Hersey, G., & Freedman, R. (1992). *Possible Palladian Villas: (Plus a Few Instructively Impossible Ones)*, MIT Press. ISBN: 9780262581103

Jansen, I., & Piątek, Ł. (2020). The Evolutionary-algorithm-based Automation of the Initial Stage of Apartment Building Design*. D1.T1.S3. Design and Computation of Urban and Local System, XS to XL Vol.2,eCAADe 38* | 105.

King, RA., Deb, K., & Rughooputh, H (2010). 'Comparison of NSGA-ii and SPEA 2 on the multiobjective environmental/economic dispatch problem', *University of Mauritius Research Journal*, 16(1), pp. 485-511.

Kiss, B., & Szalay, Z. (2020). Modular approach to multi-objective environmental optimization of buildings. *Automation in Construction*, 111, 103044. https://doi.org/10.1016/j.autcon.2019.103044

Knecht K.,& Konig R. (2010). 'Generating floor plan layouts with kd trees and evolutionary algorithms',*Generative Art Conf*, pp. 238-253.

Koh, I. (2023). Voxel Substitutional Sampling: Generative Machine Learning for Architectural Design. In Design Computing and Cognition'22 (pp. 341-358). Cham: Springer International Publishing.

Kwiecinski, K., & Pasternak, A. (2016). 'Automatyzacja w projektowaniu architektonicznym', *Builder*, 232(11), pp. 32-36.

LUNCHBOX (2022). Grasshoper App. Retrieved November 25, 2022, from https://www.food4rhino.com/en/app/lunchbox.

Larsen, N.M. (2012). Generative algorithmic techniques for architectural design, Ph.D. Thesis, Aarhus School of Architecture.

Meredith, M., & Sasaki, M (2008). From control to design: Parametric/Algorithmic architecture, *Actar-D*.

Mitchell, W.J. (1970). 'A computer-aided approach to complex building layout problems', *Proceedings of EDRA2 Conference*.

Nahara T., & K. Terzidis. (2006). Multiple-Constraint Genetic Algorithm in Housing Design. *Pp. 418-425 in 25th Annual Conference of the Association for Computer-Aided Design in Architecture*.

PRiSM 2.0 (2022). Mayor of London, Bryden Wood and Cast. Retrieved November 26, 2022 from https://www.prism-app.io/.

Rhino (2022). Rhino 6 and Rhino 7. Copyright 1993-2022 Robert McNeel & Associates. Retrieved November 26, 2022, from https://www.rhino3d.com.

Rocker, I. (2006). When codes that matter Architectural Design p. 23.

Showkatbakhsh, M., & Makki, M. (2022). Multi-Objective Optimisation of Urban Form: A Framework for Selecting the Optimal Solution. Buildings, 12(9), 1473. https://doi.org/10.3390/buildings12091473

Spacemaker (2020). Copyright 2020 Spacemaker. Retrieved November 26, 2022, from https://www.spacemakerai.com.

Stiny, G., & Gips, J. (1971). 'Shape grammars and the generative specification of painting and sculpture.', *IFIP Congress (2)*, pp. 125-135.

Terzidis, K. (2004). Algorithmic design: A Paradigm Shift in Architecture? Architecture in the Network Society*, 22nd eCAADe Conference Proceedings*, p. 202.

Terzidis, K. (2003), Expressive Form: A Conceptual Approach to Computational Design, *Spon Press (London and New York)* p. 72.

Tuhus-Dubrow, D., & Moncef, K. (2010). "Genetic-Algorithm Based Approach to Optimize Building Envelope Design for Residential Buildings." *Building and Environment 45 (7):* 1574–81.

Wallacei. (2022). Wallacei. Retrieved November 26, 2022, from https://www.wallacei.com.