Growing floor plans and buildings

- generation and optimisation through algorithms and synthetic evolution

André Hamacher[1], Eirik Kjølsrud[2]

[1]*Freelance Architect, Germany,* [2,]*University of Applied Sciences, Trier, Germany*

[1]*http://www.planungen-hamacher.de,* [2]*http://www.kjolsrud.com*

[1]*hamacher@planungen-hamacher.de,* [2]*eirik@kjolsrud.com*

Abstract. Advances in computational design such as the availability of algorithms and increase in computing power make possible new and more inclusive design methods. We developed tools which generate and optimise floor plans through evolutionary algorithms. Several contradicting problems, such as flexibility, building services, construction, material usage and building regulations were taken into account, with daylight and spatial organisation as the key assessment criteria. The tools present various valid alternatives which the architect can further investigate in an iterative design process. A modular implementation allows the designer to focus on areas of interest and the digital tools thus function as a dialogue partner for the architect and can be used interactively. The aim of this paper is to evaluate the possibilities and benefits of these tools in the early stages of design projects. In prototypical design projects the tools were put to the test and delivered satisfactory results. However, the strived for speed and interactivity leaves room for improvement. An algorithm based on multi-objective optimisation and subdivision is therefore complemented and compared with a second technique based on additive space planning resulting in more diagrammatic results, with higher performance and more interactivity.

Keywords. Evolutionary Algorithms; Space planning; Interactive design; Shape grammars.

Introduction

„We, the architects, have nothing on our desks to support our daily task in the office, not a single tool that generates a useful architectural floor-plan layout." [1]

The challenges presented by urban densification and demographic change are difficult to meet with conventional planning methods. Multifaceted architectural problems are simplified by necessity and it is difficult to create adaptable designs, allow for a participatory design process and develop a multitude of relevant design alternatives.
Advances in computational design such as the availability of algorithms and increase in computing power make possible other and more inclusive design methods. Nevertheless, there is no tool available that generates valid design alternatives and evaluates them in the early investigative design phase.

We developed algorithmic tools which generate floor plan alternatives as a possible response to programme and context. The aim of this paper is to evaluate the tools and estimate the possibilities and benefits of these tools in the early stages of design projects.

1 (Donath, Lobos, 2010)

State of the Art

Three problems of algorithmic floor plan generation are identified, namely: description, generation and optimisation. The developed tools utilise methods with which each domain is solved, extend these and combine them to a comprehensive solution. The three methods that form the basis are *Shape grammars*, *Slicing trees* and *Genetic Algorithms*.

Shape Grammars are rule-based systems with which designs can be generated through the recursive application of shaping rules. The research of Duarte (2005) into mass customisation housing and "discursive grammars" provides a conceptual starting point for the generative tools presented in this paper. Through a participatory process, an architectural programme is described. The resulting descriptive grammar is translated into a floor plan heuristically by applying (de)forming rules onto a starting shape. The recursive structure of rules can be easily described and is thus suited for algorithmic manipulation.
Knecht and König (2011) suggest to generate architectural plans through subdivision. The subdivision operations are stored in a binary tree, a slicing tree. This structure allows a hierarchical organisation of rooms in zones; spaces could be divided in private and public areas and then further subdivided. Architectural morphologies and the programme can be described using tree structures.
Space planning through Shape Grammars and subdivision allows for a multitude of variants. Problems of this nature cannot be solved efficiently in a linear manner. (König, Schneider, Knecht, 2011) *Genetic Algorithms* has proved to be an applicable heuristic method. (Elezkurtaj, Franck, 2002) These algorithms control the search process through a fitness function. Architectural problems are multi-layered and it is ambitious to evaluate designs with a single function. *Multiobjective Optimisation Algorithms* consider several, often opposing, criteria and rank solutions accordingly. The ranked solutions can be browsed based on criteria and this allows for a creative and interactive search process. One of the evolutionary algorithms used in this paper is an improved *Pareto-Algorithm* described by Zitzler (2001).

Tools

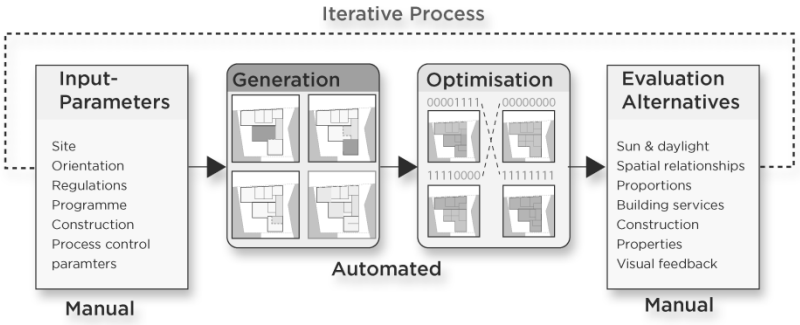


Figure 1
Iterative design process with tools

The tools presented here aim to implement an iterative process in which multiple valid building designs can be generated. The process involves a manual set up, an algorithmic generation and a manual evaluation with the option of iteratively changing the set up.

The chosen evaluation criteria are interdependent and of an opposite nature. This results in a complexity with no single solution. The idea is to provide an exploratory and participatory tool with precise feedback to assist in the first stages of a design process.

Subdivision tool

The subdivision tool was developed by Eirik Kjølsrud for his Masters thesis (2013).

Because the tool relates to a site, boundary conditions such as borders, building regulations and neighbouring structures must be defined in the setup. Results from previous runs can be used to give the project a direction.
An architectural programme is defined. This could be chosen from a catalogue or be created on the fly. [Fig 2] As flexibility is a pronounced design goal, several programmes can be chosen. A design alternative in which more programmes successfully fit can thereby be identified.

The programmes are defined as trees. Unlike the binary tree structure suggested by Knecht (2011), multiple branches are allowed, enabling more complex morphologies. Parameters including preferred location, area range, proportions, use and building service dependencies are set for each space, enabling the architect to investigate building types ranging from housing to offices.
A random starting population is generated and serves as the genetic material for the Pareto algorithm. New alternatives arise through *crossover*[2] and *mutation*.

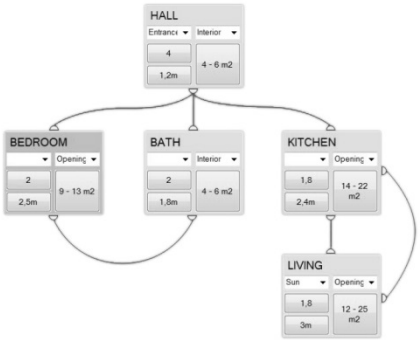


Figure 2
Spatial relationships

The *space allocation process* is divided into three steps. First, a building shape is generated through (de)formation of a seed shape. Then the shape is divided into sections. In the third step, the sections are subdivided into spaces to accommodate the programme.

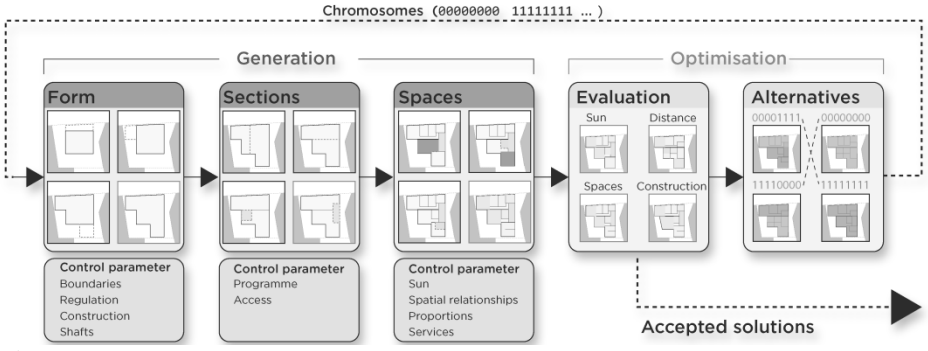


Figure 3
Space allocation process with the subdivision/pareto tool

2 Because the programmes and subdivision have different lengths, the chromosomes are a long string of operations/parameters. The maximum used number of chromosomes limit the crossover point. As the chromosome parts differ for each section a hiearchical crossover method suggested by Bentley (1996) was implemented.

Three rules shape the building by modifying a seed geometry:

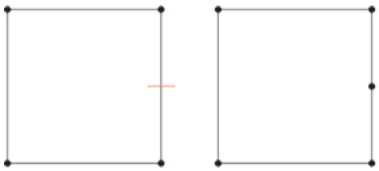


Figure 4
Divide edge – An edge is divided at a specified point

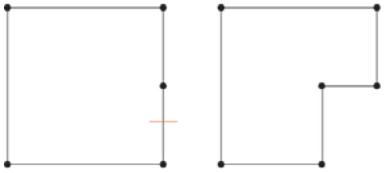


Figure 5
Shift edge – An edge is shifted orthogonally inwards or outwards, adding new edges

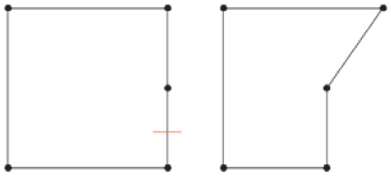


Figure 6
Move edge – An edge is moved and connected edges are modified as well

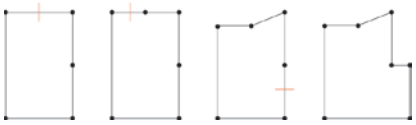


Figure 7
Recursive shaping rules

The rules are applied recursively and can change any polygon into new shapes. The rule sequence make up the first part of the Chromosome. The above transformation [Fig 7] could be described like this: {D1} {M0} {S3}

In the second step the building shape is recursively subdivided horizontally or vertically into sections to accommodate a programme. Any shape can be used as a seed and the architect can set the deformation magnitude and "freeze" parts of the shape. An access core is determined by a chromosome parameter from a predefined selection and positioned at an intersection.
Further, the building sections are divided into spaces using subdivision rules. In this morphogenetic process the programme structure is mapped recursively to a chromosome containing subdivision rules and the spaces unfold within the building shape. The programmes [Fig 2] are mixed with a genome containing shaping rules and parameter values, producing a specific floor plan, the phenome. This is a different approach than by Duarte/Knecht where spaces are generated and room descriptions can arbitrarly be used to label all spaces afterwards. When spaces are linked to space proportions and size, it ensures that the spaces fit their purpose and irrelevant spaces are not created. This is also the case for the additive tool. The coupling is kept loose, allowing deviation from the structure. This results in unforeseen alternatives.

Knecht and König (2011) only consider horizontal and vertical subdivision of space even though this „*limits the solution space of all possible floor plans*".

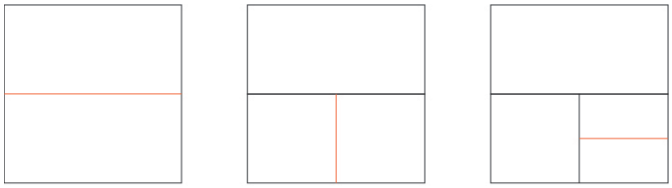


Figure 8
Horizontal and vertical subdivision rules

This paper extends the possible floor plan morphologies by adding the subdivision rules *Inserted*, where a rectangle is inserted into the space, and *Reversed Insert*, where the leftover space of the inserted rectangle is further subdivided.

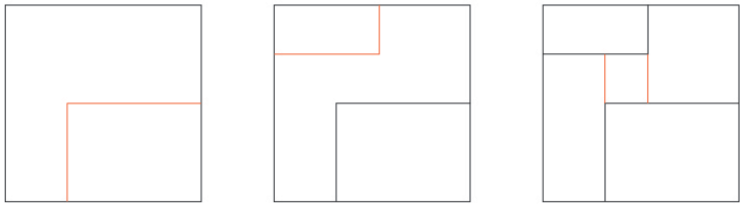


Figure 9
Extended subdivision rules

These rules can be described in the chromosome in the same way as the building shape rules: {I0} {R1} {V2} {V3}.
A benefit of subdivision is that no leftover spaces remain. The subdivision rules work orthogonally within a set grid and function in all the polygonal shapes found in previous steps.
If the programme merges with a different chromosome the result is a completely different floor plan. The same chromosome thus creates different spaces when applied to a different programme.

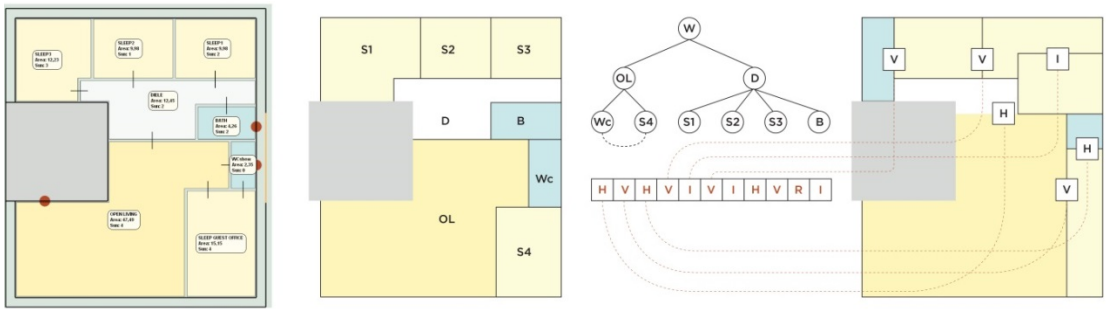


Figure 10
Chromosome and programme

A number of criteria are evaluated to verify whether the generated floor plans are valid and to estimate their objective qualities. The criteria are independent of the genetic algorithm and can be selected to fit the design purpose. We wanted to test the possible complexity of the tools and included as many evaluation areas as possible. The following criteria were used in the evaluation process for both tools:

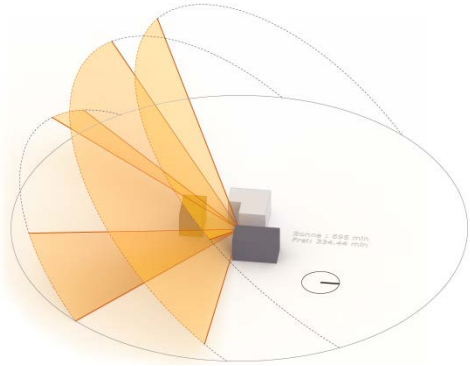


Figure 11
Azimuth solar calculations

Sun and daylight has an enormous effect on the quality of living. For each space, the amount of sun (and overshading from neighbours) is calculated [Fig 11] and thus the sunniest space and location (interior/exterior) can be detected.

Spatial relationships. Short distances and connectivity between neighbouring rooms are strived for. Relationships can be applied hierarchically or between two separately positioned spaces. Wall lengths are adapted to allow for doors.

Proportions. In the generated alternatives the opposite needs of criteria become visible. To maximise sun and spatial relationships tubular rooms are often generated. These rooms are not useful and to avoid these, proportion rules and furnishing dimensions (dining table/bed & wardrobe/accessible bathroom) are set.

Service shafts. Spaces can be defined with a dependency on building services. These are foreseen on bracing wall elements and access cores. If it is not possible to put a room close to a shaft, a new one is created, reducing the overall rating of the floor plan. If two neighbouring rooms share a shaft, the impact of a new one is reduced. For multi-storey housing existing shafts can be defined in the initial setup, enabling floor plans to be coordinated and optimised over more stories.

Construction. The building shape is evaluated according to span length (no internal load-bearing walls), modular building parts, material use and bracing.

Flexibility. In parallel searches, multiple programmes are tested for each building shape. In this manner flexibility and adaptability can be investigated.

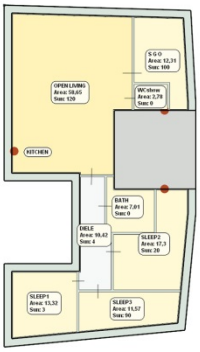
As the multi-object optimisation algorithms deliver multiple and sorted alternatives, the investigation of the design alternatives is a creative process in itself. The architect can browse alternatives based on criteria and focus on areas of interest. What the tools objectively devalue could be rendered unimportant by an architect if the floor plans show interesting qualities overall. Properties such as floor area, volume, amount of daylight and material are provided in addition to a visual presentation of the floor plan.

The tool was tested on a housing design for a difficult site in Frankfurt. The irregular site is a leftover space between buildings where distance space must be respected. The orientation of the site is not optimal for housing and there is extreme overshading from neighbours.
For the architectural programme one-bedroom-, two-bedroom- and family-apartments were defined with the purpose of creating a building shape and structure that could accommodate all and thus be adaptable to future changes in the space requirements.

After about 30 generations the tool found floor plans that allowed optimal daylight and flexible apartment designs with a minimal use of material.

To allow daylight in the second floor the building edges had to be distanced from the neighbours. [Fig 12] These narrow shapes could not contain the prescribed programme.

Eventually a shape emerged were an atrium allowed the required outside connection. This was optimised to balance material use and floor area.



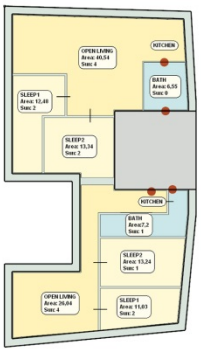
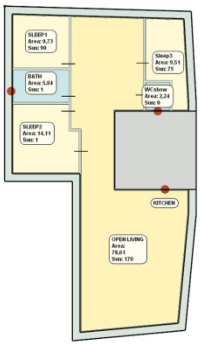


Figure 12
Second floor alternatives



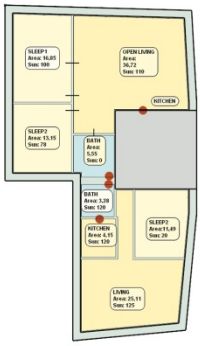
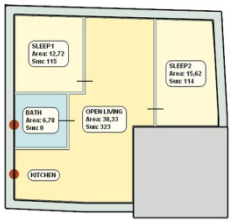


Figure 13
Third floor alternatives

On the third floor direct daylight is achieved without the building being too narrow and the tool suggests a shape were all apartments programmes fit.



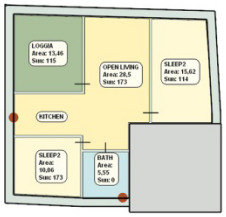


Figure 14
Fourth and fifth floor

For the fourth floor building regulations dictate a large distance to neighbouring buildings, therefore the tool suggests a compact shape. The terracing of the building produces roof spaces where an architect can imagine outdoor spaces. For the fifth floor there are no possible outdoor spaces and the programme is changed to include loggias.

The tool suggested valid alternatives for the boundary conditions. As promising directions emerged it proved successful to focus the search and adjust the evaluation criteria. Through iterative use service shaft locations across floors could also be optimised. High-ranking alternatives accommodate the programme, ensure adaptability and reflect the evaluation criteria. As the number of evaluated alternative building shapes and floor plan layouts grow, the higher-ranked represent the logical directions. This eliminates a lot of unsucessful design directions without a lot of time invested.

The dialogue between architect and tool is advantageous. Because of the complexity the tool does not deliver results in real-time and the interaction is limited to iterative use. In order to investigate the interactivity a second tool is developed by André Hamacher in which the benefits of genetic algorithms is combined with the ability of direct interplay.

Additive tool

The additive tool evaluates the same criteria as the other; these are however experimentally combined in a single fitness function of a genetic algorithm. The building shape is vaguely suggested through the clustering of spaces.
The fundamental evaluation criterion in the additive method is the relationship between spaces. Based on their relationships (positive/negative), the spaces are organised in clusters. The importance of other criteria can be adjusted to allow different focus.

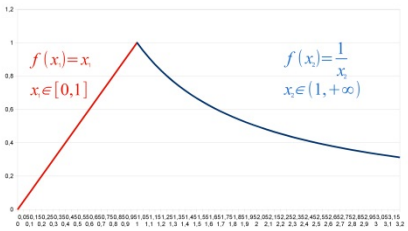


Figure 15
Spatial relationship calculation

Figure 16
Interactive intermediate step

To improve performance and allow for interaction the genetic algorithm is combined with an iterative optimisation algorithm. After each generation the fittest individuals are cloned and alternatives are generated in which the spatial relationships work as attractors/deflectors to optimise space positioning. In this phase the spaces are self-organising. After this step, the modified alternatives are injected into the pool of individuals and in effect part of the next generation in the evolutionary process. In the iterative phase, it is possible for the architect, at any given time, to change and add alternatives to the population and thereby interact with the tool. The goal of the second tool, and the reason for this modifications, is to create a real-time dialog-partner for the daily workflow.

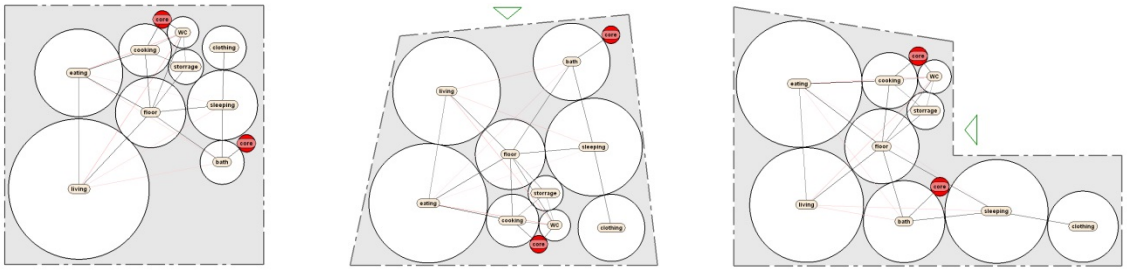


Figure 17
Additive space planning with circles

In a first version spaces were added as circles reflecting their area and displaying the spatial relationships. Through the *gentic algorithm* the positioning of the spaces was optimised.
Further development of the tool included geometric shapes with proportions related to use. [Fig 18] Daylight and connectivity are the main criteria to optimise the floor plan layout.

Figure 18
Rectangular and proportional additive space planning

The tool is quicker than the subdivision alternative and can diagrammatically layout programmes to a satisfactory degree. The real-time interplay offered to the architect is a benefit of the method. The primary shape of the spatial structure is controlled by a user-defined boundary. Because of its additive nature, the tool creates leftover spaces. Efficiency is achieved with a loss of spatial effectivity.

Decisions and the role of the architect

The prototypical studies show, that the tools can generate relevant, optimised and flexible floor plans. This however, can only take place in cooperation with an architect as the tool only evaluates alternatives based on quantifiable objectives. To direct the tool and sort alternatives the architect needs a thorough understanding of the process and an available interface. Both tools offer a visual as well as a property-based presentation of the floor plans. An architect can assess these quickly and bring subjective design decisions and architectural know-how into the process. The role of the architect and the creative efforts thus shifts towards that of an assessor and director rather than those of an author. The decision is ultimately left to the architect and not the tool.
In order to make a decision, there has to be alternatives. A conventional planning process where only a few alternatives are examined does not leave much room for decision making outside the detail level. The abundance of alternative effortlessly evaluated and suggested by the generative tools offer inspiration and the possibility of comparing alternatives in a way that permit the architect to make informed decisions.

Conclusion

The tools were able to output valid design suggestions on the basis of as many as ten, often opposite, evaluation criteria. With an increase in complexity the speed of the calculation and the ability to achieve real-time results decline. The process thus demands a close interplay between tool and architect. Within an iterative process alternatives can be found quicker and more precisely if the architect actively engages in a dialogue with the process and add architectural creativity to the equation. The accuracy of the tool feedback and the ability to evaluate and sort a huge amount of alternatives make the tools a great asset and a potential time-saver in early design phases. Even if no perfect result is found, the floor plan suggestions can be the sparks that trigger the creative design process. The idea of starting a design phase knowing that a number of alternatives could “meet the brief” is appealing. The tools can be used in participatory processes where requirements change in design phase. As multiple directions can be explored in parallel, the tools aid in thinking “outside the box” and investigating the adaptable potential of designs and structures.

Future work

The two tools differ in their approach, speed and output. The more complex evaluation and the more precise architectural floor plans delivered by the subdivision tool should be coupled with the speed and interactivity of the additive tool. The subdivision tool was not optimised for speed. Simple code optimisation resulted in an increase in generated floor plans from 5 to 30 pro second. There is room for improvement in this area. As the tool is written as a multithreading application it should be possible to move the calculation to cloud computing.
Both tools are modular and adaptable to different design tasks. It proved to be relatively simple to add new evaluation criteria to the process. Additional criteria such as views, costs and more complex 3-dimensional spatial proportions could be added. As an extension of this the tools could be made more scalable, allowing them to be used on both urban and detail scale as well as for floor plan generation.
To handle the complexity the tools only consider one floor at a time. As building shape, shafts, cores and bracing can be set up in the initial stage of the process, this allows for a manual communication of the different floors. In future work the floors should be generated in parallel.

References

Bentley, Peter J., and Jonathan P. Wakefield. "Hierarchical crossover in genetic algorithms." *Proceedings of the 1st On-line Workshop on Soft Computing (WSC1)*, 1996.

Donath, Dirk and Danny Lobos. „The problem of space layout in architecture: A survey and reflections“ *arquiteturarevista Vol. 6, n° 2 Dezember,* 2010.

Duarte, José Pinto. "Towards the mass customization of housing: the grammar of Siza's houses at Malagueira." *Environment and planning B: Planning and Design* 32.3, 2005.

Elezkurtaj, Tomor, and Georg Franck. "Algorithmic support of creative architectural design." *organization 2*, 2002.

Hamacher, André. "Der grundlegender Aufbau von evolutionären Algorithmen und deren Möglichkeiten zur Anwendung in der Architektur" *Adato 1*, 2011.

Kjølsrud, Eirik. "Master thesis: Digital design in urban densification – algorithmic floor plan generation and optimisation." *University of Applied Sciences Trier,* 2013.

Knecht, Katja and Reinhard König. "Evolutionäre Generierung von grundriss-layouts mithilfe von Unterteilungsalgorithmen." *Bauhaus Uni Weimar*, 2011.

König, Reinhard, Sven Schneider, and Katja Knecht. "KREMLAS: Entwicklung einer kreativen evolutionären Entwurfsmethode für Layoutprobleme in Architektur und Städtebau." *Bauhaus Uni Weimar*, 2012.

Laumanns, Marco and Eckart Zitzler. "SPEA2: Improving the strength Pareto evolutionary algorithm." *ETH Zürich,* 2001.