Evolutionary Algorithms in Urban Planning

Tomor ELEZKURTAJ & Georg FRANCK

Dipl.-Ing: Tomor Elezkurtaj; o.Univ.Prof. Dipl.-Ing. Dr. Georg Franck, Vienna University of Technology,
Department of Computer Aided Planning and Architecture, Treitelstrasse 3, A-1040 Vienna, Austria; e-mail: Tomor@osiris.iemar.tuwien.ac.at

ABSTRACT

The functions supported by commercial CAD software are drawing, construction and presentation. Until now, no programs supporting the creative part of architectural and urban problem solving are on the market. The grand hopes of symbolic AI of programming creative architectural and urban design have been disappointed. In the meantime, methods called New AI are available. Among these methods, evolutionary algorithms are particularly promising for solving design problems. The paper presents an approach to town panning and architectural problem solving that combines an evolutionary strategy (ES), a genetic algorithm (GA) and a Particle System. The problem that remains incapable of being solved algorithmically has to do with the fact that in architecture and urbanizm form as well as function count. Because function relates to comfort, easiness of use, and aesthetics as well, it is hopeless to fully specify the fitness function of architecture. The approach presented circumvents a full specification through dividing labor between the software and its user. The fitness function of town plans is defined in terms only of proportions of the shapes, areas and buildings to be accommodated and topological relations between them. The rest is left to the human designer who interactively intervenes in the evolution game as displayed on the screen.

1 A BRIEF HISTORY OF AI IN COMPUTERAIDEDDESIGN

The history of computer aided design is not without irony. Starting with raising hopes of turning creative design into a scientifically disciplined method of problem solving[1], the computer eventually entered the planners business as a down-to-earth means of saving costs. Instead of substituting the human designer, the computer proved a useful tool for drawing and constructing, a convenient mailbox and filing cabinet. Today, creative design is as intuitive, non-scientific and chaotic as it has ever been. The most conspicuous jobs done by computers in architecture and town planning are sophisticated presentation and on-line co-operation.

The high rising hopes of the early days were fuelled by the then impressive progresses of symbolic AI. The approach of symbolic AI to human intelligence is that of programming the use of language. Language is a broad concept, encompassing the use of words, symbols and even shapes. The way suggesting itself of combining CAAD with symbolic AI is formalizing shape grammars. Shape grammars are sets of forms, symbols and rules defining the way in which, e.g., meaningful architectural plans are composed of elements symbolizing walls, ceilings, windows, doors, stairs etc. Plans are meaningful only if they are well formed, i.e., that the elements are defined in a clear-cut way and manipulated according to syntactical rules. Take a shape grammar rich enough for composing plans, formalize it, program it, and you have enabled the machine to enter a trial-and-error process of producing plans which, in turn, are capable of being evaluated and selected in the manner candidate moves in chess are.

Remarkably, the use of computer driven shape grammars came close to passing the architectural Turing test. Computerized grammars came up with, e.g., mock Palladian villas and fake Frank Lloyd Wright prairie houses. It would be hard to identify these imitations as fakes if trustworthily presented as long-forgotten originals (figure 1).[2] Nonetheless, symbolic AI never came up with modules suitable for commercial CAAD software. The reason is that design strategies promising in architecture resist being reduced to a calculus of winning.

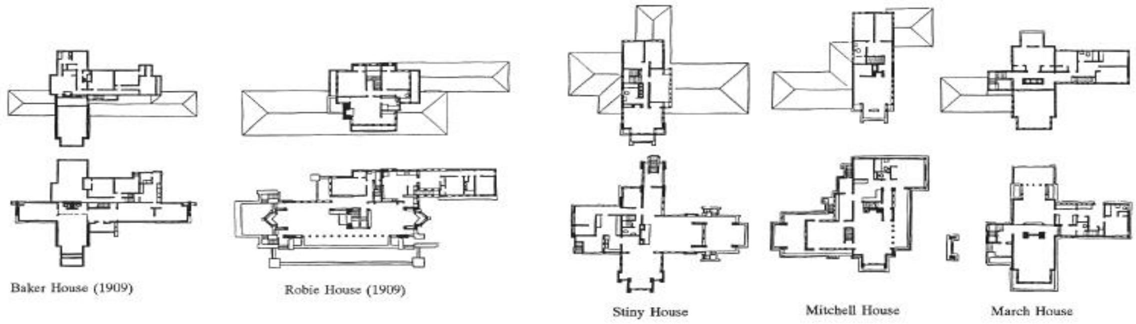


Figure 1. originals (1,2) and imitations (3,4,5)

The problem of winning a chess match is well defined. The problem of winning an architectural or urban competition is ill defined. In chess, problem solving consists in scanning alternatives that are given. Even though problem solving in architecture or urbanizm may consist of adapting existing designs, creative solutions tend to result from redefining the problem in certain ways. It is not unusual, to say the least, that the solution results from redefining the problem until a design favored for heterogeneous reasons arguably becomes a solution. In chess, redefining the problem to be solved is forbidden. In architecture and urbanizm, the problem to be solved is systematically open to revision since function in architecture is an open concept.

The strategy of problem solving pursued by symbolic AI was that of decomposing complex problems into simpler problems until a level of elementary problems is reached. This strategy proved successful in areas where two conditions are fulfilled. The first condition is that such a level of elementary problems in fact exists, the second condition is that the solutions of these elementary problems can be formalized. In creative design, neither of these conditions is fulfilled. Preparation of the design problems to be solved by manipulating shape grammars thus consisted in a radical re-interpretation of what architecture means. It consisted in consistently disregarding the functional aspect of architecture. The 'sentences' formed by the use of shape grammars only ever were

[1] See, e.g., Alexander (1964).

[2] See Koning/ Eizenberg (1981) and Stiny/ Mitchell (1981).

syntactically well formed. The search strategies only ever looked for solutions satisfying the *formal* prerequisites for being a possibly meaningful plan. The meaning itself was elaborately kept out of the process. The designs emulating the famous examples did so by restricting themselves to the purely formal aspect of the shapes manipulated. The design strategies were successful because of, not in spite of, disregarding any pragmatic or semantic meaning of the designs produced.

Emulating a style of design is one thing. Helping the designer in being creative is another. In the first case the emphasis is on reproduction, in the latter case it is on exploration. Exploration in architectural design rarely is a play of form only. Whether or not it is consciously guided by functional viewpoints, it obeys functional criteria as long as it is supposed to be architectural or urban design and not just graphics. Exploration in architectural and urban design may very well include re-interpretation of functional requirements; functional requirements, however, that are fixed and accepted can be disregarded at the cost only of turning exploration into an idle play of forms. In order to facilitate and not just inspire exploration, the software should be capable of obeying functional requirements to a certain, non-vanishing, extent.

As soon as function is involved, the strategy of solving the problem by way of its final analysis comes to an end. The functional description of a building and thus of its components never is complete. Without giving a complete description of the function to be served, the design problem remains ill defined and open to interpretation. Creative human design is capable of turning the vice of being ill defined into the chance of inventing things never seen before. How to circumvent the need of a final analysis without foregoing the powers of AI, however?

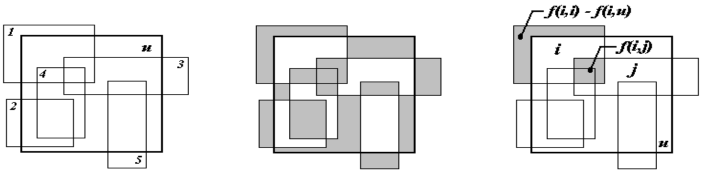
It is new AI that offers a non-analytic approach to problems such as these. New AI differs from old, symbolic, AI in that the paradigm of intelligent behavior has shifted. Instead of human language, it is now biological life that provides the paradigm cases of intelligent strategies. These strategies are not closer to, but even further away than good old symbolic AI from human way understanding the problem we are dealing with. Evolutionary algorithms simulate a generative process that is explicitly supposed to be 'blind'. There is no understanding and thus no meaning whatsoever in the way in which artificial evolution works. Nevertheless, evolutionary processes are the most creative and inventive known. Though allegedly primitive in comparison with biological evolution, theses strategies have proved capable problem solvers in various fields of engineering.[3] They wait to be applied to architecture.[4]

2 ARTIFICIAL EVOLUTION

The approach presented makes use of the fact that a good deal of the functional requirements to be observed in town plans can be expressed in terms of proportion and topology. The fitness functions that the system is supposed to optimise are restricted to (1) eliminating gap between and overlap of the shapes to be accommodated, (2) approximating the proportions preferred, and (3) optimising the neighbourhood relations between the areas. The rest is left to the human designer who interactively intervenes into the game of artificial evolution as displayed on the screen.

The strategy of eliminating the gaps and overlaps that occur when the shapes are fitted into the outline consists of a mutation driven evolutionary strategy. The fitness function minimizes the sum total of gap, overlap, and overflow (see figure 2 and figure 3). After being initiated, a population of design variants is subject to random change concerning position and proportion. Selection acts through reproduction from generation to generation. The fitter a variant, the higher is its reproduction rate. The proportion preferred is approximated through filtering probabilities.

The search space of this particular problem is characterised by a multitude of global optima. Since the risk of being caught in a local optimum is minimal, this simple evolutionary strategy is adopted for reasons of speed.



$$Sg = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \left(S[i] \cap S[j]\right) + \lambda \sum_{i=1}^{n} \left(S[i] \setminus S[u]\right)$$

Figure 2

The search space of optimising the neighbourhood relations between the areas is much more complex. Moreover, the search space asks to be worked through more thoroughly. In order to accomplish this, a genetic algorithm (GA) is adopted which combines mutation with cross-over.

The operation performed by the GA is a kind of re-interpretation of the areas arranged. It changes the functions attributed to the areas in order to optimise the neighbourhood relations (see figure 5). The output of the GA is thus turned into an input of the strategy fitting in the rooms and vice versa. The fitness that the GA is supposed to maximise is measured in terms of the weights specified in the topological matrix (W). Elements w_{ij} of this symmetric matrix express the importance of the neighbourhood of areas i and j. The weights w_{ij} are specified by the user. Solutions of the arrangement problem have the form of matrix T. The value of element t_{ij} is higher in the case that shapes i and j are nearby and lower when they are not. The fitness function ($W_T \rightarrow$ max) to be maximised sums the products $w_{ij} * t_{ij.}$

[3] As introductory texts see Goldberg (1989) and Mitchell (1996).

[4] For a general presentation of the idea and an overview of the approaches having surfaced until then see Frazer (1995).

$$W=\begin{pmatrix} \theta & w_{12} & w_{13} & w_{14} & \cdots & w_{1n} \\ w_{21} & \theta & w_{23} & w_{24} & \cdots & w_{2n} \\ w_{31} & w_{32} & \theta & w_{34} & \cdots & w_{3n} \\ w_{41} & w_{42} & w_{43} & \theta & & \vdots \\ \vdots & \vdots & \vdots & & & w_{n-1,n} \\ w_{n1} & w_{n2} & w_{n3} & \cdots & w_{n,n-1} & \theta \end{pmatrix} \qquad T=\begin{pmatrix} \theta & t_{12} & t_{13} & t_{14} & \cdots & t_{1n} \\ t_{21} & \theta & t_{23} & t_{24} & \cdots & t_{2n} \\ t_{31} & t_{32} & \theta & t_{34} & \cdots & t_{3n} \\ t_{41} & t_{42} & t_{43} & \theta & & \vdots \\ \vdots & \vdots & \vdots & & & t_{n-1,n} \\ t_{n1} & t_{n2} & t_{n3} & \cdots & t_{n,n-1} & \theta \end{pmatrix}$$

$$W_T=\sum_{i=1}^{n}\sum_{j=1}^{n}\left(w_{ij}*t_{ij}\right)\; w_{ij}\in\Re;\quad t_{ij}\in(0,1)$$

The reason for adopting this mixed strategy lies, among other things, in speeding up the process. Speed is crucial for interaction with the user. In the same way in which strategies ES and GA interact, their interplay interacts with the interventions on the part of the user. The user intervenes into the game of artificial evolution via mouse and editing. The interface through which the user interacts with the system is characterised by the following features: (a) The design variant being the fittest at the moment is displayed on the screen. (b) The arrangement as well as the geometry of the shapes can be changed via the mouse (see figure 4). (c) The weights of the neighbourhood relations can be edited during the run.

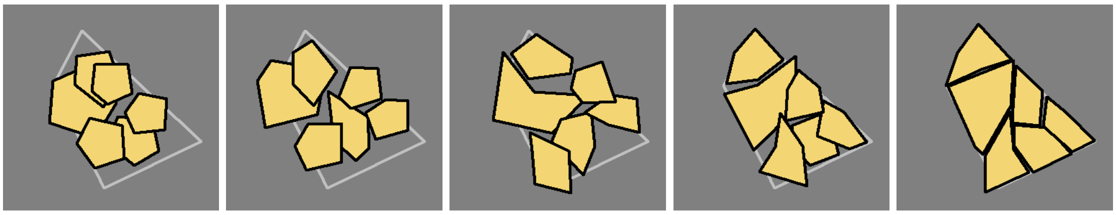


Figure 3 (minimize overlap and overflow / ES)

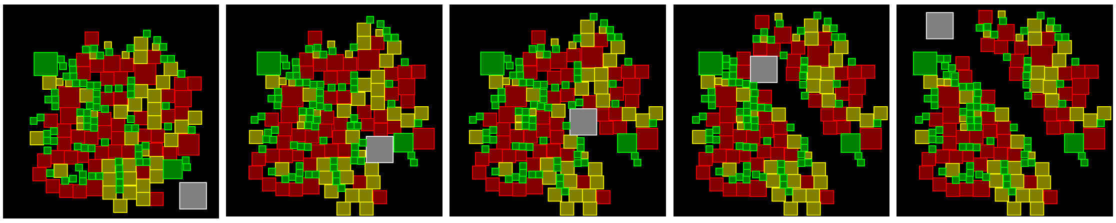


Figure 4 (interactive intervention)

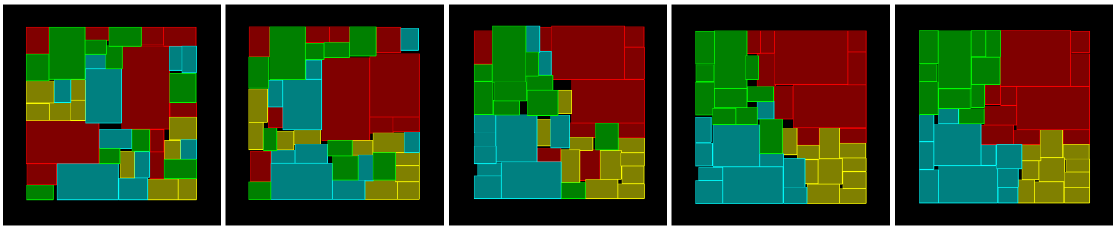


Figure 5 (re-interpretation of the Shapes / GA)

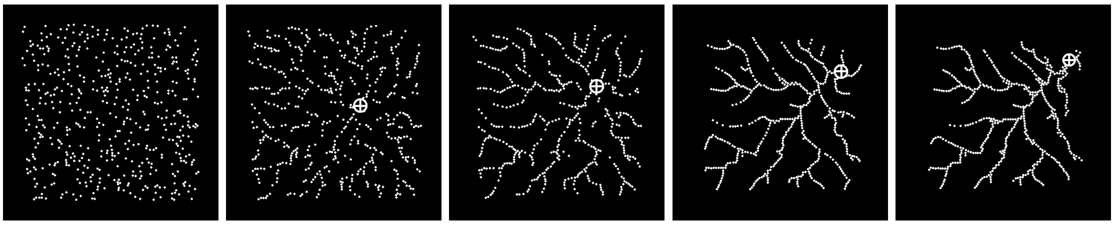


Figure 6 (self-organization of a particle system)

REFERENCES

Alexander, Christopher (1964), Notes on the Synthesis of Form, Cambridge, Mass.: Harvard University Press
Frazer, John (1995), An Evolutionary Architecture, London: Architectural Association
Goldberg, David E. (1989), Genetic Algorithms in Science, Optimization, and Machine Learning, Reading, Mass.: Addison Wesley
Holland, John H. (1995), Hidden Order, Reading, Mass.: Addison Wesley
Koning H./ J. Eizenberg (1981), The language of the prairie: Frank Lloyd Wright's prairie houses, in: Environment and Planning B 8, pp. 295-323
Mitchell, M. (1996), An Introduction to Genetic Algorithms, Cambridge, Mass.: MIT Press
Stiny, Georg/ William J. Mitchell (1981), The Palladian grammar, in: Environment and Planning B 5, no. 1, pp. 5-18