A Genetic Algorithm for use in Creative Design Processes

Philippe Marin, UMR MAP 694 / ARIA Ecole Nationale Supérieure d’Architecture de Lyon
Jean-Claude Bignon, UMR MAP 694 / CRAI Ecole Nationale Supérieure d’Architecture de Nancy
Hervé Lequay, UMR MAP 694 / ARIA Ecole Nationale Supérieure d’Architecture de Lyon

Abstract

This paper deals with natural growth mechanisms applied to architectural design processes. We implement a genetic algorithm as part of a digital tool to be used in the creative design process. This evolutionary process is evaluated by means of environmental parameters, passive solar qualities and the designer's individual requirements. A morphogenetic process is put forward, based on a “metamorphosis strategy”.

1 Introduction

Phylogenies symbolize the evolution of species: mutation, crossover and natural selection govern this evolution. These concepts provide a guide to the development of artificial and digital devices capable of simulating natural mechanisms, and bringing into being new properties and qualities. Evolutionary design has its roots in computer science, design and evolutionary biology. It is a branch of evolutionary computation, which extends and combines CAD and analysis software. It does not hesitate to borrow ideas from natural evolution.

In this paper, we deal with morphogenetic mechanisms applied to architectural design. We consider that evolutionary systems could help to guide designers in creative directions. We were not looking for an organic or genetic form, but for digital tools inspired by genetic mechanisms capable of assisting and supporting design projects. We feel that such tools could be conducive to architectural creativity. We focussed, in particular, on the initial phases of the design process.

2 Related work

2.1 Integral evolutionary design

There are numerous examples of evolutionary algorithms: genetic algorithms, proposed by J. H. Holland in 1975, evolution strategies, proposed by P. Bienert, I. Rechenberg and H. P. Schwefel in 1960, evolutionary programming, proposed by L. F. Fogel in 1966, and genetic programming, developed by J. Koza in 1992.

Genetic algorithms are probably the best-known evolutionary search algorithms. Starting with J. Holland in 1975, whose aim was to explain the adaptive processes of natural systems and to design artificial systems based on them, there have been several applications of genetic algorithms. Caldas (Caldas and Norford 2003) used a genetic algorithm to optimize construction budgets by minimizing HVAC, lighting energy and construction costs. Malkawi (Malkawi et al. 2003) offered a Java environment using a genetic algorithm as an evolutionary algorithm and a CFD performance as an evaluation mechanism. Nishino (Nishino et al. 2001) provided an example of an interactive evolutionary computation applicable to a creative design process. One of the most famous authors in the field of the evolutionary architecture is John Frazer, who has been involved in the use of genetic techniques for building envelope designs since 1968, and the use of genetic algorithms since 1990 (Frazer et al. 2002). He explores the possibilities of expressing architectural concepts as generative rules so that their evolution and development can be accelerated and tested by the use of computer models.

In general, evolutionary design can be divided into four main categories: evolutionary design optimization, creative evolutionary design, evolutionary art and evolutionary artificial life (Bentley 1999). Here, we are essentially interested in creative evolutionary design and evolutionary design optimization, whose overlap is usually known as "integral evolutionary design".

Evolutionary algorithms are traditionally used to solve optimization problems. In addition, they can be used as a design aid. The evolutionary approach is a generative testing method that fits the procedures for design synthesis and evaluation in the design process. The characteristics of the approach are:

- A pool or population of design solutions, rather than a single solution.
- The selection of individuals according to their adjustment to the fitness function.
- The generation of new solutions through mutations and crossovers of previous elites.

With the advent of new technologies in the field of evolutionary design, the designer's role shifts from that of a creator of individual styles to that of a meta-designer or creator of an entire style family (Soddu 2004). Design-as-Product or Ideas-as-Product, as defined by Soddu, is focused on the act of designing the species representation or "DNA" of a designed object. In addition, changes in design can be used to stimulate creativity (Todd and Latham 1992). Large numbers of evolutionary steps can be generated in a short time, and the emergent forms are often unexpected.

2.2 Morphogenesis

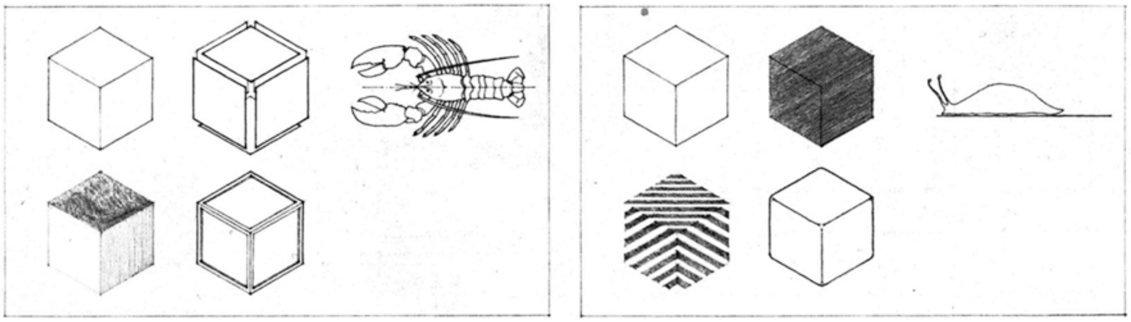

Figure 1. Strategies of morphogenesis

Work done by Ching (Ching 1979) (Figure 1) on the conception of forms in architecture has made it possible to identify two main strategies for the production of form. The first can be metaphorically represented by a lobster, and consists of creating forms through adjustment and combination. The original form is composed of unitary forms that can be added, juxtaposed or superimposed. The second is represented by a slug, the original form undergoing morphological but not topological modifications, with operators such as twist, stretch and pinch. We will refer to this as "transformation through metamorphosis".

Wetzel (Wetzel et al. 2006) offers two classes of morpho-semantic operators: unitary operators that act on a single object, and binary operators that act on two or more forms. The unitary operators cover isometrics (translation, rotation, symmetry) and homothetic transformations. Transformations on a higher scale, known as "modifiers" can be used to obtain the desired shape. These are: bending, tapering, skewing, twisting and stretching.

Natural growth is an expression of morphogenesis. As far as biology is concerned, the understanding of a form implies the description of a generative process, i.e. a type of morphogenesis. We based our morphogenesis transformations on strategies of metamorphosis. We applied certain "modifiers" in a recursive, successive way in order to transform our initial pattern, and thus to explore a solution space. This strategy guided our morphogenesis. Rather than looking for the solution of the starting problem, the idea was to define a program whose execution would lead to the solution.

3 Experiment

3.1 The development environment

We focussed on the initial phase of the design process, where the designer is looking for ideas, with reference to Gero (Gero and Maher 1998) regarding the matching of the generative process to the architectural design process. This paper deals with the question of creativity, and the definition of the design process.

In our proposal, a generative process is defined as an evolutionary process and a solar passive fitness condition. This means that the designer is looking for an appropriate formal means of pushing forward his work-in-progress. Figure 2 gives an illustration of a possible population.

This experiment used 3DS Max® software, maxscript being used for scripting and encoding. A genetic algorithm was scripted in maxscript. The final experiment is still in the process of being developed.

Environmental parameters were used to drive the evolutionary process. Passive solar evaluation was based on the Unified Day Degree method (UDD), which is embedded in maxscript.

Figure 2. A possible population

3.2 Initial Pattern

The initial pattern matches the definition of an elementary genome, that is to say the first individual (Figure 3.). Here, the user is initializing the procedure. The initial pattern is represented by a schematic geometric description, a sketched volume, a primary envelope containing constraints not yet explicitly described, such as the plot dimensions, the desired surfaces, the personal formal intention of the designer and the mental representation of the designer. For our first experiment, we used a simple box of fixed size.

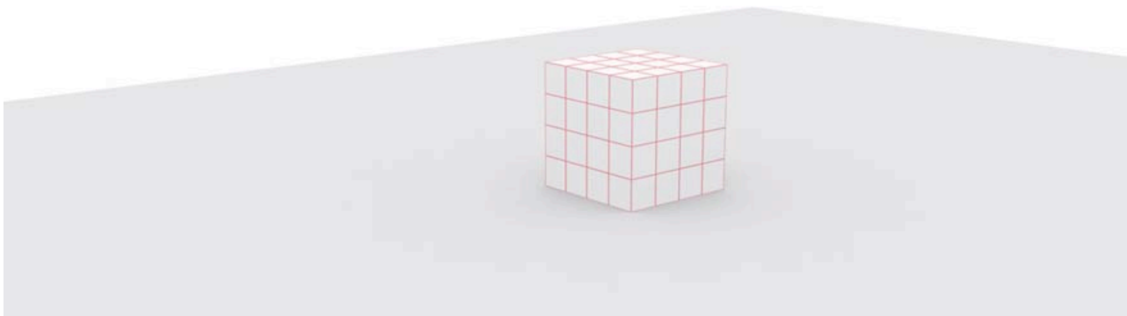

Figure 3. Initial pattern

The following three parameters played a key role in the solution: the overall dimensions, the overall form and the topological description (the number of segments, the number of vertices and the number of faces). They did not evolve during the evolutionary process.

It is clear that the first two parameters were highly influential. The influence of the third one was less direct, but we might suppose that the more defined segments there are, the more facets will be manipulated, and consequently the more continuous the deformation will be.

3.3 Shape exploration

The shape exploration was based on transformations through metamorphosis. Unitary operators were applied; in 3DS software they are called "modifiers". We used 5 main modifiers: bending, tapering, skewing, twisting and stretching. For example, the available parameters for tapering modifiers were: amount (between -100 and 100), curve (between -3 and 3), primary axis (x, y or z), effect axis (x, y or z) and gizmo rotation.

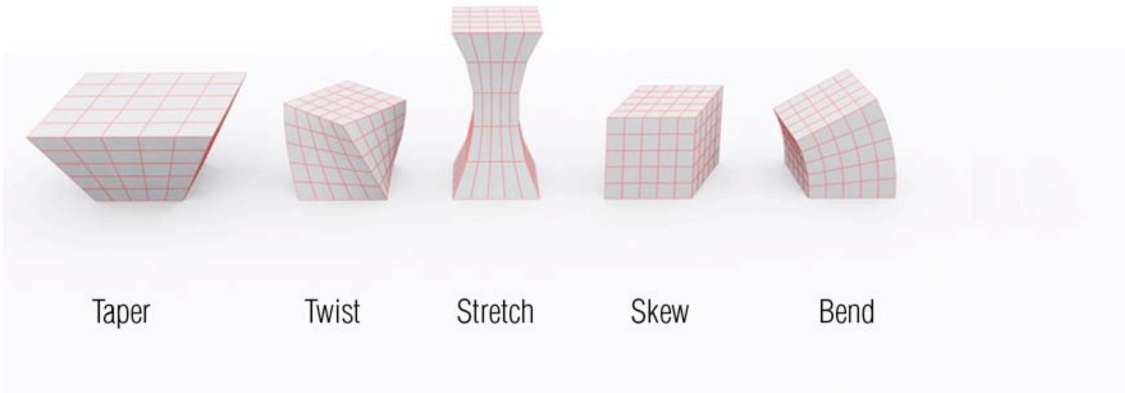

Figure 4. List of modifiers

The shape explorer takes the initial model as an input that triggers shape exploration, and automatically derives various shapes by simulating natural evolution through crossover and the mutation of genomes.

3.4 Material exploration

The material explorer makes it possible to modify the properties of each facet, regarding the opacity and the thermal resistance coefficient. These physical qualities are stored in an array, and to each facet the algorithm randomly assigns an index which refers to a physical quality. The polygons are labeled, and the evaluation engine then uses properties. This array is the “material chromosome”.

3.5 P-type and G-type

Using genetic algorithms, each individual is represented on the one hand by its P-type, i.e. its phenotype or geometric representation, and on the other hand its G-type, i.e. its genotype, or an encoded P-Type representation. The G-type symbolizes the individual's genome.

In our model, the G-type is based on a derivation approach. Rosenman (1997) makes a distinction between the transformation and the derivation approaches. The derivation approach is based on the use of rules. Executing the G-Type sequence generates a form, the G-Type representing a recipe rather than a blueprint. To go further, regarding the representation and structure of our G-Type, two main chromosomes made up our genome: one was associated with the physical properties of each facet, and we called it the “material chromosome”, while a second represented the description of the shape, and we called this the “shape chromosome” (Figure 5). At any time, the designer had the ability to edit the “shape chromosome” and modify its parameters.

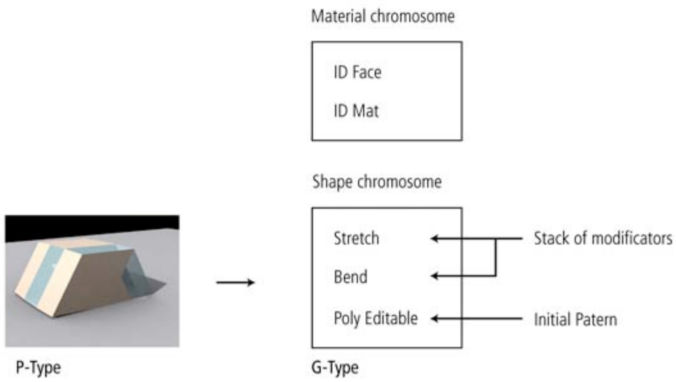

Figure 5. P-type and G-Type translations

3.6 Crossover and mutation

To begin with, a random population was defined. Each individual was evaluated by the UDD engine. Parents were treated two by two, and their chromosomes were cut at a random point, then reconnected (the process being known as "crossover"). One “modifier” of the first parent replaced a ”modifier” of the second parent. Material properties of the facets were combined in the same way. The ”children” formed a new generation of population, and were evaluated once more. The cycle continued until an acceptable result was attained, or a given limit of generations was reached.

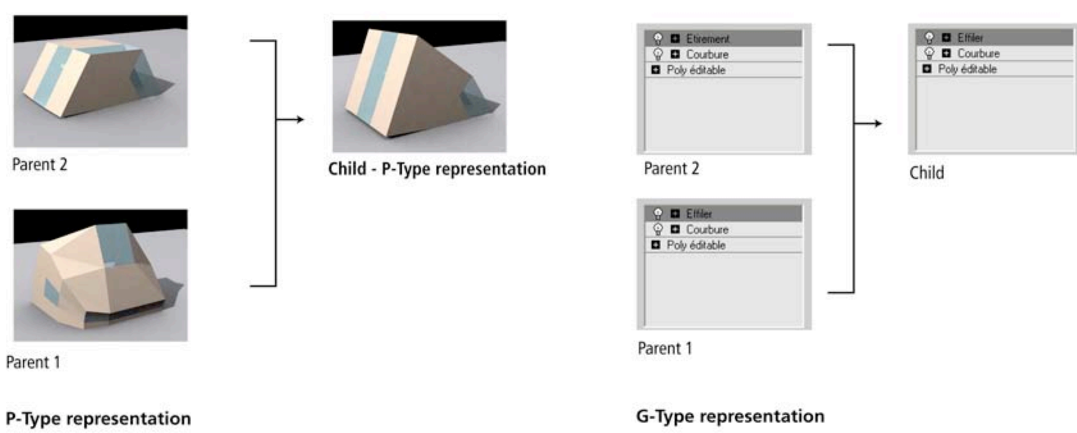

Figure 6. P-type and G-type representation of the shape chromosome crossover

Mutation mechanisms started from a selected individual, then randomly replaced certain parameters of each chromosome. The mutation was placed in a new generation for evaluation and selection.

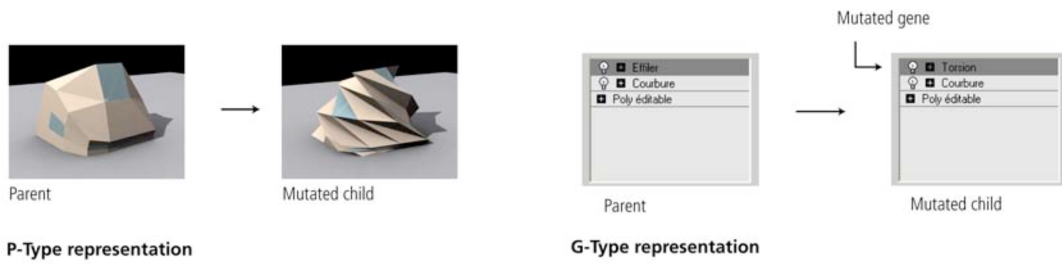

Figure 7. P-type and G-type representations of a shape chromosome mutation

Crossover and mutation can be carried out within a species ("intra-species"), or between species ("inter-species"). In the former case, we changed only the value of the modifier parameters; in the latter, we changed the entire modifier.

3.7 Individual evaluation

The main aim of a non-routine context is to generate suitable forms that are not necessarily optimal, but satisfy a range of customer, social, technical and designer requirements (Rosenman 97). In our example, the evolutionary process was intended to stimulate the creativity of the designer, and to suggest an optimal solution with regard to passive solar properties.

Our UDD engine evaluates passive solar qualities. Its mode of operation is based on the Unified Day Degree method (Figure 8.), which was selected because of the simplification of the problem it provides. At an early stage of the design, all the parameters were not known; some approximations were required. We focused on winter comfort and heating needs.

D = Ht.Dh(Ωa)

D : Heat loss from the building [kWh/year]
Ht : Loss coefficient of the building [W/K]
Dh(Ωa) : Value of degree-day
Ωa : ambient temperature of record

Ht = Henv + Hrev

Henv : loss of building envelope [W/K]
Hrev : loss by ventillation [W/K]

Henv = Σ(A.U)

A : Wall surface [m2]
U : Loss coefficient of surface [W/m2.K]
Hrev : Neglected

Heat loss (D) is offset by free inputs (AG).

AG = AI + AS

AG : Free inputs [W]
AI : Internal input [W]
AS : Solar input [W]

AS = E.Sv.c

E : Solar radiation function of tilt and orientation [W]
Sv : Glass surface [m2]
c : Transmission coefficient of the glazing

f = AG/D

f : Free input divided by heat loss

μ = f(f, inertia class)

B = D - μAG

B : heat needs [kWh], including free input.

Figure 8. The UDD method

The environmental parameters were stored in an array: solar radiation at a specific angle and orientation panel, external temperatures, internal inputs and inertial classes. Each individual in our population was rated according to heat needs. The lower the heat needs, the higher the individual was rated.

A subjective interaction was added to the evaluation process. The engine displayed the best models that had evolved over the different generations. The user could control the evolution process by exercising selection preferences. The evolutionary process could then be reiterated, based on this new initial pattern. The final evaluation of satisfactory design solutions was to some extent subjective, in that it involved an aesthetic or symbolic content.

3.8 First validation of our genetic algorithm

In order to validate our genetic algorithm, we carried out an initial experiment with a simplified fitness function. The parameters of our genetic algorithm were: the generation number, the elite number, the tournament size, the mutation rate and the population size. Figure 9 shows a pseudo code for the algorithm.

```
generation = 0;
initialize population;
while generation < max-generation
        evaluate fitness of population members
        for i from 1 to elites
                select best individual
        endfor
        for i from elites to population-size
                for j from 1 to tourmanentsize;
                        select best parents;
                endfor
                for k from elites to population-size*(1-mutationrate)
                        crossover parents -> child;
                endfor
                for k from population-size*(1-mutationrate) to population-size
                        mutate parent->child;
                endfor
                insert child into next generation's population;
        endfor;
        update current population
        generation++;
endwhile;
```

Figure 9. Pseudo code for the genetic algorithm

The simplified fitness function was based on the total surface value of the envelope for each individual, which we wanted to minimize. The value of the surface matched the evaluation rank. The lower the rank, the higher the rating. The initial pattern was a simple box of fixed size.

Figure 10 follows the populations through 20 generations, the fourth best parents of each generation being selected, then used to build the elite population. The exploration of the shape uses the Taper modifier to bring about evolution in the population; crossover and mutation are "intra-species". Figure 11 shows the evolution of the simplified fitness function through 100 generations. Four series are studied as a function of the genetic algorithm's parameters. Each configuration can give an optimal solution, but at different levels of efficiency.

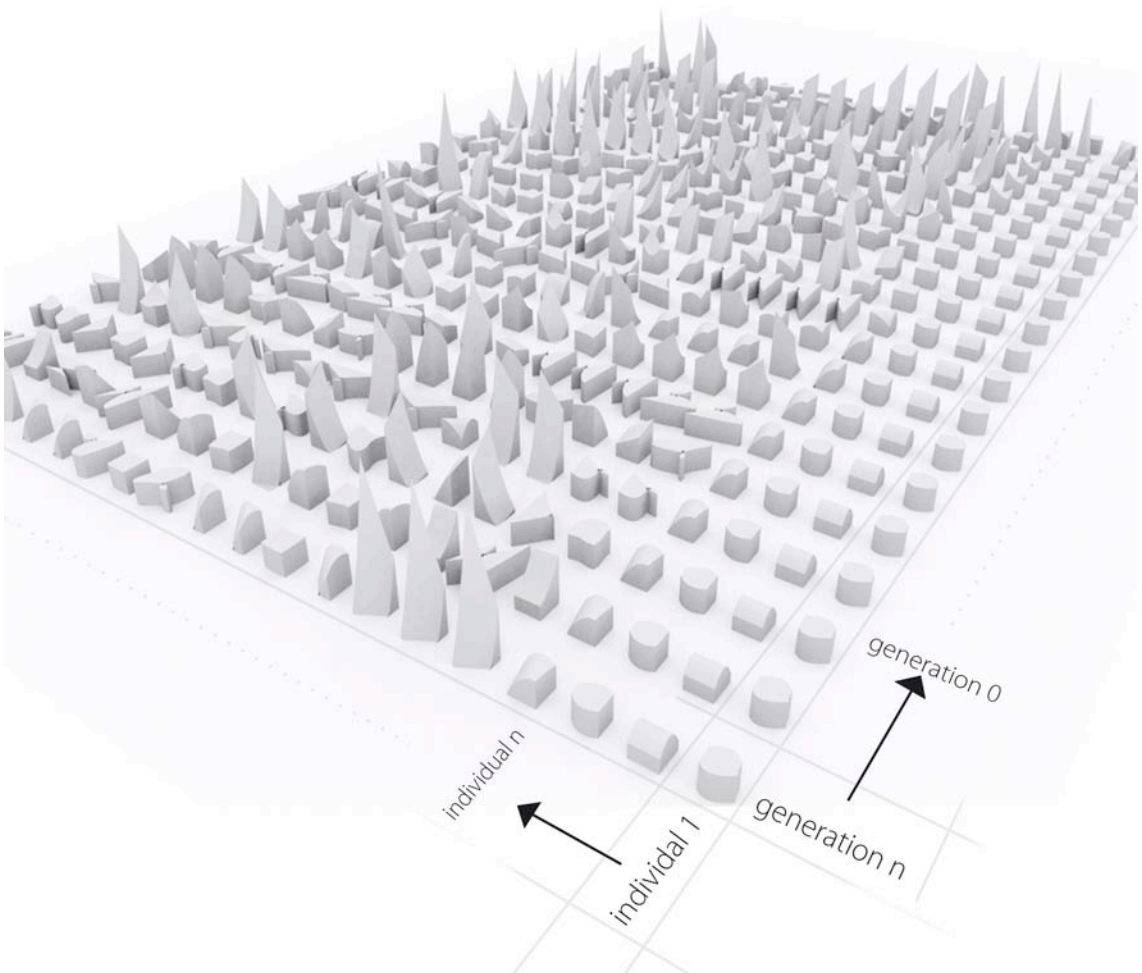

Figure 10. A population over 20 generations

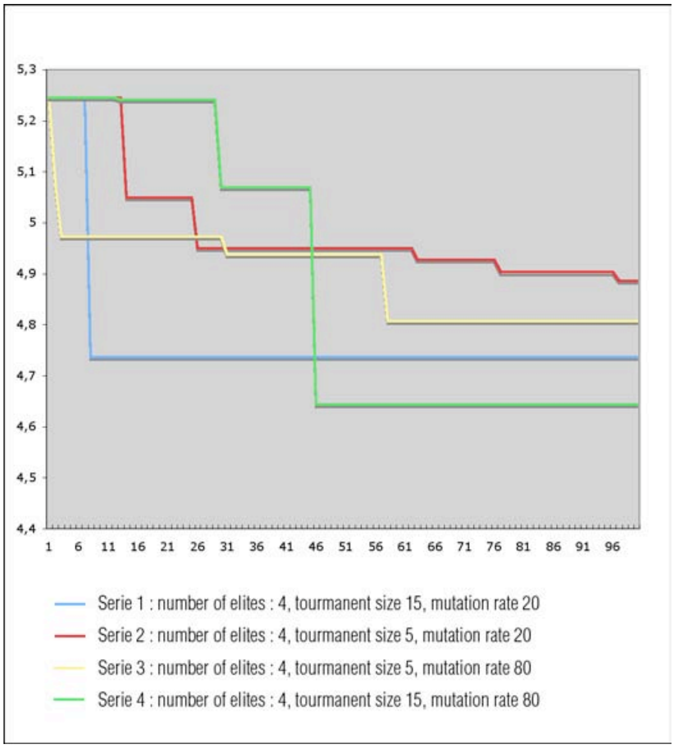

Figure 11. Fitness convergence

4 Conclusion

Having validated the experimental protocol and the efficiency of the genetic algorithm, we are now working on the implementation of the final fitness function. The exploration of the shape takes place correctly, but we still have to define limiting values for each of the modifier parameters, in order to preserve the integrity of the shape. Questions about interactions between the user and the digital tool also have to be examined. An overall representation of all the different generations can be implemented through a phylogenetic tree presentation. This should provide an understanding of filiation relationships. We might also postulate interactive functionalities going back to past generations, and the interactive selection of a grandparent for a new initial pattern.

Environmental parameters can be changed, and design exploration can begin from other locations. Extreme conditions could be used to evaluate the influence of environmental parameters.

Natural processes can be used to develop digital tools applicable to the creative design process, and to generate unexpected solutions. The use of evolutionary processes marks a change in the way computers are used: designers no longer work on single solutions, but rather on processes, which have to be made explicit in order to be encoded. The designer's creativity and imagination are not restricted by indirect manipulations, or by the particular information imposed by univocal descriptions of models. The design process changes with the cognitive level.

References

Caldas, L. G., Norford, L. K. (1999). Genetic Algorithms tool for optimization of envelopes and the design and control HVAC systems, Journal of solar energy engineering.

Chink, F. D. K. (1979). Architecture: Form, Space & Order. New York, Van Nostrand Reinhomds.

Bentley, PJ. (1999). An introduction to evolutionary design by computers, in PJ Bentley (ed.) Evolutionary Design by Computers, Morgan Kaufmann Publishers, San Francisco, CA.

Frazer, J., Frazer, J., Liu, X., Tang, M. Janssen, P. (2002). Generative and evolutionary techniques for building envelope design, Generative 2002.

Gero, J., S., and Maher, M., L. (1998). Adaptating evolutionary computing for exploration in creative designing; Computation models of creative design IV.

Malkawi Ali M. et Al (2003). Performance-bases design evolution: The use of genetic algorithms and CFD, Building Simulation, Eindhoven, Nethrlands.

Nishino, H., Takagi, H., CHO, S. and UTSUMIYA, K. (2001). A 3D modeling system for creative design. ICOIN 15, Japan.

Rosenman, M., A. (1997). An exploration into evolutionary models for non-routine design, Artificial Intelligence in Engineering, 11(3):287-293.

Soddu, C. (2004). Agenia, a mother tougue in infinitive variations, 1st International Conference on Design Computing and Cognition.

Todd, S. and Latham, W. (1992). Evolutionary art & computer graphics. Academic Press.

Wetzel JP., Belblidia S. and Bignon JC. (2006). A proposition for parametric morpho-semantic operators to assist architectural conception at the drafting stage. Colloque - CGIV 06, third International Conference on Computer Graphics, Imaging and Visualization, 26-28 juillet 2006, Sydney, Australie.