Evolutive design of car silhouettes using an interactive genetic algorithm

François Cluzel, Bernard Yannou, Markus Dihlmann

Technical Report

Evolutive design of car silhouettes using an interactive genetic Algorithm

François Cluzel, Bernard Yannou, Markus Dihlmann

Laboratoire Génie Industriel, Ecole Centrale Paris
Chatenay-Malabry, France
Tel: +33 1 41 13 15 21 (Bernard Yannou)
E-mail: francois.cluzel@ecp.fr, bernard.yannou@ecp.fr

November 2010

Abstract

So as to create innovative car silhouettes, we propose in this technical report a model based on an Interactive Genetic Algorithm using an encoding of a design solution by a Fourier analysis approach. This model permits the designer to browse through generations of car profiles from an initial population of existing silhouettes. By qualitatively assessing each individual, the user converge towards solutions complying with his/her requirements and so potentially create novelty. We describe here tests for assessing the efficiency of this innovative design platform. These tests are based on a similarity matrix, a similarity measure being the perceived distance between two cars silhouettes. The results show a really satisfactory behavior of the model and open perspectives thanks to its flexible and extensible aspects.

Key-words: interactive genetic algorithm, evolutionary design, shape design, subjective evaluation, user tests, car profile.

Acknowledgments

We gratefully thank François Bleibel, Nicolas Cordier, Gilles Foinet and Thomas Ricatte for their contributions to this work.

François Cluzel, Bernard Yannou & Markus Dilhmann
Ecole Centrale Paris **November 2010**

Technical Report

Evolutive design of car silhouettes using an interactive genetic Algorithm

Table of contents

1. Introduction

Design is an engineering activity for creating new and innovative structures and shapes. Finding a new shape and style for an object can be seen as a profound human and sometimes artistic refinement process. Indeed, starting from an initial idea, the style designers continuously refine it through multiple sketches and drawings using their intuition and perception of their own production in a reflexive manner. Is it possible to help such style designers in their refinement process? Such an aiding tool should help him or her to explore more easily and systematically a large space of possible styles or shapes, and also to converge towards an ideal shape the designers could have more or less represented in their mind.

In the field of implementing this creative design process, Evolutionary Computation (EC) has become one of the primary approaches. A method in EC uses basically genetic algorithms (GA) [1,2], which were originally used to find solutions for complex optimization problems. For example, Poirson use Gas to optimize the design of brass musical instruments considering mathematical and perceptual objectives [3]. Taking the evolution in nature as paradigm, the GAs work on the basis of a population of individuals, where each individual represents a possible solution for the initial problem. The structure and the qualities of each individual are encoded in their genomes. Through recombination of these genomes the individuals can reproduce themselves and produce new individuals (solutions), while by a sort of natural selection the individuals who are not adapted to the environment (what is expected of their properties) are not selected for procreation. In this way, the individuals display better and better qualities over the generations. Interactive Genetic Algorithms (IGA, see [4,5]) represent a special class of GAs where a human (here, the style designer) is a key player embedded within the task of selection of individuals of a generation. IGAs are then particularly adapted to situations where it is impossible to explicitly express a preference function (the fitting function) on individuals or even when it is hard to qualify expected properties. This is typically the case with style designers.

A major difficulty when using GAs in automatic design systems is the encoding of the genome (see [4]), which means the way of coding the phenotype (physical structure) of the individual into the genotype (genome). Most systems use a direct encoding where geometrical dimensions and structures of the design object are directly represented in the genome (see for example [5]). When designing a bottle for example [6,7] or finding a design for cylinder shapes [8] the phenotype is represented in the genome by a sequence of geometrical parameters like radii, lengths and part locations. Consequently, the encoding is context dependent. Other works use tree structures [9] or shape grammars [10] to encode the genome. Kim and Cho [11] have used a set of predefined parts of clothes to find new designs in fashion by recombining these parts. But here the space of possible solutions is limited and we wonder, for all these methods, if an actual innovation results of these design processes.

In addition, all these systems are conceived for a given design domain. Implementing these methods in new fields of design is a difficult and time consuming process. However, a good design method should be applicable, as much as possible, on a large spectrum of situations.
In this paper, we first propose a method of encoding a 2D-closed-curve which is supposed to meet a desired style. This method can be applied to all possible objects represented by their 2D-silhouettes. For instance a car silhouette or profile is a primordial style feature of a car. Indeed, Cheutet [12] has shown that the character of a car profile is primarily expressed through a series of about ten lines (see Figure 1). Five of them: *hood line*, *windshield line*, *roof line*, *wheelbase line* and *wheel arch*, may be merged into a silhouette closed line. These lines, and especially the silhouette have been proved to have a strong determining influence on the car perception while embedding perceptual attributes such as: sportiveness, aggressivity or peacefulness, etc. In addition, it has been proved that the aesthetic aspects of a car amounts for 70% of purchase intents for customers [12].

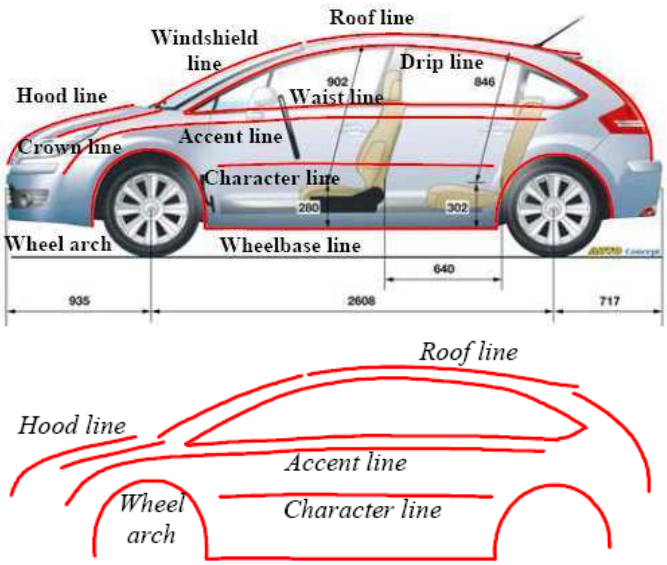


Figure 1. The main style lines of a Citroën car (from [12])

We propose in this technical report to detail a principle for encoding the genes of a car silhouette after a Fourier decomposition. Next, an Interactive GA (IGA) has been developed in defining a crossing-over operation between genes. The interactivity consists in letting a style designer qualitatively assessing individuals at each generation. In this manner, new innovative designs are expected to emerge by a balanced collaboration between an automatic process of design space exploration and the interaction of a designer. Finally, we provide measures and user tests for proving that innovation and surprise may emerge from this process. Indeed, we show that the initial population of individuals contains a sufficient richness of genes so as to be able to quickly converge towards a desired silhouette which is not an individual of this initial population.

In this way, innovative and new concepts are expected from the collaboration between an automatic process of design space exploration and the interaction of a designer. But validations are necessary. That is why we have proposed to use a similarity index. Whereas Petiot and Dagher use a manual similarity assessment method [13], the aim of this indicator is to automatically quantify the perceived distance between 2 individuals. We want to answer to the following questions:

- Are the users satisfied by our model?
- Are the results better with our model than without?
- Does our model really create novelty?
- Are the results different from a user to another?

We first propose an automatic test to ensure the ability of our algorithm to converge. But user tests are also essential to prove that innovation and surprise may emerge from this process. Thus we performed a subjective evaluation workshop to collect user data. Then the results of 2 main tests and 4 post-processing tests using the similarity index have been established. They show a very satisfactory behavior of the model in terms of convergence, diversity, dependence to the initial population, but they also show interesting results about the user perception.

The paper presents in section 2 the process of the Interactive Genetic Algorithm with the different operations required for encoding a car profile, generating an initial population and combining the genomes. In section 3, a similarity index is proposed to measure the perceived distance between two individuals. Section 4 deals with user and post-processing tests to ensure the validation of the model, whereas some reflections on the mutation operator are proposed in section 5. Finally, section 6 concludes on some forthcoming perspectives.

2. Model description

2.1. *2D closed curves modeling*

In this project, we have decided to represent a car by a 2D « side-view » closed curve. Mathematically, we define $\mathcal{M}$, the set of closed curves in $\mathbb{R}^2$. We define the quotient set $\mathcal{M}\backslash\mathcal{R}$ where $\mathcal{R}$ is the following equivalence relation : « Two curves of $\mathcal{M}$ are equivalent with respect to $\mathcal{R}$ if and only if they are similar through rotations, homotheties and translations ».

Numerically, there are different ways to describe an equivalence class (resampled curve, Fourier spectrum, wavelet coefficients…). The book [14] gives a quite exhaustive introduction to this field. One must keep in mind that the different representations are complementary and designate a unique object.
They could be especially useful for:

- Avoiding the apparition of « degenerated individuals » (aberrations are likely to be invisible for frequential representation) (improving « consistency »),
- The cross-over operations,
- The mutation operations.

Finding a way to efficiently represent the cars is an essential point in our study. This issue is closely related to the notion of ontology. Indeed, we need a formal representation of our knowledge in order to decide how to manipulate our objects while staying in the restricted field of car design. Thus we have to ensure that, for example, crossing two vehicles will create a new object that will still be an « acceptable vehicle ». The main principles of designing an efficient ontology have already been deeply investigated by a few researchers like Tom Gruber [15]. Gruber gives some important dimensions that one shall take into consideration when designing a new ontology:

- *Clarity:* In our case, this implies that our representation must be objective and does not depend on the context.
- *Consistency:* A very consistent ontology shouldn't be able to create « monsters » from valid rules. However in our case, too much consistency will constrain creativity.
- *Extensibility:* One should keep in mind that extensions are likely to be added in the future.
- *Weak coding distortion:* When we are representing a real object like a car, there's always some « distortion » due to the fact that our ontology cannot represent all sides of the object. An essential drawbacks of 2D descriptive methods (cf. infra) is their strong distortion.
- *Ontological complexity:* Our representation shouldn't be « over-elaborate ».

One should keep in mind that the main goal of ontologies is above all to conceptualize a « fixed » field of knowledge. A trade-off between the « quality » of the ontology and potential creativity is essential in our work. Table 1 gives some example of such ontologies.

	Completeness	**Robustness**	**Capacity to create "monsters"**	**Restrictivity (to one particular application area)**	**Adaptivity to genetic algorithms**
Shape grammars	++	++	++	-	--
Spatial contour	--	--	--	++	++
Fourier harmonics	--	++	--	++	++
Wavelets	--	++	-	++	++

Table 1. Comparison of different encoding methods

Many ontologies today are based on taxonomic relationships: entities like cars are classified in different families organized in a global hierarchical structure. Thus, a car can be described as a special realization of some taxonomic scheme. This method leads to generative models.

In this study, we decided to use a quite simplistic ontology that we will describe in the next part and that we can call « purely descriptive model ».

2.2. *The genome*

Concerning the encoding of a 2D-closed-curve, McGarva [16] has proposed its development into a Fourier series as a method for coding its phenotype. We have personally already used this theory in [17] for encoding a 2D-closed-curves into the five first Fourier harmonics of this decomposition. In that way, we have been able to build an Artificial Neural Network for synthesizing four-bar linkage mechanisms following targeted trajectories. This approach is not as rigid as the approach of parameterization for multiple reasons:

- This encoding is supposed to embrace a much vaster space of possible 2D-closed-curves – or 2D-silhouettes – than by a parameterization approach;
- All kind of shapes may be represented even with small details, that can be of the highest importance for provoking feelings and emotions;
- The encoding may be performed through a constant length of genotype, which simplifies a lot crucial GA stages such as the cross-over operation between parent individuals;
- Finally, the genes in our solution have proved to be narrowly associated to apparent characteristics which are primordial to converge after several generations to the ideal 2D shapes.

The Mac Garva's theory of Fourier decomposition of a closed curve [16] considers that the position of each point belonging to this curve can be expressed by a complex function in the complex plane:

$$z(t) = x(t) + i\,y(t) z(t) = x(t) + iy(t) \quad (1)$$

As *z(t)* is a closed curve, its function is periodic. The period is normalized with: *z(t+1)* = *z(t)*. This function *z(t)* can be developed into a Fourier series:

$$z(t) = \sum_{m=-\infty}^{\infty} a_m \exp(2\pi i m t) \quad (2)$$

where the complex Fourier coefficients can be calculated by this formula:

$$a_m = \int_0^1 z(t) \exp(-2\pi i m t)\, dt \quad (3)$$

Coefficient a_0 is called fundamental, a_1 and a_{-1} represent the first harmonic, a_2 and a_{-2} the second harmonic, etc.

As we will see later, the function *z(t)* is not known as an explicit function from the beginning. Instead, we assume that the curve has been initially defined by a set of successive points z_k (k=0,..,N) which belong to the curve. So, in order to calculate the a_m coefficients (3) we need a numeric approximation. We obtain this approximation by dividing the curve into N segments connecting each point with its successor. We call t_k the length of the curve between the first point z_0 and the point z_k. Under these conditions the integral can be calculated by the trapezium formula:

$$a_m = \sum_{k=0}^{N} \left(\frac{t_{k+1} - t_k}{2}\right) \left(z_{k+1} \exp(-2\pi i m t_{k+1}) + z_k \exp(-2\pi i m t_k)\right) \quad (4)$$

while z is a periodic function, ($z_{N+1} = z_0$).

The value of t_k is the ratio of the length of the curve to the point k and the total length of the curve.

$$t_k = \frac{L_k}{L} \;,\; L = \sum_{i=0}^{N} \sqrt{(x_i - x_{i+1})^2 + (y_i - y_{i+1})^2} \text{ and}$$

$$L_k = \sum_{i=0}^{k-1} \sqrt{(x_i - x_{i+1})^2 + (y_i - y_{i+1})^2} \quad (5)$$

where the total length L is the sum of the lengths of all segments, L_k is length from the origin to the current point, $x_{N+1} = x_0$ and $y_{N+1} = y_0$.

To construct the genome of an object, we develop its silhouette into a Fourier series and define the fundamental (the coefficient a_0) as gene number zero. The first harmonic (a_1, a_{-1}) will be called the first gene, the second harmonic the second gene, etc.

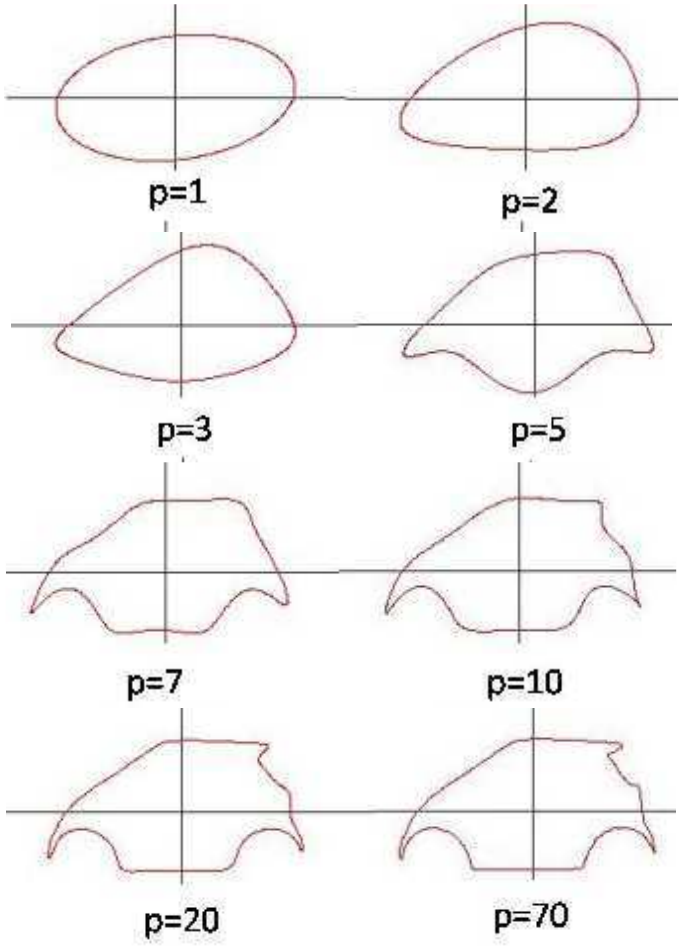


Figure 2. Decoding of a genome of a Smart car with different precisions

On the basis of the genome, the original shape of the individual can be reconstructed. Every point P_k with the coordinates (x_k, y_k) on the curve z^* which approximates the silhouette of the car, can be calculated by formula (6).

$$z^*(t_k) = x_k + iy_k = \sum_{m=-p}^{p} a_m \exp(2\pi i m t_k) \quad (6)$$

where t_k ($0 \le t_k \le 1$) is the position on the curve and p fixes the number of harmonics used for the decoding. When p equals 1 for example, we use one harmonic to reconstruct the silhouette of the car. The more harmonics used for the decoding the more precise will be the approximation to the original curve (as seen in figure 2). We call p the "precision" of decoding.

It can be easily proved that the first harmonic (the sole complex coefficient a_0) represents the coordinates of the centre of gravity of the curve in a complex plane. The second gene (a_1 and a_{-1}) contains the information defining an ellipse. The influence of the other genes cannot be illustrated easily. But we can say that the first genes influence the very basic structure and shape of the silhouette while the higher genes bring in the details of the

shape.

2.3. *The process of the interactive genetic algorithm*

The process of finding new design solutions can be divided into two phases (see Figure 3). During Phase 1 an initial population of individuals is created. Phase 2 consists of a loop where the user evaluates the current population and a genetic algorithm evolves the population respecting the evaluation of the user.

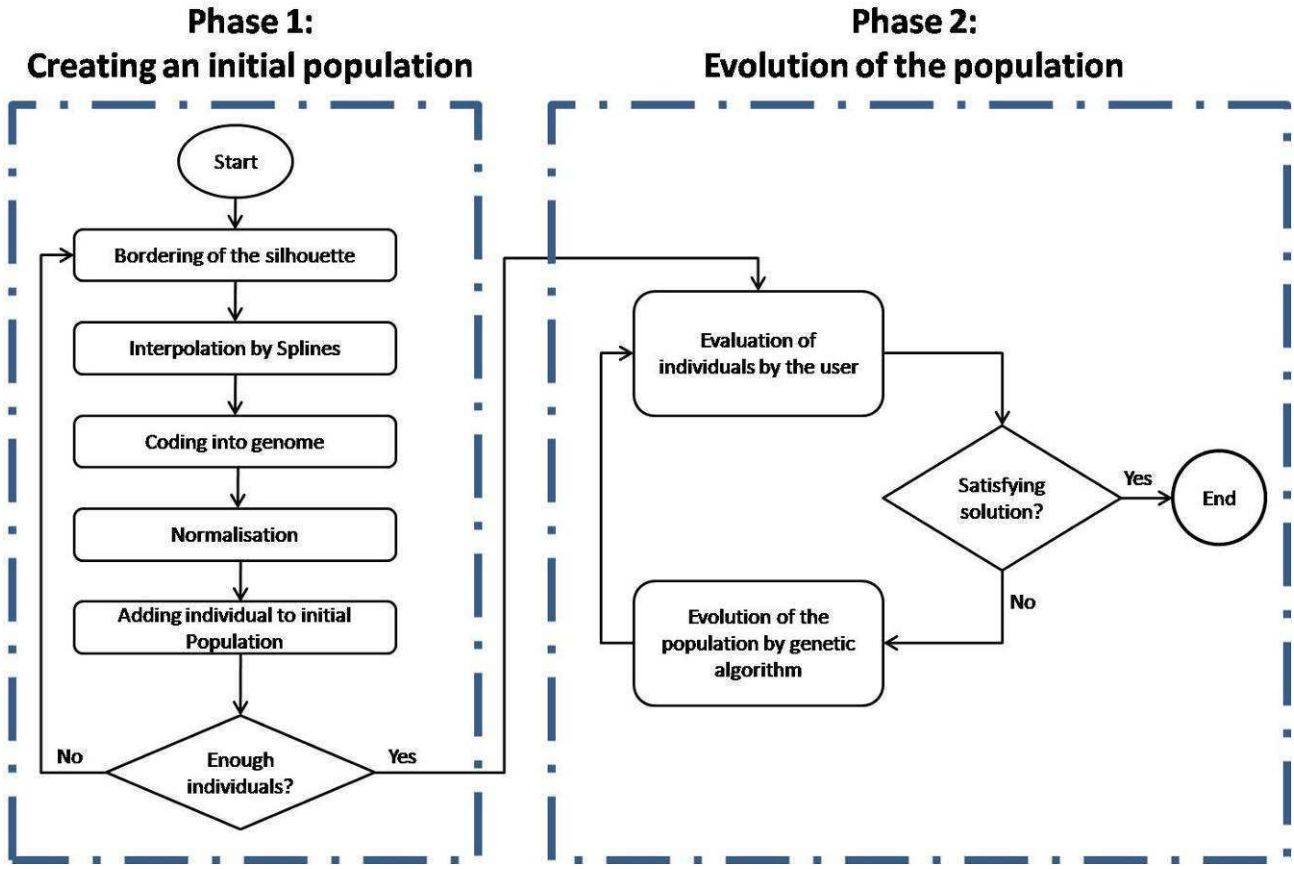


Figure 3. Diagrammatic plan of the IGA process

2.3.1. Phase 1: Creating an initial population

The genetic algorithm needs an initial population of individuals and their genetic code to start working. This initial population consists of silhouettes of 30 already existing car bodies. In order to easily sketch these silhouettes we programmed an interface in Java which allows drawing curves on a plain and coding them into a genome. To border a silhouette we display the image of an existing car in the background of the screen and draw a contour-chart around the car on the image by clicking on the screen. The result is a closed curve representing the silhouette of an existing car-body (see Figure 4). During bordering, a sufficient amount of points should be used to represent as many details as possible. When using a number M of 60 to 80 points per silhouette the result is satisfactory.

However this amount *M* of 60 to 80 points is not sufficient to calculate a genome which is precise enough to allow a highly detailed decoding into the phenotype. Consequently we need to augment the number of points on the curve by smooth interpolations. The curve produced by interpolation should be very close to the original curve and should be continuously derivable in each point. If the curve is not continuously derivable, the decoding from the genotype into the phenotype produces high-frequency oscillations and is therefore useless. We chose to solve this problem with bicubic splines linking three successive points (see Figure 5), because this method provides a curve which is very close to the original curve without producing oscillations (as it is the case when using polynomial interpolations

like Lagrange's interpolation formula). Within each spline, a given number of points are interpolated, leading to a total number of *N* points with *N*>*M*.

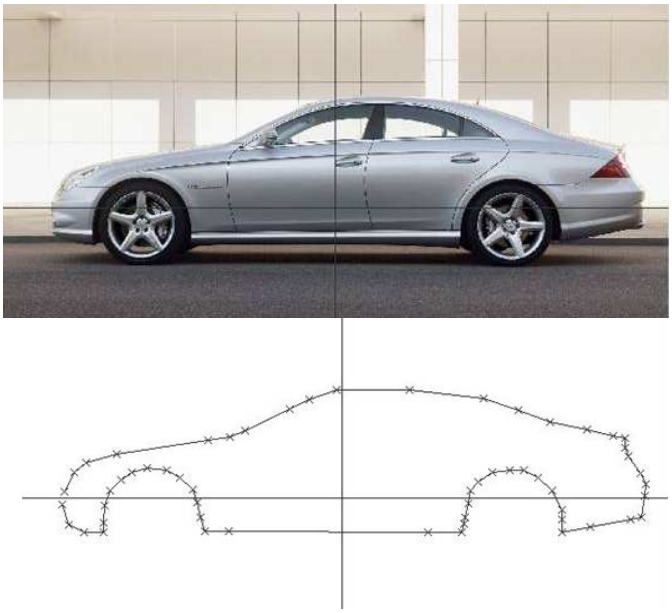


Figure 4. After bordering we obtain a closed curve representing the car silhouette of an existing car

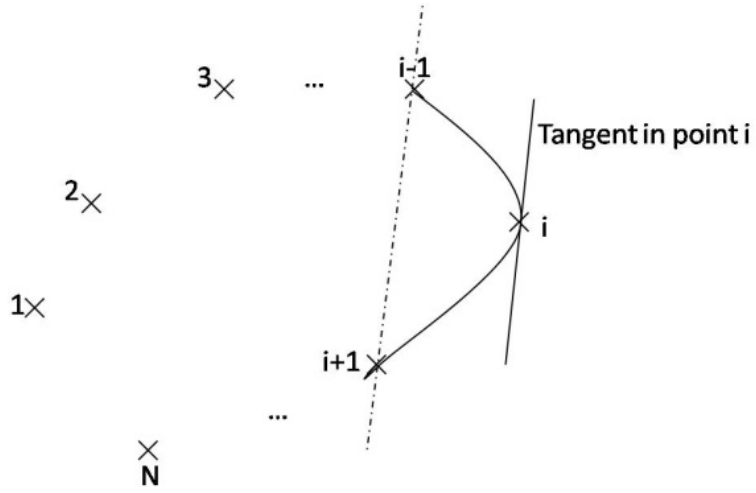


Figure 5. The tangent of the spline at point *i* is parallel to the line passing by points *i-1* and *i+1*

Taking care of the quality of the encoding amounts to find a satisfactory balance between the number *N* of points on the curve used for coding and the number *p* of harmonics used when decoding the genome into a curve. The number *p* of harmonics used for decoding has an influence on the production of details. The more harmonics used for the decoding the more precise will be the approximation to the original curve. The number *N* of points on the curve used for coding the genome has an influence on the precision of the Fourier coefficients. This is due to the fact that we use the trapezium formula in (4) to approximate the integral during the calculation of the coefficients. We achieved numerous trials of (1) bordering a silhouette, (2) interpolating with *N* points, (3) encoding with *p* harmonics, (4) decoding, for finally comparing the initial and the resulting silhouettes (see Figure 6). A qualitative design of experiments has been carried out (see Figure 7) with *p* varying from 80 to 2000 and *N* varying from 5 to 200. We clearly noticed that if *p* is too low, the coding-decoding sequence – visually - fails to accurately represent the initial silhouette. In addition, for a given number *p*, there is a minimal number of points *N* beyond which the reconstructed

curve displays strong oscillations (see such oscillations in Figure 6). In definitive, we found out that a satisfactory choice was achieved with a genome size of 71 and a number *N* of approximately 1500 points for the interpolation since both initial and resulting silhouettes were visually identical.

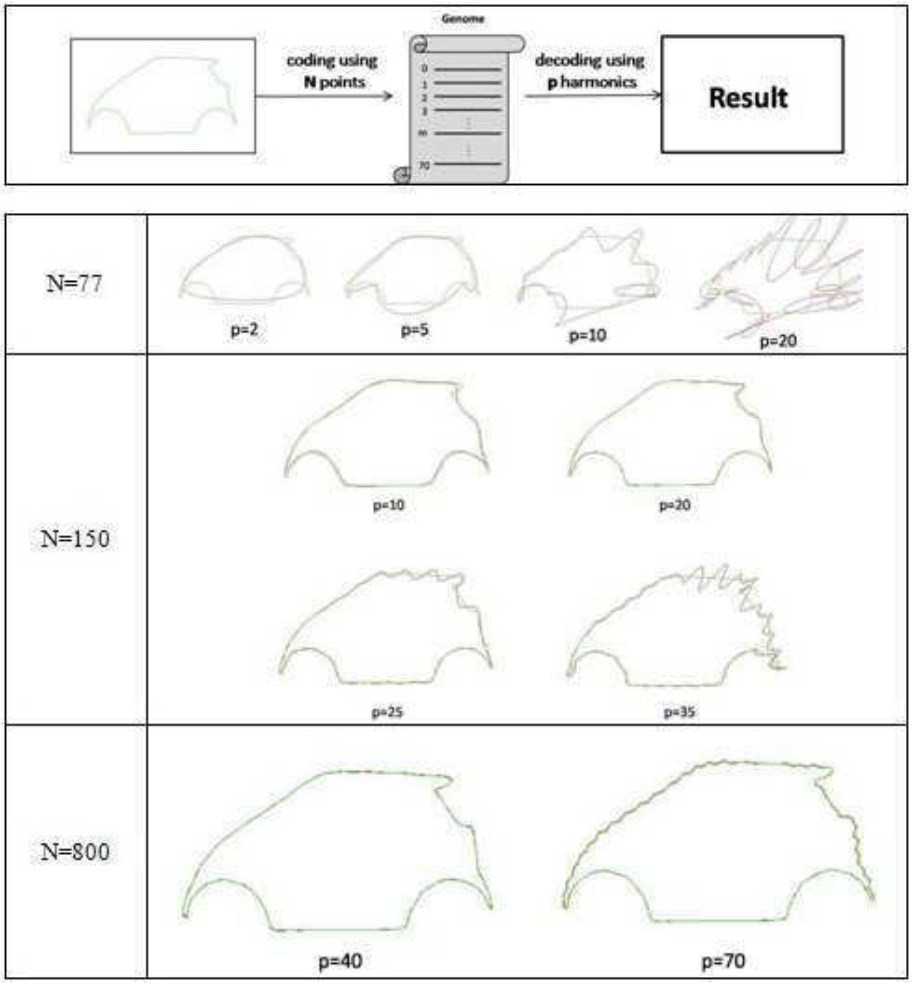


Figure 6. Comparison of silhouettes after interpolation with N points, encoding with p harmonics and decoding

A last operation of normalization is necessary to the genomes so that the phenotypes – silhouettes – be independent of a particular location, size or rotation but be compared uniquely in terms of their shape. The coefficient a_0 is simply set to 0 to fix the centre of gravity of all individuals at the origin of the representation plane. The invariance by rotation is useless because car silhouettes of the initial population are sketched horizontally and the next generations turn out to stay horizontal. McGarva [16] proposes to normalize the size of the curve in setting to 1 the small axis of the ellipse defined by harmonics 2. It would amount in our case to fix to a constant height the car silhouettes which is not fair for short cars. We prefer to have a surface area invariance instead. The calculus is then a bit more sophisticated but simply consists in dividing all coefficients a_m by a value function of $|a_1|$ and $|a_{-1}|$ (formula not detailed here).

p\N	80	100	200	500	700	1000	1200	1500	2000
5	I	I	I	I	I	I	I	I	I
7	I	I	I	I	I	I	I	I	I
10	SO	I	I	I	I	I	I	I	I
15	SO	I	I	I	I	I	I	I	I
20	SO	O	G	G	G	G	G	G	G
30	SO	O	G	G	G	G	G	G	G
40	SO	SO	O	O	GG	GG	GG	GG	GG
50	SO	SO	O	O	GG	GG	GG	GG	GG

55	SO	SO	SO	O	O	GG	GG	GG	GG
60	SO	SO	SO	SO	O	O	GG	GG	GG
70	SO	SO	SO	SO	O	O	O	**GGG**	GG
80	SO	SO	SO	SO	O	O	O	GG	GG
90	SO	SO	SO	SO	SO	O	O	O	GG
100	SO	SO	SO	SO	SO	O	O	O	GG
120	SO	SO	SO	SO	SO	O	O	O	GG
140	SO	SO	SO	SO	SO	O	O	O	GG
170								O	GG
200								O	GG

Table 2. The design of experiments carried out for finding an ideal (p, N). Initial and reconstructed silhouettes are visually compared to result in subjective assessments: I – inaccurate, O – oscillations, SO – strong oscillations, G – Good result, GG – very good result

2.3.2. Phase 2: Evolution of the population

We use an interactive genetic algorithm to evolve the population and create innovation. As Kelly says, "by using IGAs we hope to allow designers to enhance their creativity through design space exploration" [5]. The individuals can reproduce among themselves and produce in this way new solutions.

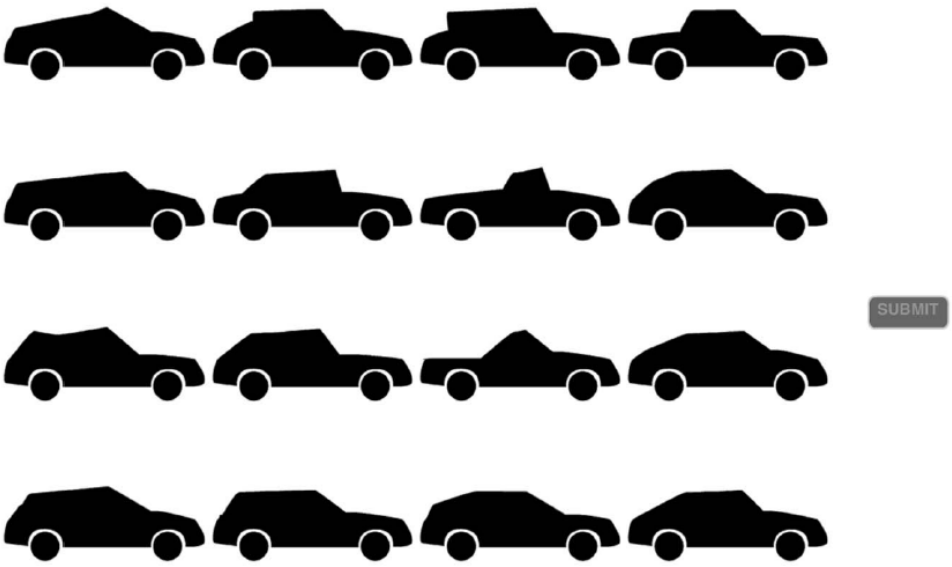


Figure 7. Example of IGA interface applied on car silhouettes, taken from [5]

In our case the genetic algorithm handles a population of individuals where each individual represents a possible design for a car body silhouette. A fitness value is assigned to each individual by the user. Consequently the fitness value f is a number between 0 and 6 according to the grade given by the user via an interface. The interface developed (see Figure 8) displays six individuals at a time and the user can browse through all the individuals of a population. The user is supposed to evaluate all the individuals of a population on a scale from 0 to 6, where 0 is the worst and six the best evaluation. This fitness decides if an individual has a good chance to reproduce and create children. Furthermore it influences the chance of an individual to survive and to live on in the next generation. This development is reached by applying the following genetic operators to the population:

- *Selection*: decides which individuals will reproduce and create children.
- *Crossover*: builds a child's genome from two parent genomes.
- *Mutation*: changes in a random way a genome after the crossover.
- *Killing*: decides which individuals from the parents' population will survive in the new generation.

We decided to adopt some conventional choices in term of selection and killing operators

and to propose an original crossover operator. First, apart the initial population of 30 individuals, we have fixed the number of individuals to 100 at each generation. We chose a turnover rate of 0.7, meaning that, for a coming generation, 30 individuals are kept from the previous one and 70 children are generated. In this way we do not lose potential good design solutions. The probability for an individual to be selected to be a parent is proportional to its fitness value (between 0 and 6). After choosing two individuals from the parents' population, their genomes are combined into the genome of a child by applying the crossover and the mutation operators. Afterwards the two individuals are re-put into the parents' population. Indeed, an individual can be selected more than once by the selection operator.

We envisaged several possibilities to crossover the two genomes of parents into the one of the child. For instance, we envisaged a "Two-Part-Crossover"-method which seemed promising at the beginning. It consisted in choosing randomly a crossover point *X*, where *X* is a number between 2 and 69. The child's genome was built by the first *X* genes from the genome of parent A and the last (70-*X*) genes from the genome of parent B. This method produced innovative designs for car silhouettes and few useless forms. However the method didn't produce stable results over the generations. This means that after some generations the car silhouettes were useless because they lost the tires or began to oscillate.

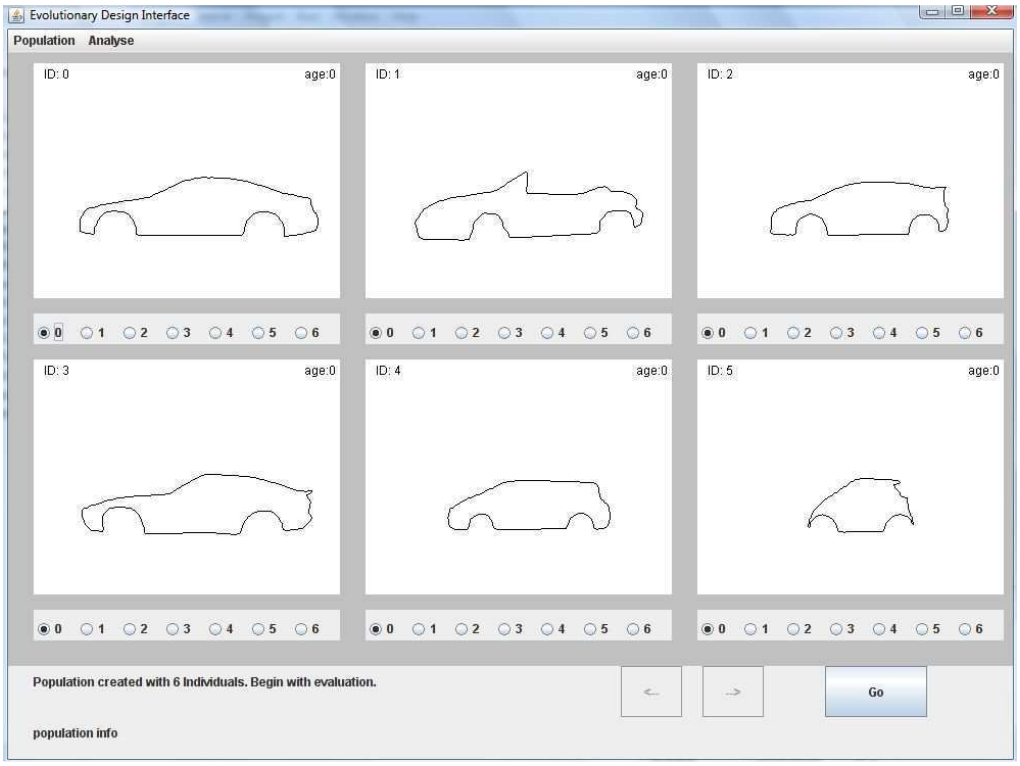


Figure 8. The User Interface for the designer evaluation showing 6 individuals of a larger population. The designer can browse the individuals by clicking on the arrow buttons

The good idea is to operate a weighted average between the gene values of the two parents to build the genome of the child. A crossover weight W is chosen randomly between 0 and 100. A new gene g^* is formed by calculating the weighted average of the genes $g_{m,1}$ and $g_{m,2}$ of the parents after formula (7).

$$g_m^* = \frac{W g_{m,1} + (100 - W) g_{m,2}}{100} \tag{7}$$

In function of the weight *W* we obtain different new design solutions which continuously interpolate a silhouette between the two parents' silhouettes (see Figure 10). The advantage of this method is the fact that a child resembles a lot to its parents and that it produces almost no useless car solutions (the tires keep their rounded shapes). The disadvantage is

the relatively small explored space of possible solutions. In consequence, the population of design solutions has a tendency to converge rapidly. To enlarge the space of possible solutions we must apply a mutation operator (not detailed here).

The killing operator is applied to the original population and kills at first all the individuals who have a fitness of 0. These individuals are considered totally useless or totally non-satisfactory and shall no more contribute to the evolution of the population. All the other individuals have a chance to survive. The individuals to be killed are chosen by an inverse roulette wheel method. That means that the probability pk_i for an individual to be killed can be expressed by formula (8).

$$pk_i = \frac{(7-f_i)}{\sum_{i=1}^{N^*}(7-f_i)} \tag{8}$$

where f_i is the fitness of individual i and N^* is the number of individuals in the population who have not been evaluated with a fitness of 0.

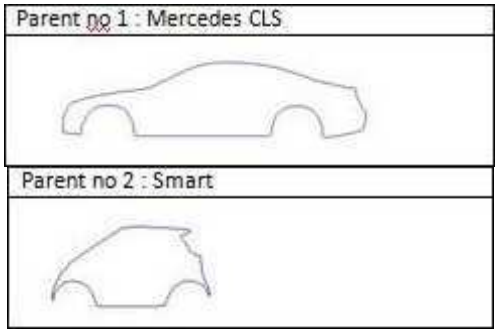


Figure 9. Two parent individuals

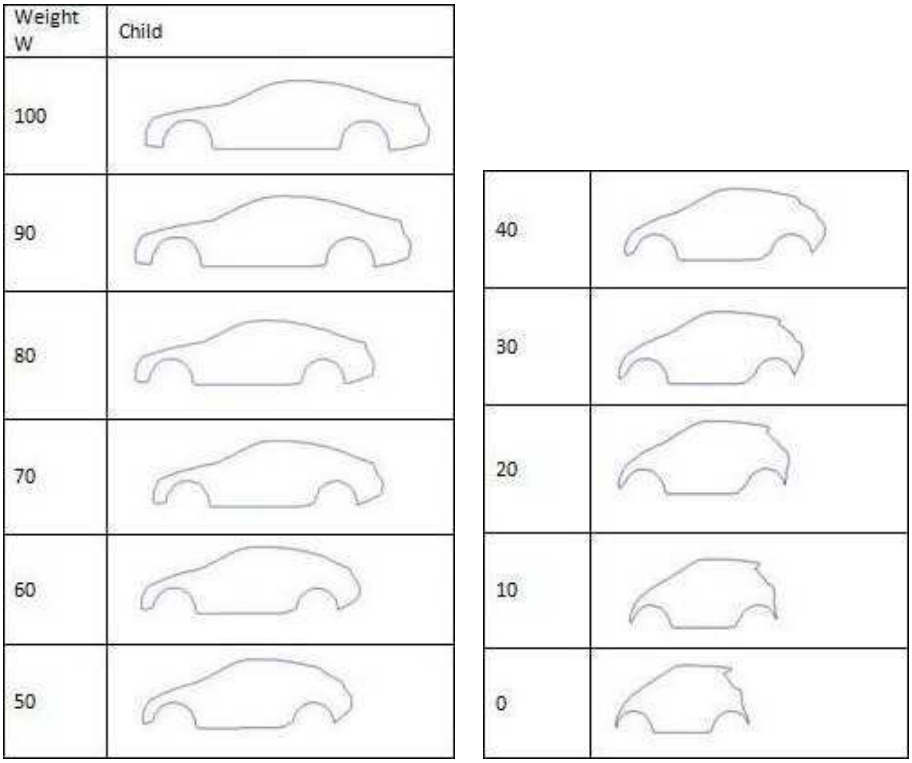


Figure 10. Results of a weighted average crossover between the genes of the two parents of Figure 9, using different weights W

3. Similarity index

Is our system really capable to produce innovation and novelty? Is it possible for a user to design with the help of our system a new car body silhouette which was not part of the initial population?

To answer these questions, it could be useful to have a tool that permits to automatically measure the perceived difference between two car silhouettes and to prove that two car silhouettes are really close or not. So we propose to create a similarity index. Based on user assessments, it would permit to easily measure the difference between two car profiles. It would be a helpful tool to realize tests and prove results. The similarity index is an essential point in this study. We can quote two important criteria for the design of the index:

- It must be mathematically founded.
- The calibration process must be stable and user-friendly.

The objectiveness issue is not a basic criterion: each designer can have its own conception of distance but it must in all cases remain consistent (that what we mean by the stability of the calibration process).

We propose here a description of the process to get this similarity index. Two different ways have been tested. The first is based on an exponential formula, and the second uses weights associated to genes.

3.1. *Mathematic definition*

We first define *D(k,l)* the distance between two genomes G_k and G_l. As the modifications on the ten first genes only are significant (modifications on other genes do not change anything on the car profile visual perception), the sum only consider those ten genes. And *D(k,l)* is:

$$D(k,l) = \sum_{m=1}^{10} \alpha(m) \parallel g_{k,m} - g_{l,m} \parallel^2 \tag{8}$$

The factor $\alpha(m)$ is a weighting factor which should give more importance to some genes according to their participation in the modification of the silhouettes. Here $g_{k,m}$ is gene number *m* from genome *k* and $g_{l,m}$ is gene number *m* from genome *l*. One gene consists of two harmonics, called a_m and a_{-m}, which are complex numbers. So they can be written as: $a_m = u_m + i.v_m$.

Then we define:

$$\parallel g_{k,m} - g_{l,m} \parallel^2 = \frac{(u_{k,m}-u_{l,m})^2}{(u_{max,m}-u_{min,m})^2} + \frac{(u_{k,-m}-u_{l,-m})^2}{(u_{max,-m}-u_{min,-m})^2} + \frac{(v_{k,m}-v_{l,m})^2}{(v_{max,m}-v_{min,m})^2} + \frac{(v_{k,-m}-v_{l,-m})^2}{(v_{max,-m}-v_{min,-m})^2} \tag{9}$$

where $u_{max,k}$ and $u_{min,k}$ (respectively $v_{max,k}$ and $v_{min,k}$) are the maximal and the minimal values of u_k and v_k on the whole initial population.

And we finally define the similarity index between two genomes *k* and *l* as:

$$SimInd\ (k,l) = \frac{100}{1+D(k,l)}\% = \frac{100}{1+\sum_{m=1}^{10} \alpha(m) \parallel g_{k,m}-g_{l,m} \parallel^2}\% \tag{10}$$

So with this definition, the similarity index in included between 0 and 100%, where 100% means that the two individuals are identical.

We now have to define the factor series $\alpha(m)$.

3.2. *Exponential form*

In this section we assume that $\alpha(m)$ can be written as an exponential expression which gives

more importance to the first genes than to higher order genes, because a modification of the first genes impacts more the car silhouette than a modification of the last ones. So $\alpha(m)$ is expressed as:

$$\alpha(m) = a\, e^{bm} \tag{11}$$

where *a* and *b* are two constant terms. So *D(k,l)* becomes:

$$D(k,l) = \sum_{m=1}^{10} a\, e^{bm} \parallel g_{k,m} - g_{l,m} \parallel^2 = a \times \sum_{m=1}^{10} e^{bm} \parallel g_{k,m} - g_{l,m} \parallel^2 \tag{12}$$

But now we need to find significant values of *a* and *b*.

We propose the following process to measure *b*:

- Choose a genome, which is copied 3 times: *G0*, *G1*, *G2*.
- Choose a gene *i* in *G1* (better with a low weight to be significant): $g1_i$.
- Choose a gene *j* ($i \neq j$) in *G2* (better with a low weight to be significant): $g2_j$.
- Modify the gene $g1_i$ of *G1* in an arbitrary way.
- Modify the gene $g2_j$ of *G2* in such a way that there is an *iso-similarity* (defined below) between *G0* and *G1* on the one hand, and *G0* and *G2* on the other hand.

We define the *iso-similarity* in this way:

> Two couples of car silhouettes are *iso-similar* if the perceived level of similarity is the same for the two couples. For example, it would mean here that the level of similarity is the same between *G0* and *G1*, and between *G0* and *G2*. Practically, it means that the user has to modify the gene $g2_j$ until the level of perceived similarity becomes the same between *G0* and *G1* as between *G0* and *G2*. In that way, *G2* and *G1* are not identical, but their level of similarity according to *G0* is the same.

Then we can write the following equality:

$$\alpha(i) \times \parallel g0_i - g1_i \parallel^2 = \alpha(j) \times \parallel g0_j - g2_j \parallel^2 \tag{13}$$

And *b* is:

$$b = \frac{1}{(j-i)} \times \ln \frac{\|g0_i - g1_i\|^2}{\|g0_j - g2_j\|^2} \tag{14}$$

By making *n* times this tests with different car profiles and different users, we get *n* different *b* values. The final value of *b* adopted is the average.

Level of similarity	Value of similarity index
0	5%
1	30%
2	50%
3	65%
4	80%
5	90%
6	100%

Table 3. Scale of similarity for user assessments

The next step consists in measuring *a*. We propose the following process: for each of the previous comparison (between *G0*, *G1* and *G2*), the user defines the level of similarity ("the similarity between G0 and G1 on the one hand and G0 and G2 on the other hand is 70%" for example). As it is very hard to express such a value, we propose to work with a 7 degrees

scale; the user just chooses the level of similarity in Table 3.

So then it is possible to write:

$$x\% = \frac{100}{1+a\times\sum_{m=1}^{70} e^{bm}\|g0_m - g1_m\|^2} \tag{15}$$

where *b* is the average value of the previous tests. And *a* is:

$$a = \frac{1}{\sum_{m=1}^{70} e^{bm}\|g0_m - g1_m\|^2} \times \left(\frac{100}{x\%} - 1\right) \tag{16}$$

We also obtain 2*n* different *a*. The average value is acceptable if the standard deviation is low. The calculation of the similarity index is now completed.

Practically, we use a Java interface (see Figure 11) that permits to follow the processes described above by loading a population.

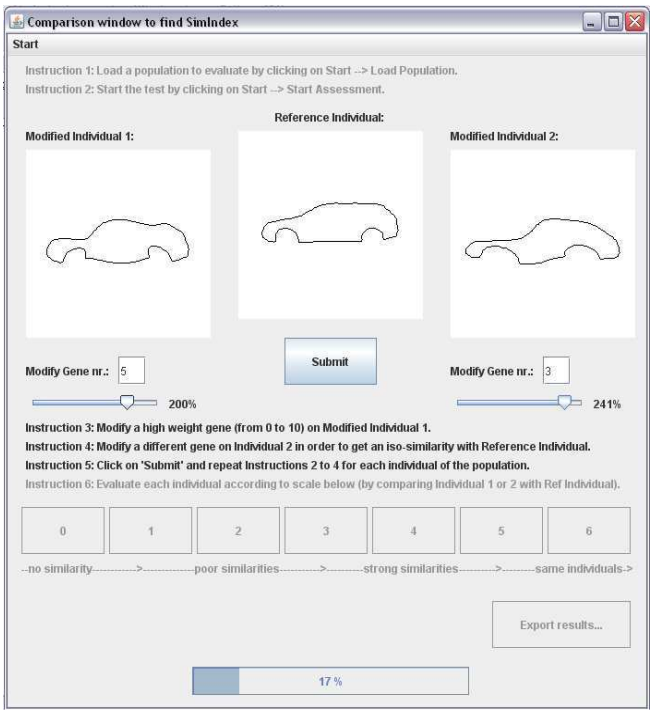


Figure 11. Java interface to calculate the parameters *a* and *b*

3.3. *Weighted form*

In the previous section, we assumed that $\alpha(m)$ should give more importance to the first genes that to the last ones and can be written as an exponential expression. But some experiments showed us that this assessment could be wrong when using the ten first genes only: the importance of these genes is not necessarily relative to their rank. Each $\alpha(m)$ is now associated to a given weight p_m.

So *D(k,l)* becomes:

$$D(k,l) = \sum_{m=1}^{10} p_m \parallel g_{k,m} - g_{l,m} \parallel^2 \tag{17}$$

To obtain the weights values, we follow the same process as with the exponential form. And we finally obtain the following expression:

$$p_i = \frac{\|g0_j - g2_j\|^2}{\|g0_i - g1_i\|^2}.p_j \tag{18}$$

By performing this test *n* times (*n>10*) with different values for *i* and *j* (to cover all the ten first genes), we have a system of *n* equations, that can easily be resolved with the *logarithmic least square* method.

The tests are performed with a modified version of the Java interface described in the previous section.

3.4. *Performance comparison and choice of the index*

To find out the best version of the similarity index, we perform the tests described in Figure 12. The same 30 car silhouettes are used to build the two similarity indexes. After that, the same 10 car silhouettes are used to validate it and permit to choose the best one. We obtain 2 similarity matrices, and we ask the users to assess the same 20 profiles in a third matrix, according to the scale of the table. These three matrices permit to calculate the RMSE (Root Mean Squared Error) associated to the two cases.

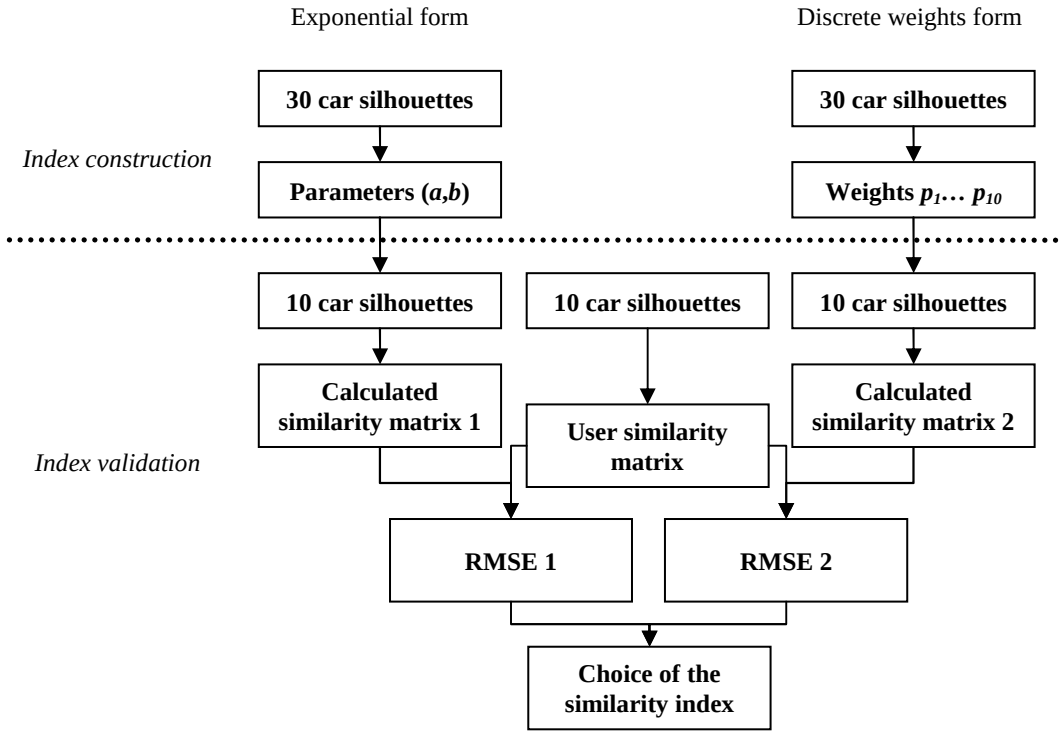


Figure 12. Choice of the similarity index type

We obtain with the exponential form of the index an average RMSE of 7.04, and the value is 31.82 with the discrete weights form. So we choose the exponential form for the similarity index. We can now perform user tests to validate our model thanks to this similarity index.

4. Validation tests

To validate our model we first performed simple user tests without the similarity index, based on Kim and Cho works [11].

4.1. *First tests*

4.1.1. Tests construction

Kim and Cho worked in 2000 on fashion design with an Interactive Genetic Algorithm [11]. They proposed a model to create innovative dresses from a catalogue of components (predefined necklines, sleeves, skirts… See Figure 13). The main difference between our model and Kim's one is the space of possible solutions. As Kim worked with a set of predefined parts, which is a finite and discrete set, this space is limited, whereas our possible solution space if infinite and continuous.

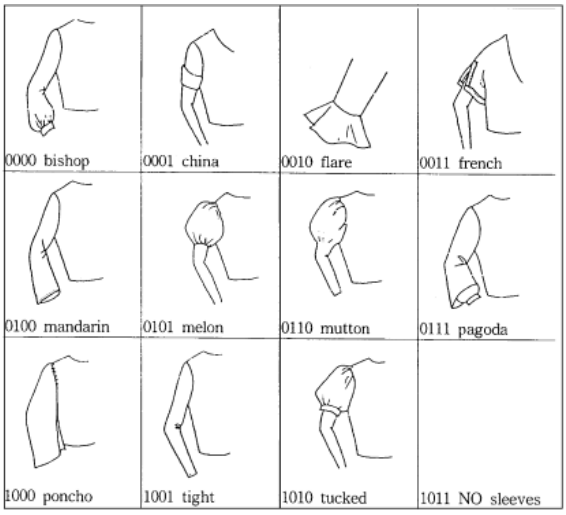


Figure 13. Examples of possible choices for arms and sleeves in Kim's fashion design model (taken from [11]).

To validate his model, Kim performed two different tests.

- The first test is called *convergence test*. It permits to ensure that the average fitness value according to each generation of the Genetic Algorithm increases with the generations. He made the test along 10 generations composed of 8 dresses. He used a panel of 10 users, working on two semantic attributes (*splendid* and *cool-looking*), that means that he asked the users to evaluate 10 generations of dresses according to the first attribute, and then to do the same process with the second attribute. The results showed a significant increase of the fitness.
- The second test is called *subjective test*. Its goal is to show that the results obtained with the model are better than without. Kim asked 3 users to find the 10 best dresses according to the two semantic attributes (*splendid* and *cool-looking*) among 500 individuals randomly created from the catalogue of permitted combinations (i.e. without his IGA). Then he asked the 10 previous users to find the best dress of the 10^{th} generation of the previous *convergence test*, according to the two attributes. The last stage was to compare for each attribute and for each user their own best dress obtained with the model with the 10 best dresses obtained without. The comparisons were made by pairwise comparisons, on a 7 degrees scale (from -3 to 3). Finally, the results showed that the individuals obtained with the algorithm reached on average a degree of about 2, compared to the individuals obtained without algorithm. So the model was really satisfactory.

But two main criticisms can be formulated on these two tests:

- Are the evaluations hedonistic or not? A hedonistic evaluation includes the preference

of the user (for example *"Please rate these individuals according to your preferences in terms of sportiveness"*; the user maybe likes sportive individuals, but not too much), whereas a non-hedonistic evaluation does not (for example *"Please rate these individuals according to their apparent degree of sportiveness"*).

- What is the meaning given by Kim on the pairwise comparisons? He asked the users to compare their own best dress (from the algorithm) with the 10 best dresses found by 3 other people. So it seems logical that the first one (found by the user himself) is better than the others (found by other people).

We built our test according to these criticisms.

4.1.2. Workshop realization

We have chosen to work for the two tests with a non-hedonistic evaluation (*"Please rate these individuals according to their apparent degree of sportiveness"*). We have also chosen two semantic attributes: *friendly* and *sportive* (half of the users work with the first one, the others with the second one).

The first test that we have defined is the same that the *convergence test* defined by Kim. We called it test of "*designer satisfaction*". We have worked with 10 generations composed of 20 individuals, and with a mutation probability of 0.05 and a selection rate of 0.7.

We have defined the second test (that we called test of "*satisfaction superiority of the IGA model*") with regard to the problem of the pairwise comparisons (expressed in Section 4.1). The process is the following:

- 400 car silhouettes are created randomly from an initial population of 20 real individuals. To clarify the notations in the next parts of the paper, these 400 individuals are called *paper individuals*. Each user has to find the 3 bests individuals according to his/her semantic attribute. Practically, these individuals are printed on paper and numbered, and the users record their evaluation in an Excel according to the following process:
- Each user evaluates the 400 paper individuals on the same scale as in the IGA (from 0 to 6). The best car silhouettes in his/her mind have to receive a 6.
- Only the individuals which receive a 6 are selected for the next evaluation. An Excel macro sorts out these individuals and a new evaluation table is presented to each user.
- The two first stages are repeated until each user finds the 3 best individuals. That means that the users must be more and more selective (to always have scores between 0 and 6 in order to progress).
- Then each user has to find the 3 best individuals of the 10th generation of the test of "*designer satisfaction*".
- Then a pairwise comparison [18] matrix permits to each user to compare pair to pair his/her own 3 best individuals of the algorithm with his/her own 3 best individuals obtained without the algorithm. The scale used has 7 degrees (from -3 to 3). This stage can be compared to the process followed by Kelly to validate his IGA [7].

		Best paper individuals		
Best individuals from the IGA		>>	>>>	>
		>	>>>	=
		>	>>>	=

Figure 14. Example of pairwise comparison matrix used for the tests.

Pairwise comparisons are explained in more details in [18]. They permit to evaluate simply a set of individuals without any absolute scale. An example of such a matrix is given on Figure 14. The comparisons have a direction to follow: here the individuals in lines are compared to the individuals in columns.

The evaluation scale for these comparisons is given in Table 4. The users work with the mathematic symbols (>, >>, =…), which are then replaced by numbers to analyze the data (see section 4.1.3.2).

-3	<<<	highly inferior
-2	<<	inferior
-1	<	slightly inferior
0	=	equal
1	>	slightly superior
2	>>	superior
3	>>>	highly superior

Table 4. Pairwise comparison scale.

The tests were realized during a short workshop. The panel of users was composed of 7 students (6 men and a woman) and a professor. 4 users received the semantic attribute *friendly* and the 4 others the attribute *sportive*.

About 2 hours were necessary to complete the three stages (time for presentation of the tests included):

- Evaluation of the 400 random printed individuals: 1 hour.
- Use of the Interactive Genetic Algorithm: 30 minutes.
- Pairwise comparisons: 15 minutes.

4.1.3. Results

4.1.3.1. Test of "designer satisfaction"

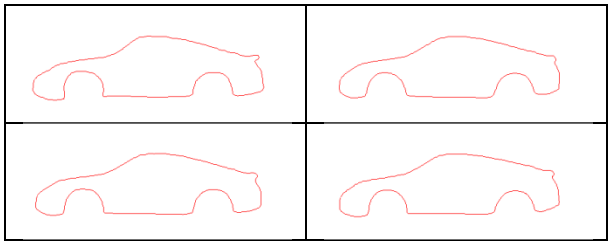


Figure 15. Examples of best "sportive" car silhouettes (one from each user).

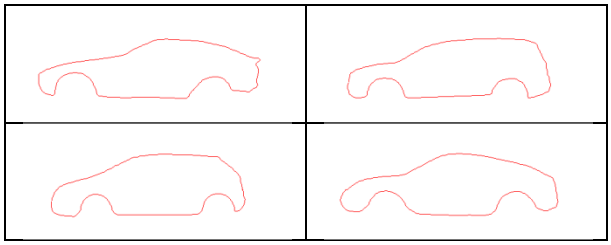


Figure 16. Examples of best "friendly" car silhouettes (one from each user).

The analysis of the car silhouettes obtained by the users during the first test shows interesting results.

For the two semantic attributes, the users reach in their own last generation the same type of car silhouettes. Moreover, those profiles are nearly the same for the semantic attribute

sportive, whereas they are really different from a user to another with the attribute *friendly*. Figure 15 and Figure 16 illustrate this phenomenon. The displayed silhouettes are chosen from the 3 best ones of each user. It is also interesting to notice that the car silhouettes from the 6th generation are almost the same for each user, which means that the diversity of the populations is weak from the 6th generation. In other words, the IGA seems to converge very quickly.

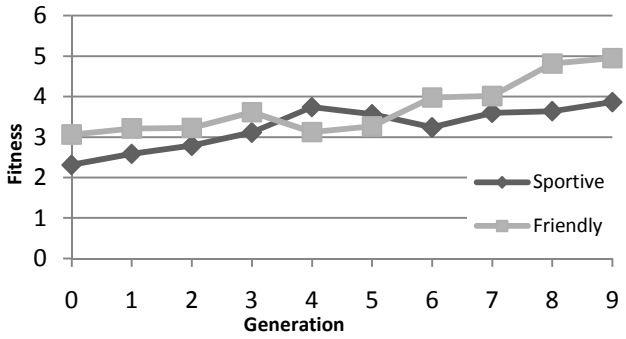


Figure 17. Fitness evolution for the convergence test.

It could be interesting to know if this quick convergence is wanted or not. This observation is not tackled is this paper. However, a way to change this speed is the mutation operator. For the tests, a really low mutation rate has been chosen. With a high mutation rate, the convergence speed would probably be lower, because a lot of novelty would appear at each generation.

Finally, the curves presented on Figure 17 can be drawn. In spite of local minima, the average fitness increases for the two semantic attributes, that shows a good global behavior of the model: the user satisfaction increases on average from 3.0 to 5.0 for the attribute *friendly*, and from 2.3 to 3.9 for the attribute *sportive*. So it is an increase of about 70%.

4.1.3.2. Test of "satisfaction superiority of the IGA model"

Table 5 shows the results of the test for each user and for each semantic attribute.

Sportive	User 1	+1,78
	User 2	+1,67
	User 3	+1,56
	User 4	+1,00
	Average	+1,50
Friendly	User 5	-0,33
	User 6	+0,67
	User 7	+0,78
	User 8	+1,22
	Average	+0,58
Total Average		**+1,04**

Table 5. Average values of the results of the subjectivity test (for the evaluation scale, see Table 4).

Some details have to be explained to understand the meaning of Table 5. The pairwise comparisons are made according to the following scheme: the best individuals from the IGA are compared to the best paper individuals. User 4 has for example an average evaluation of +1.00. According to Table 4, it means that user 4 found the best individuals from the IGA slightly superior to the best paper individuals. Thus positive numbers prove that the results obtained with our model are better than without.

The results show a good behavior for the semantic attribute *sportive*: the average score is

+1.50, so the best IGA individuals are between "slightly superior" and "superior" to the best paper individuals.

For the attribute *friendly*, the results show a good behavior too, even if the difference is slightly less perceptible. All the users prefer the individuals from the IGA except user 5, who prefers the individuals without the model. Globally, the results for this attribute are not really homogenous, which can be explained by a more subjective comprehension (and so characteristic to each user) of the word *friendly* than the word *sportive* (this observation will be highlighted in other tests presented after).

4.1.4. Synthesis

Those first two tests show a satisfactory behavior of our model. The user satisfaction increases with the generations, and the results are globally better with the IGA than without.
However, some major differences can be noticed between the two semantic attributes. Indeed the user perception is really different from a user to another: *sportive* seems to be perceived in the same way by everyone, whereas each user has a different perception of *friendly*. So to confirm these observations we have proceeded to more sophisticated post-processing analysis, using the similarity index described in section 3. We propose here five user tests that use the similarity index to process the same data (except the first one) as those used for the previous tests.

4.2. *Tests based on the similarity index*

4.2.1. Test of "novelty emergence"

The goal of this test is to show that it is possible to reach a defined individual which is not part of the initial population.

To answer this question we can execute a simple test. A subject draws on a sheet of paper a car body silhouette which comes spontaneously to his mind and which is not part of the initial population. This car body silhouette is taken as "reference individual". By working with our system he should try to obtain in the end the silhouette he had drawn before on the paper. To cope with this, he is supposed to evaluate the car solutions which look close to the reference individual with higher grades and those who look different with lower grades. By counting the number of generations he needs to reach the reference individual, we can estimate the quality of our design system.

Alternatively the target car silhouette may be an individual of the initial population that is removed from this initial population.

We have preferred to make abstraction of the designer subjectivity in automating the ability of the system to converge towards an ideal car silhouette, so as to measure the sole quality of the method. The role of the designer is played by an algorithm, which automatically evaluates the individuals of a generation in terms of their similarity to the target individual, thanks to the similarity index previously defined.

For the test we used the car in figure Figure 18.a as reference individual. The parameters for the genetic algorithm were the following: population of 100 individuals, turnover rate of 0.7 and mutation probability of 0.3. The mutation could change a gene in a range of ±(50%-200%). After 10 generations our system reached the car body silhouette in Figure 18.b which has a similarity index of 92%, which can be considered as a much satisfactory result.

The average fitness of the population converges over the generations to a high value (see Figure 19), whereas the value of the best similarity index in the population (the fitness of the fittest individual) raises rapidly from relative low 44% to 92%.

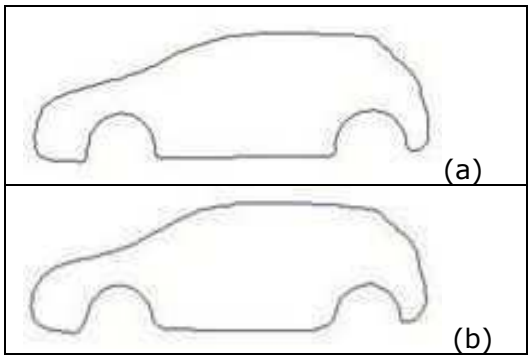


Figure 18. Comparison between the reference silhouette (a) and the final resulting silhouette (b)

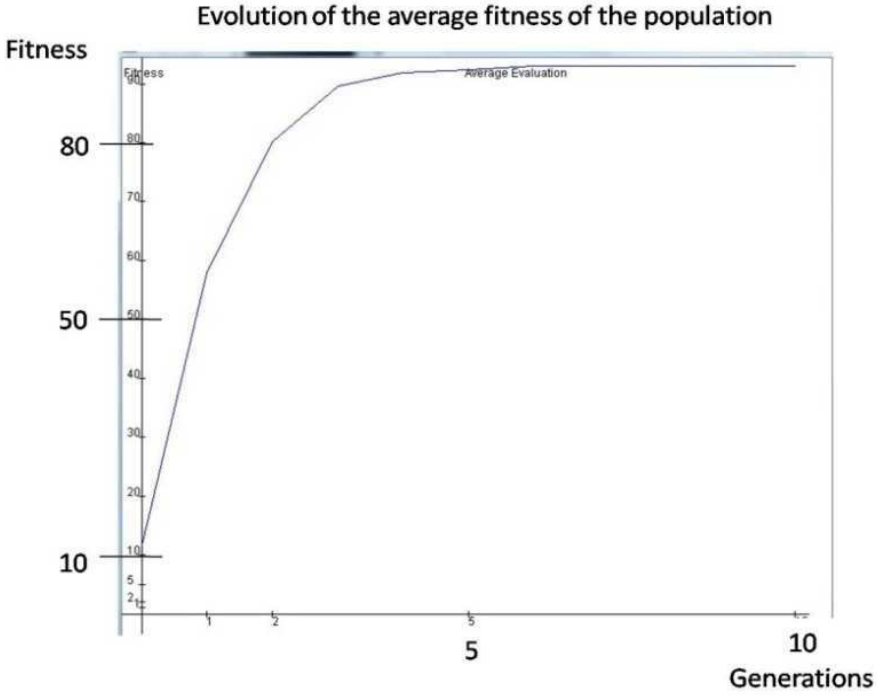


Figure 19. The average fitness of all individuals in the population over the generations

4.2.2. Test of "diversity lowering"

The aim of this test, called test of "*diversity lowering*" is to study:

- The evolution of the similarity index for each user along the ten generations.
- The similarities that could exist between the individuals of the last generation of each designer.

The idea consists simply to calculate the similarity index between all the individuals of all the users and for all the generations of the previous workshop, and to group together the data to observe potential correlations.

A high value of similarity for the last generation would mean that the IGA converges towards a single individuals family, and so that there is a diversity lowering.

These results concern the evolution of the similarity index of each user. The curves are shown on Figure 20. The initial population has an average of 6.5% of similarity. In all the cases and for the two semantic attributes, the values increase quickly. At the 6th generation (Generation #5 on the graph), the averages of all the users are above 40% similarity, and at about 75% for the last generation. The averages of each user (in italic in tables 4 and 5) are all included between 60 and 90% of similarity, which correspond to a level of very

strong similarities.

It shows that for each user, the model converges towards a single family of car silhouettes, which are very close in terms of visual aspect. But it means too that there is a real diversity lowering.

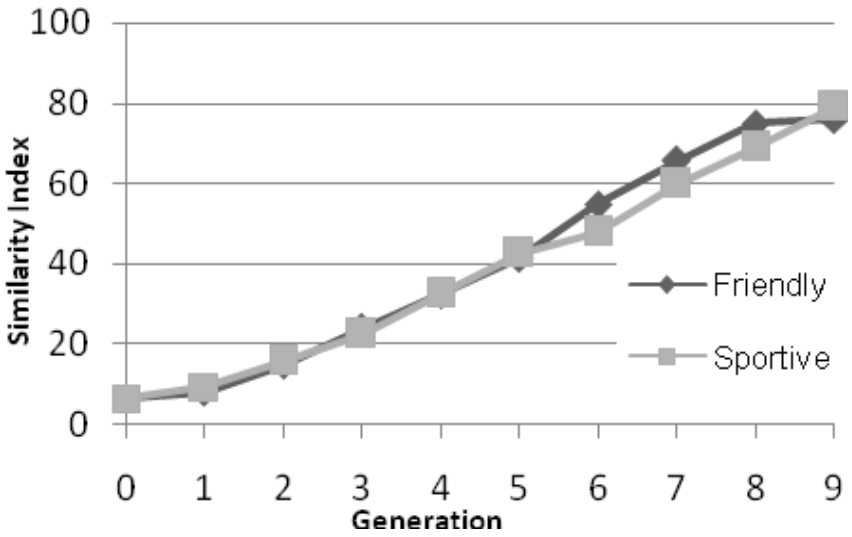


Figure 20. Evolution of the similarity index along the generations for the two semantic attributes

4.2.3. Test of "inter-designer convergence"

The goal of this test, called test of "*inter-designer convergence*" is to study the similarities that could exist between the last generations of all the users.

As in the previous test, the idea consists simply to calculate the similarity index between all the individuals of all the users and for all the generations of the previous workshop, and to group together the data to observe potential correlations.

The results are displayed on Table 6 and Table 7. These matrices represent the averages of the similarity index for the two semantic attributes:

- Between all the users (in bold).
- For each user (in italic). These values are treated in the previous test.

The values are really consistent with the visual aspect of the individuals. For the semantic attribute *friendly*, the values inter-designer (in bold) are not high, and the total average is only 9.21% similarity, whereas this total average is 50.83% for the attribute *sportive*. It means that the similarity between the last population of the users with the attribute *sportive* is very strong, and that there is no connection between those with the attribute *friendly*

	User 5	User 6	User 7	User 8
User 5	*79,07*	**6,91**	**4,02**	**2,89**
User 6		*89,65*	**20,65**	**8,81**
User 7			*61,02*	**11,97**
User 8				*74,16*
Total average			**9,21**	

Table 6. Similarity values between users for the semantic attribute friendly (in %)

These results confirm rigorously that the perception of the word *sportive* is the same for all users (they all came to the same type of profiles), whereas each user has its own perception of the word *friendly* (they all have a different kind of car silhouettes at the end of the

workshop).

	User 1	User 2	User 3	User 4
User 1	*79,95*	**47,87**	**63,09**	**36,20**
User 2		*72,34*	**74,14**	**42,62**
User 3			*89,61*	**41,04**
User 4				*74,74*
Total average			**50,83**	

Table 7. Similarity values between users for the semantic attribute sportive (in %).

4.2.4. Test of "superiority of the IGA model"

This test is called test of "*superiority of the IGA model".* Its goal is to show that our model obtains better results than without, in particular in terms of time and number of individuals. It consists in comparing for each user the 3 best individuals of the IGA with the 3 best individuals obtained without the IGA (printed on paper) with the similarity index.

The maxima of similarity are shown in Table 8. For 5 users out of 8, the maximum is above 70% of similarity. For 2 users, the maximum is included between 20 and 30%. For the last one, it is only 2.67% of similarity.

These values mean that it is possible to obtain the same results with and without our model. 5 users have found with the IGA *at least* one individual that is common or very close to one of the paper individuals.

400 individuals were proposed on paper. It took about 45 minutes for the user to evaluate them. Less than 200 individuals were used with the IGA (10 generations of 20 individuals, minus those who survive from one generation to the next one), during about 20 minutes. So we can say that our system seems to be able to bring the same results as without the model, more quickly and with fewer individuals.

Sportive	User 1	84,49
	User 2	21,26
	User 3	80,92
	User 4	97,93
	Average	71,15
Friendly	User 5	2,67
	User 6	98,09
	User 7	71,32
	User 8	26,94
	Average	49,75
Total Average		**60,45**

Table 8. Maxima of similarity values between best individuals of the IGA and paper individuals (in %).

4.2.5. Test of 'attraction in the surroundings of initial individuals"

The last test proposed in this paper is called test of "*attraction in the surroundings of initial individuals*". It tries to answer the following question: is the user really able to design his/her own car silhouette, or are the final individuals influenced by the initial individuals and close to their genotypes?

Our solution to this problem is to compare the best individuals of the IGA with the 20 car silhouettes of the initial population.

		User 2		
		Best Individual 1	**Best Individual 2**	**Best Individual 3**
Initial Population	**Ind 0**	2,04	1,93	1,84
	Ind 1	2,95	2,77	2,65
	Ind 2	10,02	9,02	8,80
	Ind 3	0,91	0,88	0,90
	Ind 4	3,45	3,24	3,35
	Ind 5	2,04	1,94	1,91
	Ind 6	1,42	1,36	1,36
	Ind 7	1,78	1,74	1,52
	Ind 8	13,90	13,73	10,75
	Ind 9	6,62	6,17	6,53
	Ind 10	2,76	2,60	2,47
	Ind 11	3,28	3,41	2,91
	Ind 12	2,08	1,97	1,97
	Ind 13	69,65	69,89	53,21
	Ind 14	3,50	3,30	3,46
	Ind 15	7,15	7,23	10,83
	Ind 16	2,07	1,97	1,93
	Ind 17	3,05	2,88	2,87
	Ind 18	1,65	1,58	1,47
	Ind 19	1,99	1,93	1,69

Table 9. Example of similarity values for the comparison between the best individuals of the IGA and the initial population (in %).

The results of this test are very interesting, but too sizeable to be displayed here. That is why only the case of user 2 is presented on Table 9.

This table shows that the best individuals of user 2 are very close to Individual #13 of the initial population (more than 50% of similarity). Individual #13 represents the silhouette of a Porsche 911. Two other minor influences can be noticed (with Individuals #2 and #8), but the values are below 20%, which is not significant. User 2 worked with the semantic attribute *sportive*, and it is important to notice that the same results are obtained with the 3 other users who worked with this attribute.

That means that they all came to the same type of profile which is very close to a Porsche 911. In other words, all the users perceived a *sportive* car as a Porsche 911.

For the semantic attribute *friendly*, the results are totally different. No car that influences all the users can be identified. In 2 cases out of 4, no significant influence of an initial individual is noticed. In the two other cases, such a car can be identified, with the scores of 40% of similarity, but this car is different for the two users (Porsche 911 for the one, Chevrolet Corvette C4 for the other). It shows again that this attribute is differently perceived.

4.2.6. Synthesis

Those five tests based on the similarity index permit to prove rigorously some results. Indeed in all cases the system turns out to converge towards a uniformized population. For the semantic attribute *sportive*, this profile is the same for any of the four users and is close to a Porsche 911. For the semantic attribute *friendly*, no similar influence from the initial population is identified, and the results really depend on the user. The perception of the

attributes is really different from a user to another.

Moreover, our model converges quickly and with less individuals than without the model, which is really satisfactory.

However, some remarks can be formulated:

- The model always converges towards one profile family. Is it not possible to converge towards at least two different families?
- What happens if the Porsche 911 or the other influencing cars are not in the initial population? Do the users even find them?
- Why are there so many differences in the perception of the semantic attributes? Some concepts of Emotional Design could be introduced to develop a more powerful model.

4.3. *Additional tests*

In this work, we have investigated two sides of the interactive genetic algorithm application, related to user preference identification: convergence and bias of the procedure. Can the user always reach his or her preferred shape through successive generations, i.e. can we ensure convergence? Can different users with unique personalities reach different result sets when looking for the same attribute, i.e. are the end results, in this case, biased by each designers' preferences? Since the algorithm takes an input set of shapes for the first generation, for a given designer, are his choices and selections influenced by this initial, starting set of shapes? We will especially look at the case where there is a hypothetical implicit attraction of the user to one of the elements in the initial set.

We have compared the responses of the users to two different initial sets of the algorithms. The first one being the 'reference' test set (see test in section 4.2.5), and sporting the digitized shapes of real-life cars; the second set being a copy of the first one, *except* for the thirteenth element, which is in the first set a "Porsche 911" and in the second set a copy of the first element (Audi 44). These two initial sets, or "generations zero" are composed of twenty-one elements each. At each generation of the IGA, the set of possible user choices changes and is composed of either P=21 individuals (there are as many choices as elements in the initial set) or P=6 individuals: in the latter case, there are less possible choices, but they can all be displayed on the same screen and avoid user fatigue while browsing through them. Also, a lower value of P lets the user access a higher number of assessed generations N; N is typically in the range $[\![10;30]\!]$. It may however reduce the field of possibilities, but it is shown later that the convergence of the results make this reduction irrelevant after just a few generations.

We say that the algorithm has "converged" if the individuals composing the set at the final generation are similar (in the sense given by the similarity index, see below) between themselves. A similarity matrix is constructed through pairwise comparisons with the best individuals a user got for his/her final generation compared to the twenty-one individuals composing the initial set. Another variable, the "subjectivity" of the assessment is visible with an inter-user comparison of the results. A similarity matrix is shown in order to compare the best individuals of each user between the other best individuals of another user. Finally, the "similarity index" is a simple transformation of the mathematical distance on the "genome" space, which is defined by the family of seventy gene-vectors. With this method, the index is expressed as a percentage value, thus being greater when individuals are "closer" i.e. more similar to each other according to the user. A significant value for this similarity index is 20%: this is the upper limit, below which no user could spot differences between cars. 100% means exact mathematical equality, and is accordingly never reached in practice.

For reference, in the tests, the mutation parameter of the genetic algorithm was always chosen equal to 30%, and the population selection rate set to 70%.

Having made test runs for different users and analyzed the results, we draw three main

conclusions with respect to user behavior within the context of the IGA.

4.3.1. Cross-over of two "mutants"

The cross-over of two mutants can lead to a sportive and acceptable individual.

For this, we have taken two "mutants" from one of the tests, i.e. two individuals which do not look like cars anymore, but nonetheless generated from an initial set of cars. These shapes are usually excluded right away by the user (marked 'one' or 'zero'). Their combination or cross-over can still generate an individual that looks like a proper car.

Our recommendation would then be that mutants are not to be excluded from the set of possible choices, since they can lead to innovative cars with the right cross-over (in other words, keep the algorithm like it is now and do not try to remove mutant cars). However, there are a few drawbacks: first and foremost, a user would typically exclude a mutant car as soon as it appears, since it does not look like a real car. When this happens, the global convergence curve shows "bumps", since the average mark given to the generation containing mutant cars drops for that very reason. The second limitation is that such acceptable cross-over of mutants may not often happen, since the cross-over which is used here is a based on a mere linear combination of "genes" (Fourier coefficients), which means the cross-over has an average effect on the curves. Another method of cross-over would then perhaps be more appropriate in order to make the most of "mutant" shapes.

4.3.2. Convergence of individual users' choices

The tests have shown that **convergence always happens for each user**. Even with an initial set without Porsche 911, the users who are mostly influenced by the Porsche 911 were able to get an individual similar to the one they got with the complete initial set.

The Porsche 911 notably influenced two of the four users, for whom the results are comparable. The results for the other users are really different. Without the Porsche 911, some individuals were more influenced by the Lamborghini element. However, there is no real attractor, other than the Porsche 911: no similarity index is higher than 20%.The inter-user comparison shows that individuals are not similar, which means each user was able to express his/her own preference. Also, the results are more stable: tests can be conducted again and give similar results for each user.

4.3.3. Capacity of a computer algorithm to generate surprise

We have witnessed that the genetic algorithms can indeed **create surprising shapes**, and more specifically, that **mutation is a key to creativity**.

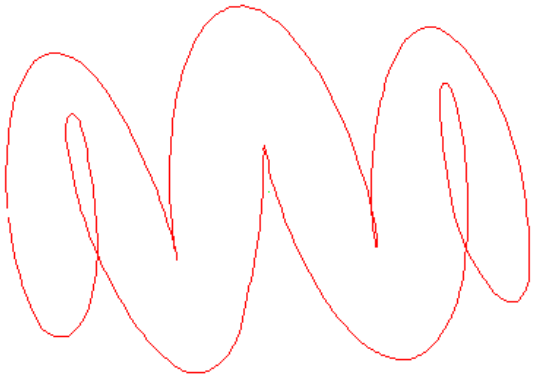


Figure 21. The "octopus" individual

A specific test was conducted with the semantic attribute "mutant" instead of "sportivity", in order to test the ability of IGA to create elements as far as possible from the initial set of car-like elements. The mutation parameter is fixed to 1. The number of generations is 100.

We were able to get an "octopus" element, which shows less than a 1% similarity index to any of the cars of the initial set which was used to create this "mutant" element (documented in the attached technical report) closely resembling an octopus. However, the gene which was the most mutated was the gene n°6, which was already known as the gene whose variations affected the appearance of generated silhouettes the most.

But beyond the perceived distance of the shape from the initial set (of cars!), it actually shows that the use of mutation in the genetic algorithms empowers the designer with a real potential for creativity. Hence, for all its aforementioned disadvantages, it can still be useful to one of the main goals of the IGA: to create surprise.

5. Mutation operator

5.1. *Some reflections*

The mutation is the phenomenon that changes the genome of an individual randomly, without taking into account the genome of the parents. In this respect, it adds innovation to the evolution process. And like Kelly, we see the mutation as "a way of introducing new variable values into the population, as well as exposing potential exciting design spaces" [5].

But it is crucial, however, that the mutation is carefully tuned, or "managed". In order to attain all the possible children from the modification of a genome, and hence show maximum creativity and surprise to the designer the mutation, should ideally be free, and unconstrained; in that sense, it is able to achieve all the opportunities offered by a genome representation.

But this freedom has to be counterbalanced by the need for "reliability": we in fact try to avoid "degenerate" individuals arising from mutation that do not resemble the original class of objects. As much as we would like to introduce variety and originality, it still has to be constrained within some acceptable bounds, so that the generated shape will still attract the eye of the designer, and not clutter him/her with shapes that could be automatically rejected.

Genetic algorithms commonly implement mutation relatively freely, because they often dispose of significant computing resources and run on a large number of generations; it is not impossible with today's computing powers for an algorithm running on a typical home computer to evaluate thousands of individuals per generation. However, our work is focused on an *interactive* genetic algorithm. The fact that the algorithm involves user interaction, since a human designer evaluates the individuals generated, necessarily limits the number of individuals that can be proposed for evaluation at each iteration, and also the number of generations that can be processed in a reasonable time. The designer being at the heart of the process, we must then limit the number of unacceptable shapes that will always be rejected, at the expense of freedom.

The mutation, as it is currently implemented, is based on the fact that "key" genes numbered 2 to 10 are the most influential figure of the vehicle. The reason stems from the Fourier series decomposition coefficients, which are used to encode the genome of the shape. Thus, the degree of mutation change, which is adjustable by using the user interface, is actually the probability that each individual mutates. In such a case, then a gene will be drawn randomly from the 9 key genes (with equal probability), then its value will be multiplied by an element of the set [-2, -1.5] u [1.5, 2] (with a uniform probability distribution).

The current Fourier method for encoding the genome of a shape is such that changing one and only one gene when an individual mutates causes an often not very pleasant geometric transformation. For example, as a result, ripples appear in the mutated shape that transform a "healthy" original individual in a mutated shape no longer has the minimum visual clues that the user would use to identify a car silhouette: the car floor may not be flat anymore, or the wheels will lose their circular shape, resulting in a car that may not be functional anymore. Whereas such alterations may be acceptable in the case of an *automated* genetic algorithm, this phenomenon makes the implementation of mutation in an *interactive* genetic algorithm prohibitive, because the designer will not accept any deformity, and eliminate the mutant individual from the current population, even though it exhibit other interesting features and innovation, or could transmitting its special features to other generations, which may have been wanted by the designer.

5.2. *Conclusions and ideas for future work*

The current method implemented needs to be revised and corrected. A proposed solution

that stems directly from the aforementioned remarks could be to identify the main features that the user usually recognizes in a class of shapes (e.g. for a car, the location of the wheels, or the flat shape of the floor), and then reconstruct (post-mutation) these features automatically.

However, such a method may prove complex to implement in practice, given the choice of the genome representation. From our work (presented in the technical reports), other methods of mutation are possible and seem more interesting:

- A first method could be a combination of genes; since the mutation of a single gene causes most of the time a very noticeable geometric distortion, a combination of several genes in a single mutation step could reduce the number of unacceptable mutations.
- A second method could be to gather dynamic statistics about the designer's choices: the mutation algorithm could collect the history of the changes it has carried out in all the generations before the current one, and the evaluation of these changes given by the user. Hence, we would statistically favor mutations that please the user, to the detriment of mutations that would never have been accepted in the first place. Although it would seem attractive in the first place, this method would have two main drawbacks. Firstly, it would discard any chance of "invisible" mutations happening, i.e. mutant individuals included in a generation's population but not shown to the user, even though these individuals could be subsequently combined in perfectly viable shapes; and secondly, it requires a large number of samples (records of changes and corresponding evaluation) to be able to provide relevant statistics. However, an interactive genetic algorithm is commonly evaluated on only a few generations, while the user is making his or her choices. A solution could be that statistics be built on a common database, enriched by different users. But besides the technical difficulty of such an implementation, we may "dilute" the specific preference of each user which may vary depending on what he/she is looking for at a particular moment in time.

6. Conclusions and perspectives

We have presented in this paper an innovative car silhouettes design model. Based on an encoding method of the genomes by Fourier decomposition, it offers to the car designers the possibility to create new car silhouettes from an initial population according to their preferences.

Several user tests have been carried out to study the convergence of the model in terms of user satisfaction, but also in terms of perceived distance, using a similarity index. The results show very satisfactory results. They permit to answer the four questions defined in the introduction:

- *Are the users satisfied by our model?* Yes, the satisfaction increases along the generations of the IGA, and the users prefer to work with our model rather than without.
- *Are the results better with our model than without?* Yes, in terms of user satisfaction, but also in terms of time and numbers of processed individuals.
- *Does our model really create novelty?* Potentially. It highly depends on the chosen semantic attribute. But individuals than are far from the initial population in terms of similarity can be created.
- *Are the results different from a user to another?* Potentially. It depends on the chosen semantic attribute too.

In consequence, our system should allow style designers to converge towards intuitive ideas and to make emerging surprise in exploring large spaces of potential silhouettes.

Further tests with a larger panel of user could be done to validate the statements described in this paper and dealing with Emotional Design. The large difference of results between the simple semantic attributes *sportive* and *friendly* open a very interesting field of study for the future with the objective of creating CAD tools that really meet the user perception.

The real role of the mutation operator has to be studied too. So the identification of the optimal parameters would permit to develop a more reliable model.
Finally these first results are very promising and numerous research perspectives appear now:

- Allowing a step of "direct modification by the designer" within an intermediary generation, i.e. modifying some details of a silhouette curve or even adding new individuals to the population. We must acknowledge here that all our 20 initial silhouettes are silhouettes of existing commercial cars. Then, the experiment in this paper has just consisted in morphing between known solutions. For really creating surprise and innovating, we must also test our system in a more creative way.
- Combining several series of closed curves to better define the important lines of a car (see [12]).
- Making more complex the interactive assessment of individuals by the style designer through multicriteria assessments under several perceptual attributes (see [19]).
- Exploring a new product field: the encoding method that we use is very flexible and applicable to many design objects. We think that excellent results could be obtained with simple shapes. The ideal form would be a simple 2D-closed-curve from a solid of revolution (which permits to describe the whole product with a single closed curve), like stemmed glasses or vases.
- Finally, a way to improve the model could be to develop a function for introducing car silhouettes drawn by the designer himself. For instance, a new form appears in his/her mind during the IGA process, so he/she can draw this form on the screen (or with some graphic tool) and introduce the new individual in the current population.

7. References

7.3. *Previous publications*

1. Yannou B., Dihlmann M., and Cluzel F., 2008, "Indirect encoding of the genes of a closed curve for interactively create innovative car silhouettes," Proceedings of International Design Conference, DESIGN 2008, Dubrovnik, Croatia.
2. Yannou B., Dihlmann M., and Awedikian R., 2008, "Evolutive design of car silhouettes," Proceedings of ASME 2008 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference, IDETC/CIE 2008, New York City, NY, USA.
3. Cluzel F., and Yannou B., 2008, "Evaluation subjective de profils évolutifs de voiture," CONFERE'08, Angers, France.
4. Cluzel F., and Yannou B., 2008, "User tests for the convergence of a car silhouettes design model using an Interactive Genetic Algorithm," Proceedings of IDMME - Virtual Concept 2008, Beijing, China.
5. Cluzel F., and Yannou B., 2009, "Efficiency Assessment of an evolutive design system of car contours," Proceedings of International Conference on Engineering Design, ICED'09, Stanford, CA, USA.
6. Cluzel F., Yannou B., and Dihlmann M., 2011, "Putting the designer in the loop of a genetic design platform for the selection of two-dimensional concept shapes. Does it work and is it efficient?," To be published in International Journal on Interactive Design and Manufacturing.

7.4. *References*

[1] Bentley P. J., and Corne D. W., 2002, "Introduction to creative evolutionary systems," Creative evolutionary systems, Morgan Kaufmann Publishers Inc., pp. 1-75.
[2] Renner G., and Ekart A., 2003, "Genetic algorithms in computer aided design," Computer-Aided Design, **35**(8), pp. 709-726.
[3] Poirson E., Dépincé P., and Petiot J., 2007, "User-centered design by genetic algorithms: Application to brass musical instrument optimization," Engineering Applications of Artificial Intelligence, **20**(4), pp. 511-518.
[4] Nicaise F., and Antonsson E., 2007, "Indirect encoding of structures for evolutionary design," Proc. International Conference on Engineering Design, ICED'07, Cité des Sciences et de l'Industrie, Paris, France.
[5] Kelly J., Papalambros P. Y., and Seifert C. M., 2008, "Interactive Genetic Algorithms for use as Creativity Enhancement Tools," AAAI 2008 Spring Symposia, Palo Alto, California, USA.
[6] Ang M., Chau H., McKay A., and De Pennington A., 2006, "Combining Evolutionary Algorithms and Shape Grammars to Generate Branded Product Design," Design Computing and Cognition '06, J.S. Gero, ed., Springer Netherlands, Dordrecht, pp. 521-539.
[7] Kelly J., Wakefield G., and Papalambros P., "Evidence for using interactive genetic algorithms in shape preference assessment," International Journal of Product Development.
[8] Yanagisawa H., and Fukuda S., 2003, "Interactive Reduct Evolutional Computation for Aesthetic Design," Proc. ASME 2003 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, IDETC/CIE2003, Chicago, Illinois, USA, pp. 1063-1072.
[9] Liu H., Tang M., and Frazer J. H., 2002, "Supporting evolution in a multi-agent cooperative design environment," Advances in Engineering Software, **33**(6), pp. 319-328.
[10] Osborn S., Cagan J., Pawlicki R., and Smith R. C., 2006, "Creating cross-over vehicles: Defining and combining vehicle classes using shape grammars," AIEDAM, **20**(03).
[11] Kim H., and Cho S., 2000, "Application of interactive genetic algorithm to fashion

design," Engineering Applications of Artificial Intelligence, **13**(6), pp. 635-644.
[12] Cheutet F., 2007, "2D semantic sketcher for a car aesthetic design," Proc. CPI'2007 : Conception et Production Intégrées, Rabat, Maroc.
[13] Petiot J., and Dagher A., 2010, "Preference-oriented form design: application to cars' headlights," Int J Interact Des Manuf.
[14] Mallat S., 1999, A Wavelet Tour of Signal Processing, Second Edition, Academic Press.
[15] Gruber T. R., 1995, "Toward principles for the design of ontologies used for knowledge sharing," International Journal of Human-Computer Studies, **43**(5-6), pp. 907-928.
[16] McGarva J., and Mullineux G., 1993, "Harmonic representation of closed curves," Applied Mathematical Modelling, **17**(4), pp. 213-218.
[17] Vasiliu A., and Yannou B., 2001, "Dimensional synthesis of planar mechanisms using neural networks: application to path generator linkages," Mechanism and Machine Theory, **36**(2), pp. 299-310.
[18] Yannou B., and Coatanéa E., 2007, "Easy and flexible specifications and product evaluations by expert and customer comparisons with existing products," Proc. International Conference on Engineering Design, ICED'07, Cité des Sciences et de l'Industrie, Paris, France.
[19] Dagher A., and Petiot J., 2007, "Study of the correlations between user preferences and design factors: application to cars front-end design," Proc. International Conference on Engineering Design, ICED'07, Cité des Sciences et de l'Industrie, Paris, France.

8. Appendixes

8.1. *Screenshots of the Java interfaces*

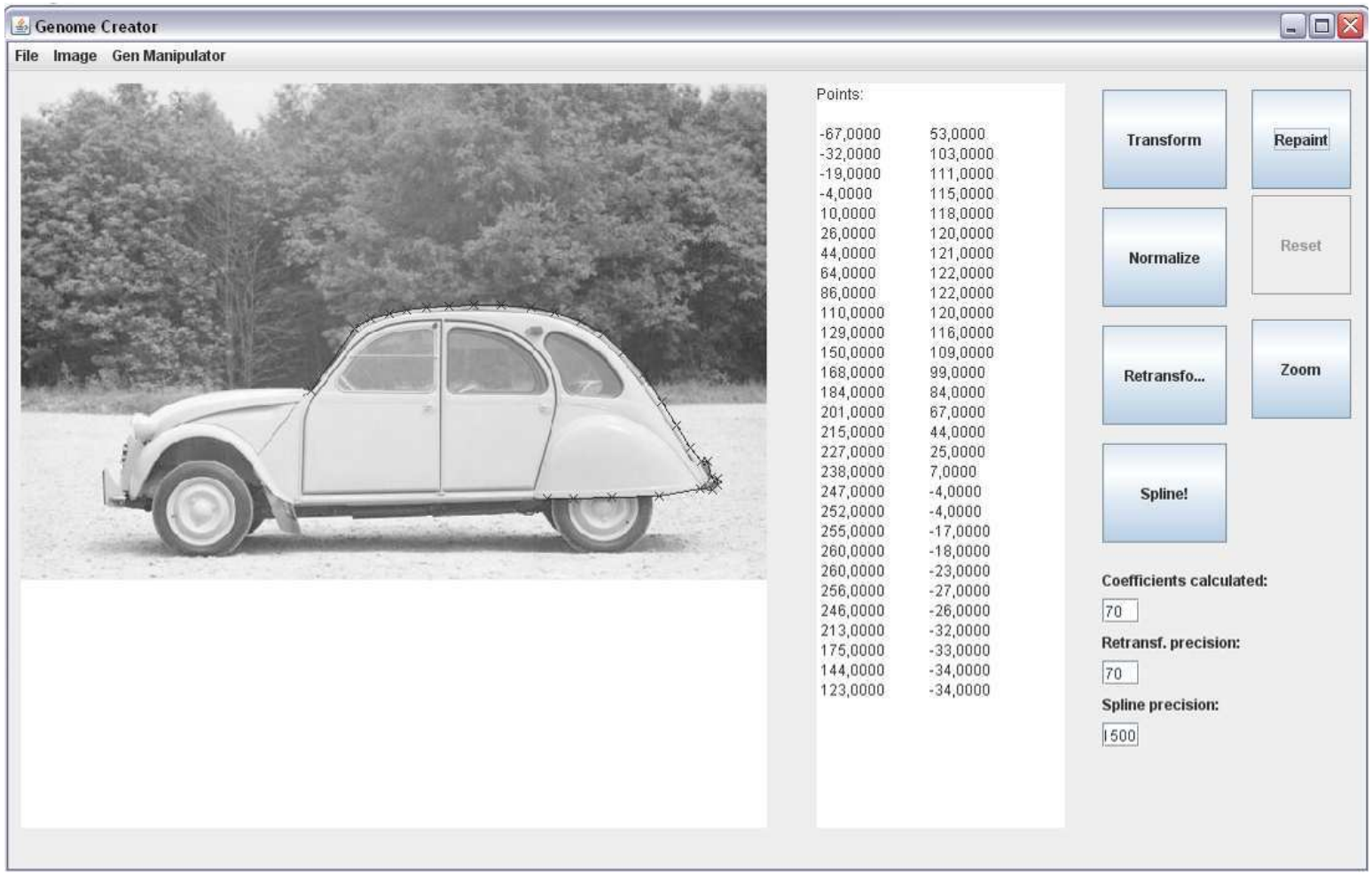


Figure 22. The genome creator

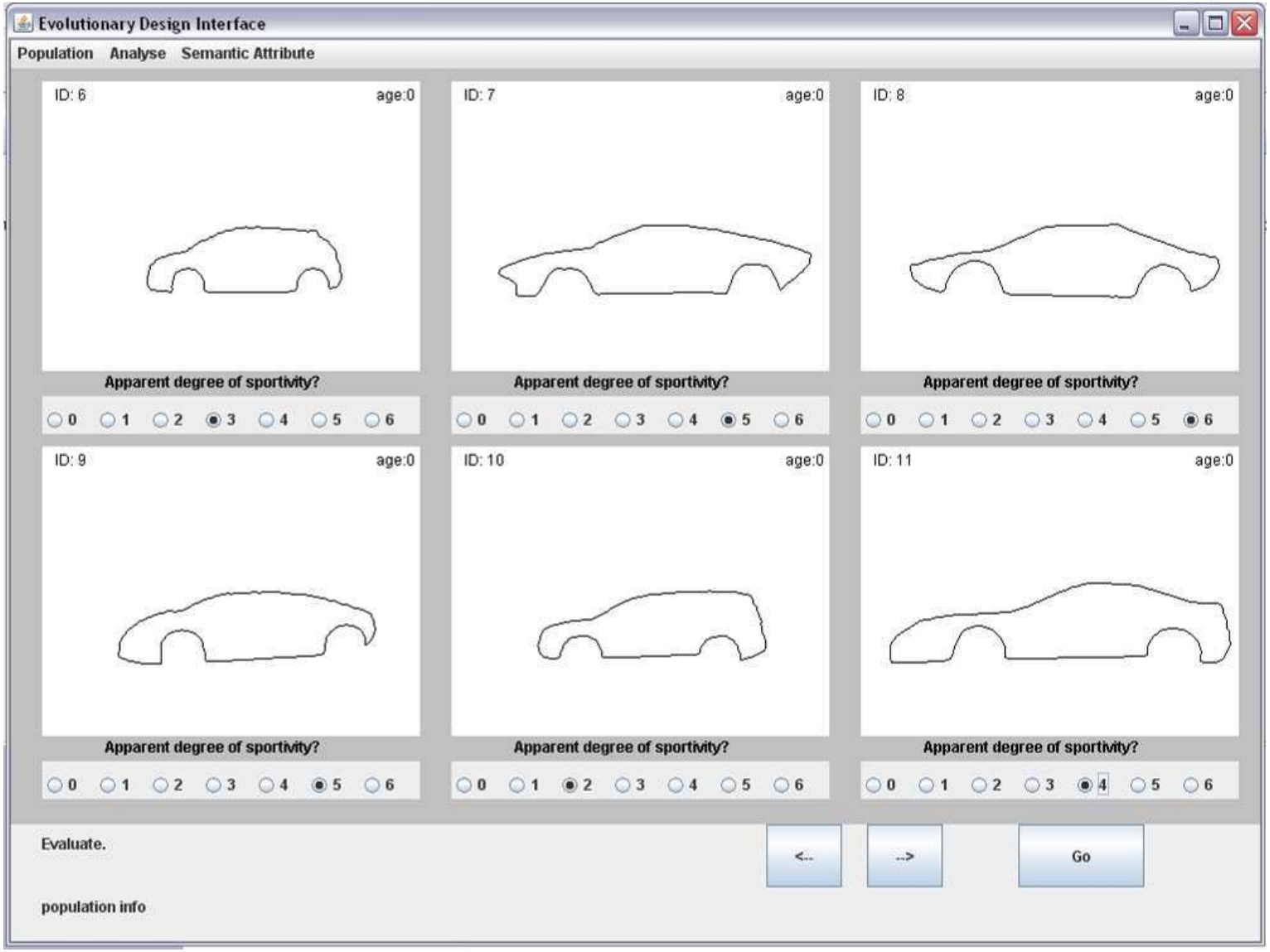


Figure 23. The user interface to run the IGA

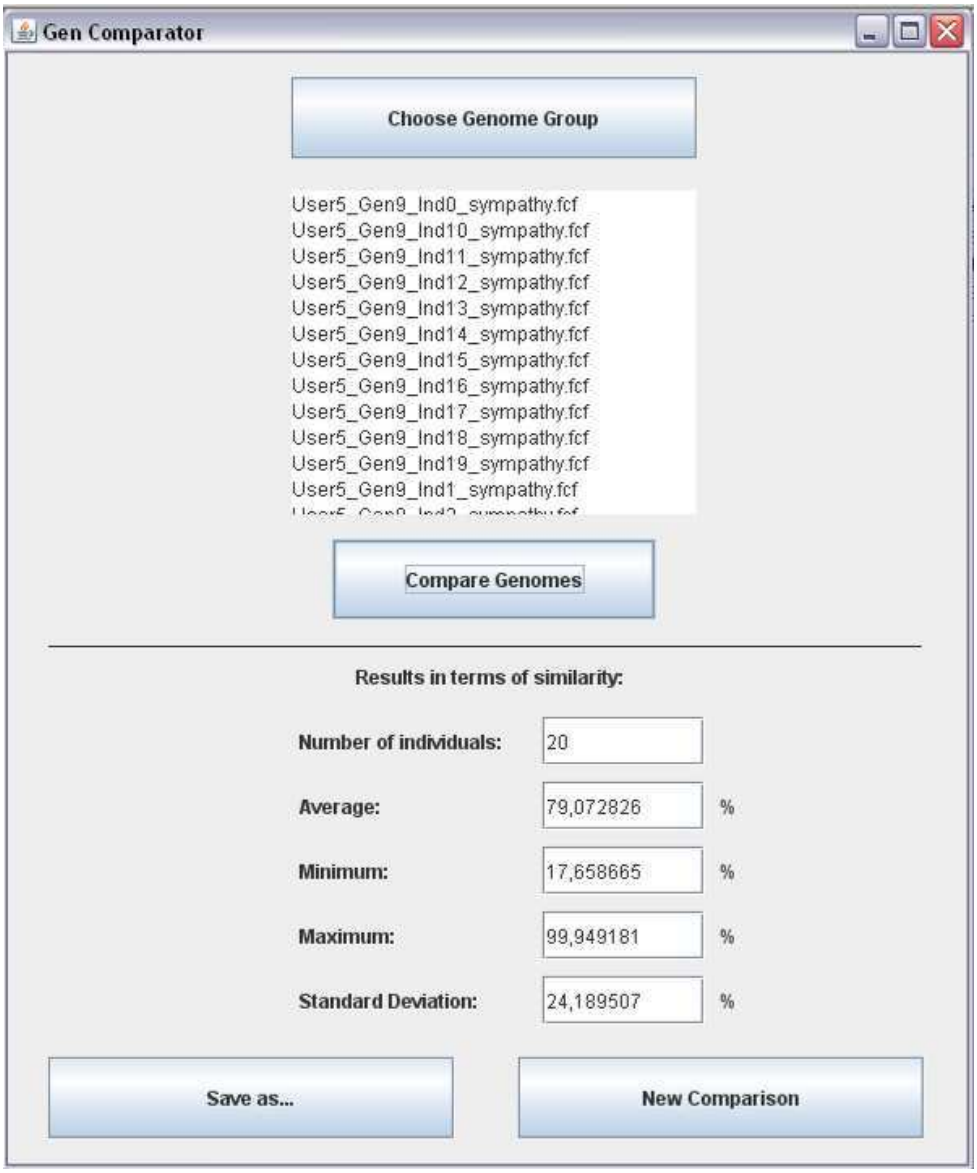


Figure 24. The genome comparator (using the similarity index)

8.2. *Best individuals chosen by the users*

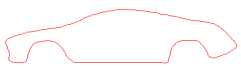
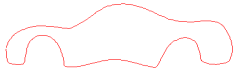
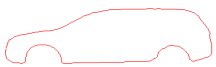
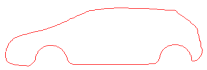
		3 best individuals of the 10th generation of the IGA			3 best individuals among the 400 individuals printed on paper		
Sportive	User 1						
	User 2						
	User 3						
	User 4						
Friendly	User 5						
	User 6						
	User 7						
	User 8						

Figure 25. Best individuals chosen by the users

8.3. *Example of individuals printed on paper*

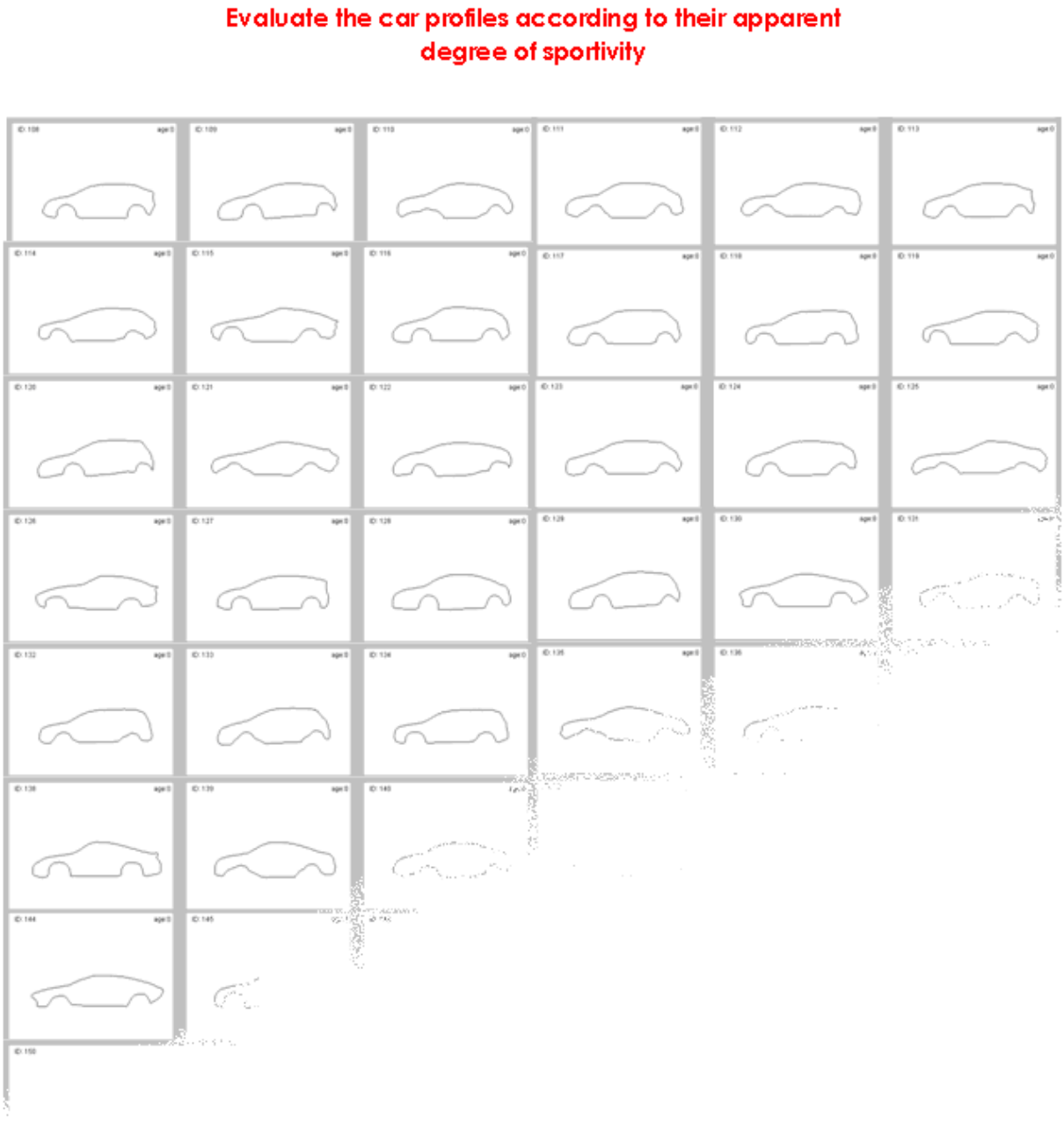


Figure 26. Extract of the 400 random "paper individuals" used for the user tests

8.4. *Process to choose the 3 best individuals printed on paper*

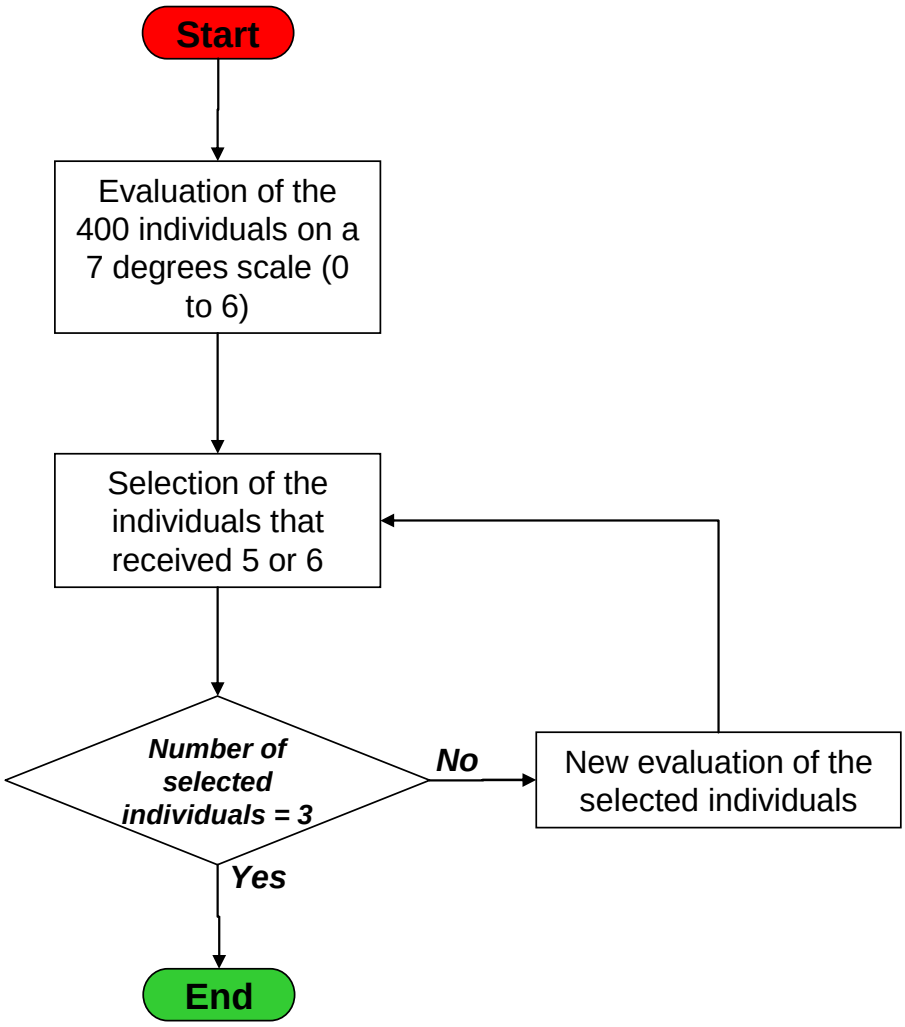


Figure 27. Process to choose the 3 best individuals printed on paper among the 400 proposed individuals

8.5. *A perspective : direct modification of a car silhouette by the user*

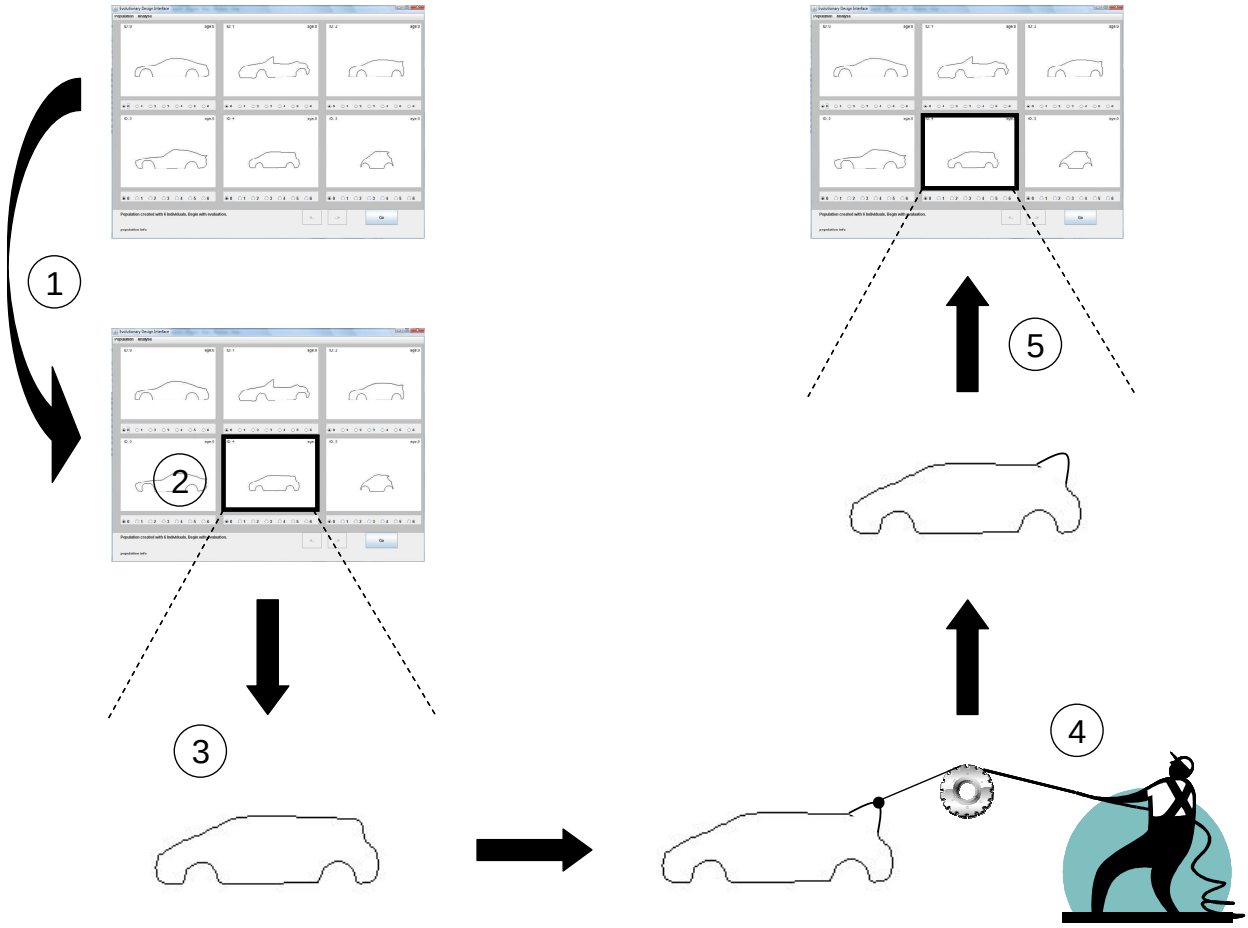


Figure 28. Direct modification of a car silhouette by the user

We can imagine the following scenario:

1. The user drives the IGA through several generations.
2. At the generation *n*, he remarks an interesting profile that but he would like to change a detail in particular.
3. He extracts the profile from the population towards a graphical editor.
4. He modifies the curves by pulling it in the graphical editor.
5. The silhouette is reinstated to the population (at the generation n) and the IGA process continues.