

## **A MULTIOBJECTIVE AND INTERACTIVE GENETIC ALGORITHM TO OPTIMIZE THE BUILDING FORM IN EARLY DESIGN STAGES**

Xavier Marsault

MAP-ARIA | UMR 3495, Ecole Nationale Supérieure d'Architecture de Lyon,

3 rue Maurice-Audin, BP 170, F – 69512 VAULX-EN-VELIN CEDEX

Tel : +(33) (0)4 78 79 50 88

### **ABSTRACT**

As simulation researchers in the field of performance-driven architecture, we mainly describe in this paper an interactive genetic algorithm (IGA) especially developed for eco-performance and real-time creative design simulations, associated with a simple and intuitive human machine interface. It has been originally created during the french ANR project EcCoGen dealing with creativity assistance, with the objective of "reducing the gap" between architectural design and current scientific knowledge needed to optimize the building form in early design stage, reduce its energy consumption and bring a real help to the architect's decisions.

### **INTRODUCTION**

The integration of building performance simulation in the early stages of the design process is a recent field of research, put in the context of sustainable development and energy savings. During the last decade, many publications on simulation for architecture and engineering design have been released, combining generative and optimization techniques, sometimes including form synthesis and multicriteria decision aid (Attia et al., 2012), (Baverel et al., 2012), (Hamdy et al., 2012), (Kicinger, 2006), (Mourshed et al. 2003), (Shi et al., 2013). This coupling, designed to drastically reduce the explorable solution space provides in reasonable time a number of optimized solutions low enough to be considered in practice.

It is nowadays a priority to assist the architects during the early design stages, by integrating most of the parameters and constraints of sustainable development through contemporary design approaches in order to guide the creation through innovative achievements. In this way, our IGA introduces creativity within the generative strategy, bringing optimization, exploration and inspiration for architects. It is an evolutionary strategy derived from Jaszkiwicz MOGLS (2002) and improved with recent advances on adaptive parameters tuning in order to preserve high performance while maintaining high diversity. The criteria chosen by the user can be aesthetic, energetic, functional or constructive (or all), as they can be based on an analysis of the potential solution's performances displayed by EcCoGen. Our IGA preserves certain characteristics of the selected objects, while it continuously optimizes and evaluates in real time their

overall performances (in the Rhinoceros software). In this way, it is very different from MIT Design Advisor, or BEopt, among other programs, because we have almost real time fitness evaluation while preserving continuous 3D interaction and performance outline visualization.

During the process, the architect interacts with populations of optimized solutions and still has the possibility to orient the evolution in various specific directions, according to performance or subjective choices. Our algorithm mixes genetic capabilities with a pheromone approach derived from Ant Colony Optimization (Angus et al., 2009), in order to enhance the user's creativity with persistence and evaporation features. The underlying concept is to achieve improved performance via evolution while keeping certain resemblance with the user selected choices. Thus, our IGA behaves as a user preference learner with a coherent stabilization effect.

An original Graphical User Interface (GUI) is conceived, embedding complex simulation model. However, the interaction between the user and the software is permanently preserved.

### **The morphogenetic model**

Within a built environment (figure 5), the architect draws a blank plot (green) cut by a 3D regular mesh whose ground frame is configurable but fixed, and sets the maximum floors number. This mesh leads to the "capable volume" (CV) which outlines and describes all possible solutions. A solution is a collection of dwelling units (variables = form genes), each one owning a programmatic function (office, housing,...) and properties (materiality, opacity, energy characteristics). The number of such units is the programmatic constraint, which can own a tolerance, or freely evolve in an exploratory approach.

### **The energetic model**

The current thermal model used in EcCoGen is restricted to heating-dominated buildings with high levels of insulation. Since our main objective is the preservation of interactivity, simplified polynomial regression models have been designed to make the evaluation time less than one second. The validation procedure of these have been published in an international journal (Mavromatidis et al. 2013).

### **The software framework**

EcCoGen framework is based on two software

environments (figure 1). The first one is *Processing*, an open source programming language based on Java. The Morphogenetic Engine, the Interactive Genetic Algorithm and the Graphical User Interface are implemented within *Processing*. The second software environment is Rhinoceros® with its Grasshopper plug-in. Rhinoceros is a 3D modeler and Grasshopper allows easy 3D programming for the evaluation engine. The communications between them are performed by UDP protocol.

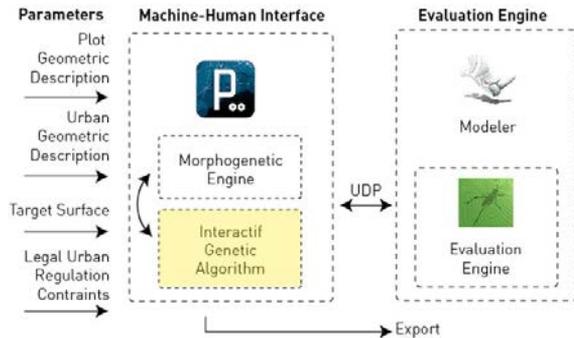


Figure 1 The IGA within EcCoGen framework

## PREVIOUS WORK

### Multiobjective IGA

IGA denotes a genetic algorithm (GA) where the user can interact with the process and the populations, especially in the selection and mutation steps. The main difficulty is to find a compromise between pure optimization and the persistence of interest areas and research directions in the landscape research.

To track the user's choices, a good IGA must ensure a certain persistence, which should not be too short (to give the GA the time to soak in) or too long (not to restrain exploration to other areas of potential interest). As in (Quiroz et al., 2008), one may accept to keep solutions "objectively worse" if they are amplified by repeated choices of the user, which relativizes the performance rating at this level. This avoids time-consuming methods of computation (local search, for example) to focus more on diversity (Carpentier, 2008).

In multiobjective optimization, we always try to find a good approximation of the Pareto front (PF), the set of all non-dominated solutions, that is to say at least as good as all the others on the overall objectives and best on at least one objective. These solutions, by definition, are not comparable with each other, and the user ultimately chooses, involving non-quantifiable criteria. This is a main characteristic of an IGA, whose goal is to find configurations that best meets the user requirements. It is often assumed that these solutions must be found somewhere on the Pareto front, since we claim that the user will necessarily prefer non dominated solutions to dominated ones.

In mixed mode, the user can occasionally let the GA take on pure optimization, without necessarily guiding

the process at each iteration. This back and forth forces the GA to select potential parents in a more diverse population, partly coming from the user's previous choices that can even be somewhat consistent. Finally, to avoid the user fatigue associated with a number of choices over time that may be too high, the IGA should promote rapid convergence toward the efficient frontier while maintaining good diversity within the population and among the optimal solutions (often contradictory).

### Quiroz's IGA

(Quiroz et al, 2008) describe an IGA in cooperative mode, which is based on NSGA-II algorithm (Deb et al., 2002) and conventionally uses a binary Pareto tournament for parent selection. In (Quiroz et al, 2009), they give recommendations to reduce user fatigue and at the same time promote rapid convergence to optimized solutions.

The main originality of their work is to build at each iteration a "virtual interpolated fitness" in order to guide the IGA toward the user choice. This choice is then associated with a maximum virtual fitness, and other individuals are evaluated based on their similarities with it in the criteria space. In this case, the virtual fitness is no longer able to take the objective quality of a solution into account, but only its subjective importance. This approach seems a bit artificial compared to the method derived from Jaszkievicz MOGLS described below.

Another advance of their work relates to the persistence of a solution chosen by the user. To prevent the IGA selections from disappearing too quickly from the population, Quiroz adds additional fitness equal to 1 for a while (which keeps the PF), and then decreases. Through this, he artificially modifies NSGA-II's behaviour to follow a research direction imposed by the user, for a time sufficient to influence populations. But he says nothing about how to take into account several choices.

### Jaszkievicz's MOGLS

MOGLS is a stationary GA (SSGA) for estimating the Pareto front, whose goal is to find a set of good non-dominated solutions using a single fitness randomly combining  $N$  independent targets  $f_i$ . The original algorithm (Mutli Objective Genetic Local Search) is due to (Ishibuchi et al, 1996). At each iteration, after a selection step, a new solution is obtained by crossover and improved by a local search method with the current "scalar function". It replaces in the population the solution of lowest fitness. This scalar function maximizes a weighted sum of the criteria, whose weight set  $\{\lambda\}$  is randomly chosen at each iteration. Each  $\lambda_i$  is given a sign that indicates whether the associated fitness has to be maximized or minimized. In 2002, Jaszkievicz shows that MOGLS, although based on a linear aggregation of the criteria approximating "goal programming" methods, allows access only to convex areas of the PF. Instead, he offers a more efficient version of MOGLS where the

weighted sum is replaced by the “scalar Tchebycheff functions” (1), more suitable than linear ones to access concave areas of the Pareto front (Jaszkiewicz, 2002).

$$\|f\|_{\lambda} = \max_i (\lambda_i \cdot f_i) \text{ with } \sum_i \lambda_i = 1 \quad (1)$$

The use of Tchebycheff functions promotes the search for diversified solutions, the weights  $\{\lambda\}$  being randomly generated by the algorithm in  $[0,1]^n$ . In practice, we evaluate at each iteration normalized fitness between 0 and 1 to overcome amplitude differences, using the lower and upper bounds of each fitness  $f_i$ . By this mean, we can define the Tchebycheff norm (2).

$$\|f\|_{\lambda} = \max_i \left( \lambda_i \cdot \frac{f_i - \min(f_i)}{\max(f_i) - \min(f_i)} \right) \quad (2)$$

In (Carpentier, 2008), the author notes that in this hybrid algorithms class, the genetic part (without mutation) has an exploration role while the local search, devoted to research intensification, is very costly in computation time. So we can decide not to implement it, since maximum optimization is not the first goal in an IGA. Instead, it shows that an operation mode like “evolution strategy ( $\mu + \lambda$ )-ES” allows MOGLS to get excellent results, faster. Recall that in this type of strategy, the parent population of  $\mu + \lambda$  children shall be reduced to  $\mu$  parents at each iteration. In the following, this algorithm implemented in EcCoGen will be called MOGLS- $(\mu + \lambda)$ -ES.

### Directional IGA optimization in MOGLS

(Ishibushi et al., 2002) suggest an improvement in their algorithm using in the local search step an optimization direction for every solution, independent of the current weight set, allowing to separate the genetic evolution from the local exploration. Even if we do not use local search, we were inspired by this improvement for directing the research effort in interactive mode, once the user has designated its preferences. We simply deduct the weights  $\{\lambda'\}$  from the “induced norm of Tchebycheff” (3), which optimizes the classification of criteria  $f_i$  associated with the user choice.

$$\lambda'_k = \frac{\prod_{j \neq k} f_j}{\sum_{i=1}^N \prod_{j \neq i} f_j} \quad (3)$$

Then, we use MOGLS as a series of single-objective optimizations (blocking each new choice values  $\{\lambda'\}$  for subsequent iterations). If the user makes several choices, we get a sampling of the preferences space that is used to build a distribution substituted to the uniform distribution in  $[0,1]^N$ .

### Maintaining population diversity

In mono criterion research, maintaining the population diversity helps avoiding being trapped in local minima, and in multi-criteria search, an additional

objective ensures uniform distribution of solutions along the Pareto front. Indeed, GA have a “natural tendency to drift” toward a particular area of the Pareto frontier (Goldberg, 1989), and to effectively counter this phenomenon, one can act :

- upstream : by encouraging convergence to the Pareto front (or global maximum) while maintaining the population diversity in the fitness space but also at the genetic level. Here we can note that MOGLS is designed to promote the first diversity because it uses a new weighting of fitness at each iteration, while ACROMUSE (see below) rather favours the latter.
- downstream : by using measurements of the local population density in the fitness space to facilitate the exploration of the less frequented zones. This mechanism is activated by penalizing the densest areas in the selection step or remove individuals during the current population upgrade.

One of the best solutions to measure the local density is the non-parametric algorithm PADE (Population Density Estimation Adaptive size, Elaoud et al., 2008). PADE is called to reduce the population size when it exceeds a threshold value which is fixed at the outset. To do this, it divides the fitness space by a hypergrid whose size and cells number are computed from the current population state. Applied with MOGLS, PADE iteratively removes individuals of maximum local density and poorer fitness (2).

### Adaptive Cross-Over Mutation and Selection

ACROMUSE is a recent single-objective GA (McGinley et al., 2011) which automatically adapts crossover / mutation rates and selection pressure depending on the current population state. Its purpose is to maintain a population of individuals both diverse and healthy, able to quickly adapt to fitness landscape changes, particularly multi-modal. This robust method exploits more the research landscape than conventional techniques : one often gets new optima even after hundreds of iterations.

The breeding pool is divided in two populations : individuals which undergo crossover and low mutation rate, and those who are only strongly mutated (figure 2). A genetic measure of the population diversity (SPD) is used to adjust the crossover and high mutation rates, while the selection pressure (number T of individuals involved in the selection tournament) is controlled by another measure (HPD) coupling genetic diversity and performance. The mutation rate is adapted to each individual by taking its performance into account (the less it is, the higher the individual is likely to mutate). The parents selection is based on their local contribution  $hpd$  to the HPD measure, rather than on pure fitness. Thus, ACROMUSE achieves a good balance between exploration and exploitation, but requires the estimation of 3 parameters : the maximum values of SDP and HPD (SPD\_max, HPD\_max) and the maximum selection pressure (the authors use  $T_{max} = |P| / k$ , k being blocked at 6.

### Using GA as a constraint solvers

GA usually do not transmit consistency (constraints imposed) : consistent parents give little consistent children. To address this shortcoming, one can use repair mechanisms (time-consuming, penalizing evolving capacities), ad-hoc genetic operators that guarantee the consistency of the produced solutions, but lose their effectiveness when the number of constraints increases, or the strain relief, which is a tolerance on which to act. Another method is to turn constraints into fitness, which increases the complexity in terms of calculations, makes the PF size grow, and fosters the emergence of random or uninteresting solutions.

Best results are obtained when a certain amount of inconsistency is tolerated within the population, which does not disturb too much the evolution. It often achieves to reduce the inconsistency rate by introducing a fitness penalty function  $z$ .

A good way to deal with this problem is to integrate the consistency notion within the dominance relationship (Deb, 2011), conducting selection tournaments based on fitness or on the minimization of the inconsistency. An even recent method redefines the dominance relation with constraint-dominance (Coello, 2007). With MOGLS, since it does not directly convoke a dominance function, we use the current scalar function  $f(x)$  defined by the Tchebycheff norm in order to penalize the fitness (4) during the selection step (in maximization) :

$$f_z(x) = f(x) \text{ if } z(x) = 0, \text{ else } \min(f) - z(x) \quad (4)$$

### ORIGINAL DEVELOPMENT OF AN IGA

The MOGLS-based method adapted by Carpentier was chosen in preference to Quiroz NSGA-II-based method. It is somewhat simpler to implement, allows natural implementation of ACROMUSE and provides more flexibility to manage a research effort from the user's choices in interactive mode.

Our contribution is threefold : MOGLS hybridization with ACROMUSE, extension of ACROMUSE capabilities to the multiobjective case (already favoured by MOGLS) with significant improvements in its operation, and finally development of a pheromone learning technique in interactive mode.

**Components.** The algorithm that we designed consists of five modules : Jaskiewicz MOGLS- $(\mu + \lambda)$ -ES (amended by Carpentier), ACROMUSE, PADE, a pheromone IGA and a constraint satisfaction one (Figure 2). The IGA pheromone is our largest contribution, but some module arrangements and substantial improvements are also part of it.

**Genomic structure.** The genetic coding assumes that all CV units are indexed. Each index is allocated : a "form gene" which is a binary occupancy index (0 or 1) and a "function gene" (1 = office, 2 = activities, 3 = housing, 4 = annex space). Other genes encode facade properties. Such a structure preserves the genetic patterns and topological consistency when

applying variation operators. Crossover and mutations (swap, insertion, deletion, random shift) affect the same locations along the chromosomes, so the result is still inside the CV. Since ACROMUSE is based on genetic diversity, we choose a representation in several chromosomes (here, two chromosomes for genes form and function). This also allows to separate the shape evolution from the functions evolution and clarify the design of mutation operators.

**Populations.** The GA manages three populations : the current one (P), the local Pareto front (non-dominated elites of the run) and the global Pareto front (elites collected on all runs based on the same set of parameters). The initial population is created pseudo-consistent regarding to the programmatic constraint. It can also be locked to the global Pareto front, for advanced optimizations from previous best solutions.

Our approach is not to calculate the optimal solutions in a large number of runs, as it is performed in (Hamdy et al., 2012). Instead, we focus on a broad exploration of the Pareto front during each run (thanks to ACROMUSE), where important decisions are made by the architect during interaction.

**Population Size.** In a GA based on a mesh representation of size N, (Cerf, 1994) showed that the size of the population P varies substantially linearly with N (number of genes). Thereby, P varies from 50 to 300 individuals in our experiments.

**Constraints.** A programmatic constraint deals with the volume or surface-to-build specification, together with a possible tolerance. To resolve this, we define the penalty function  $z$  used in formula (4) as the difference between the imposed surface and its measured value for a solution. A "form" constraint is also treated in interactive mode (see §IGA).

### ACROMUSE

**Multiobjective extension.** In order to use Acromuse in multi-objective mode, we have replaced its single fitness by the scalar function of MOGLS, with some minor adjustments.

**Selection.** Although the binary tournament ( $T = 2$ ) has been shown to be the best for Jaskiewicz MOGLS, this is not true for the MOGLS- $(\mu + \lambda)$ -ES without local search and ACROMUSE hybridization. T individuals are still randomly taken from the population, with replacement, but the best is selected according to its *hpd* contribution. Crossover and mutation operators are then applied to the candidates from the pool, and new individuals are integrated into the current population. In all our tests, the size of the breeding pool was set to 50 in order to avoid high computation time per iteration. We also worked on the best settings for the parameter  $k$  involved in Acromuse selection pressure formula  $|P|/k$ , through many tests. This setting does not really seem to be problem dependent.

**Crossover and mutation.** We obtain good results with the uniform crossover (random but aligned genes inversion on parent chromosomes), the multipoint mutation, and with the form and function genes

mutated and cross-overed separately.

We have shown that the SPD/HPD measures should be made after the reduction step (fig. 2), a point not clearly specified by McGinley in his paper. We have also proved it is better, in order to accelerate the convergence, to use dynamic values of SPD\_max and HPD\_max, updated at each iteration according to their highest values on the run. These two improvements lead to faster convergence to the Pareto frontier.

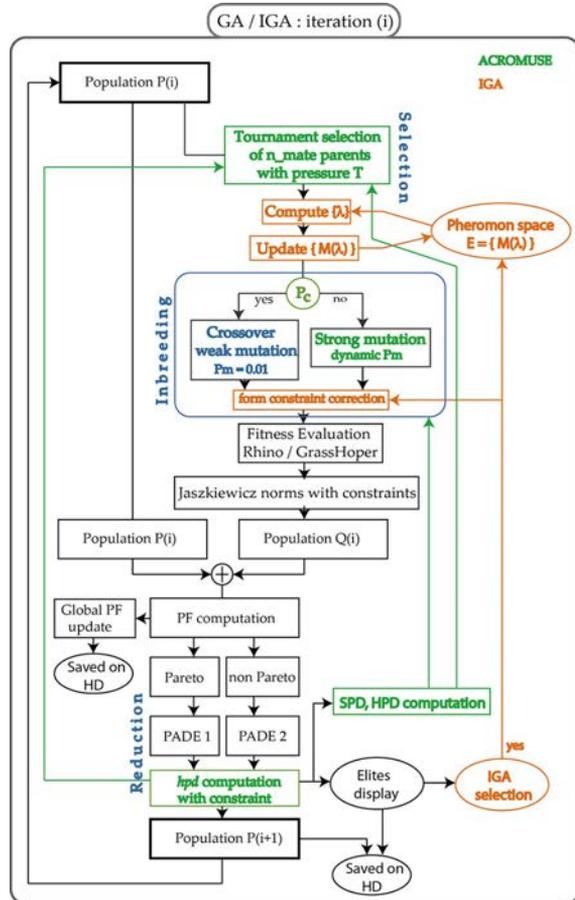


Figure 2 IGA framework

Regardless of the Deb technique (4) implementation to adapt the inconsistency fitness, the GA should be able to adjust its mutation rates to limit the number of highly inconsistent solutions that penalize the population evolution and slow convergence.

Certainly, increasing tolerance reduces inconsistency and allows faster convergence, but this is not the right way. When mutations promote too much inconsistency (e.g. gene form), an adaptive decrease of insertion/deletion mutation rate is operated in favour of an increase of the swap mutation rate (figure 2 : dynamic  $P_m$ ).

**Replacement.** The replacement step invokes PADE algorithm in a way close to the method used in (Carpentier, 2008). But the withdrawal is made by randomly removing in a grid-box of maximum density an individual with lowest *hpd* (instead of the current scalar function). We begin by first applying PADE to the Pareto front if its size exceeds a fixed threshold

currently equal to 40% of  $|P|$ . Then, we consider the non-Pareto population. Either we iteratively remove the least consistent solutions to return to the desired population size, or the number of solutions of minimal consistency always exceeds the size limit, and the population is PADE-reduced.

**Fitness functions.** EcCoGen currently deals with up to five fitness fast-computed with reduced models within *Rhinoceros/Grasshopper* software: 1-low winter consumption, 2-high compactness, 3-low shadow casting on the neighborhood, 4-summer thermal comfort, 5-luminous comfort. Figure 3 shows an evaluation of fitness (1,2,3).

**IGA.** A good IGA must let a population evolve while resembling to a space of solutions chosen by the user during the run, even without knowing his motives (implicit learning). It is well known that the shape criterion is essential for an architect. So we use it in conjunction with the performance measured by the fitness to establish research directions in the privileged landscape solutions. This can be achieved quite simply by providing the stochastic distribution of induced  $\{\lambda\}$  with pheromone markers  $M(\lambda)$ .

Recall that in nature, ants move and direct depositing pheromones (volatile olfactory substances that play the role of collective memory). This concept forms the basis of the ACO meta-heuristic (Ant Colony Optimization), applied in combinatorial optimization, using pheromone tracks to mark elements promoting the best solutions (Angus et al., 2009).

We have adopted the following principle: when a solution is chosen by the user, it is stored with its induced  $\{\lambda\}$  and associated form in a distribution space  $E$ . We define a function  $Resemblance(\lambda)$  which compares the occupancy genes of a solution  $CV$  with those of the form associated with  $\{\lambda\}$ . The overall similarity is nothing more than the sum over  $E$  of the product  $M(\lambda)$ .  $Resemblance(\lambda)$ .

During each IGA iteration, a single pair  $(\lambda, form)$  is selected in  $E$  (as in ACO) with a random stochastic roulette based on markers  $M$ . Then, we let the GA follow its normal process, but we manage to satisfy the overall similarity constraint. To do this effectively, we define an “ad-hoc” variation operator that directly provides consistent solutions, that is to say whose overall similarity is at least equal to the threshold of 72%. Our tests show that below this threshold, the similarity is lost quite quickly in the next generations.

The interface displays this global similarity index for each elite, which can help the architect making new relevant choices. Indeed, if the set of similarity indices displayed during several successive iterations is bad, it tells the user that the choices are not consistent with the previous, which can there again help to change selections. Finally, the user may convoke an additional pheromone track for function genes, in order to force the GA considering more precise functional choices in addition to the form preferences.

The pheromone award is based on the choices repetition in E and not on performance (which the weight  $\{\lambda\}$  already takes into account). This takes place at the end of each iteration, then markers  $M$  are updated by applying the evaporation mechanism (5), whose rate  $\varepsilon$  varies from 0.05 to 0.1 :

$$M = M(1 - \varepsilon) \quad (5)$$

This mechanism strengthens repeated user choices and all solutions that are close ; it lowers the influence of old choices. When the IGA mode is not activated, the markers are useless and just undergo evaporation.

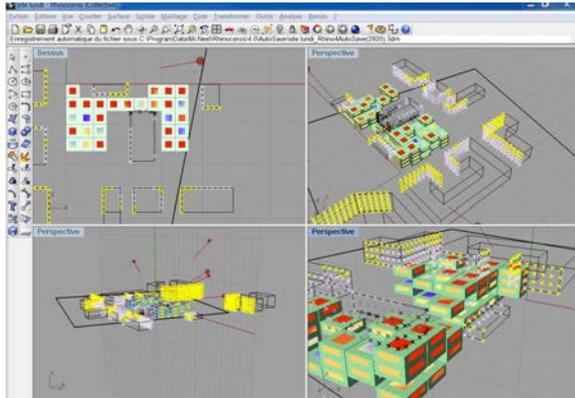


Figure 3 Fitness evaluation within Rhinoceros

### Conception of an intuitive GUI

The Graphical User Interface, developed by Renato Saleri (MAP-ARIA) is organised in two screens. The main one allows the elite population visualization, while the second one zooms in the phenotype representation. The first screen is divided in three main parts (Figure 4) : the current elites population, the selected individuals collection and the algorithm preferences composed of evaluation parameters, constraints values and evolution monitoring.

The architect has the possibility to select one or more individuals and to keep them available for subsequent manipulations. These selected solutions constitute a collection. At any time during the IGA process, the architect can export them or inject them inside the evolutionary loop in order to redirect the optimization, to re-balance the Pareto front by favoring these new entering.

The zoom window displays the phenotype representation integrated inside the urban context ; it is possible to manipulate the 3D model in rotation / scale and to display the performance outlines (Figure 5). These kind of information allow both a subjective interpretation and an access to an objective knowledge, the relative and the absolute performance of the analogon.

The GUI easily allows to control the evolution of the genetic process and its numerous options. It operates on three populations :

- the current population P, viewable on demand,
- the selection S of “heart-favoured” solutions, of

small size (12 squares on right column) : this is a privileged elites population,

- the global Pareto front of all runs started with the same parameters, viewable and linkable on demand.
- Every K iterations (K being tuned to reduce the waiting time of the user), the interface displays a subset A of nine elite solutions of the local Pareto front, distinct (if possible) and diverse.

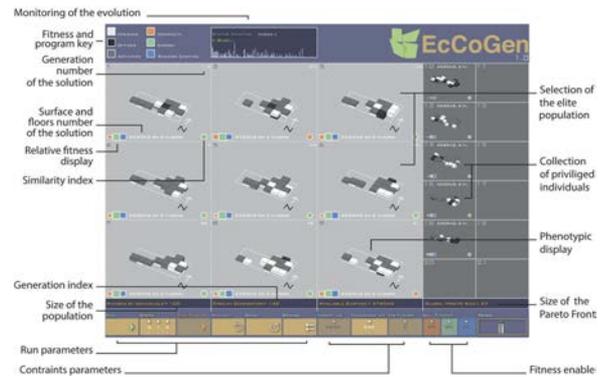


Figure 4 Main screen with elite and privileged population visualisation and GA monitoring.

With a simple “drag and drop”, elites can be transferred from A or F to S (preferential memory), or from S to E (hybridization, recovery). Once the user selects one or more solutions among A and S elites, the process switches to IGA mode, and the algorithm operates in the  $\{\lambda\}$  distribution induced by the multiple choices stored in E and the formal similarity space. It is possible to reverse the process, when the users encounters a deadlock (no more interesting proposals for several iterations), or when there is no more possible optimization (a GA which has converged has virtually no chance to diversify ! Even with MOGLS hybridized by ACROMUSE after a few hundred iterations). The process being deterministic (it depends on a seed stored with each run), a “restart” button enables to start again with the same initial population (same seed and settings). You can also use the “reset” button to start with a different initial population (new seed) and change the settings.

Each solution is legended (Figure 4) by :

- a series of scaled color squares for the visualization of its fitness values relative to the best ones obtained from the beginning of the run for all consistent solutions,
- a color indicator of similarity / consistency with previous IGA choices to visualize at a glance the local or global relevance of solutions,
- its age (the iteration index which generated it).

Finally, the user can continuously monitor the evolution progress through a dual graph displaying a mix of SPD / HPD values and a “generational distance” showing the local PF improvement (Deb, 2011). The latter is based on a comparison between

two successive PF in the fitness space, which just sum the Euclidean distances between the closest elements in both fronts. This measure is robust : it works even if the fronts are of different sizes.

## RESULTS AND DISCUSSION

Figure 6 shows an output of the evaluation and classification of potential solutions through a real time comparison of the whole building's energy rating, its shading impact in the urban fabric and its compact shape. The program is built on two floors with an imposed constructive area of 2520m<sup>2</sup> and a 240m<sup>2</sup> tolerance. The user has selected four preferences during the first IGA generations (right-up), and figure 6 displays 34 well diversified individuals obtained from the local Pareto set after 300 iterations (which is a minimum required for correct convergence).



Figure 5 Interaction and performance visualisation.

It is difficult to compare our algorithm (complex by its modular nature and the ACROMUSE add-on which operates as a good explorer tool) with other existing GA. However, a case study is still conducted and will be published in a peer reviewed journal. Preliminary results show that evolution can be oriented in more varied ways than usual techniques like the NSGA-II implementation of Quiroz, with slower convergence.

It is undeniable that ACROMUSE integration and the improvements that we have made are a strong progress in this work oriented towards creativity and performance, with a good ability to optimize in diversity. This has been thoroughly tested in single-objective and multi-objective modes.

Our IGA module, meanwhile, also benefits from this diversity, although it is oriented by the user in “niche preferences”. An intensive study questioning if the architect's creativity is enhanced by EcCoGen has been carried out by our partner Codysant-Interpsy from French Lorraine University (<http://www.aria.archi.fr/blog/wp-content/uploads/2013/04/Evaluation-Creativite-EcCoGen.pdf>). It confirms from the first user feedbacks the software potential to assist the decision-making processes of the designers, preserving at the same time their creativity. Partly because the GUI

allows to focus on the design without worrying about the model source code, the architect can interactively design building forms being more aware of the energy performance features and their impact on the energy consumption of the final building. We also observed that the genetic algorithm enriches the variety of the environmental-friendly solutions (Figure 6).



Figure 6 Example of Pareto front from an IGA optimization on 3 energy fitness

## CONCLUSION

In this project, we have developed a multi-objective and interactive genetic algorithm which meets important requirements : diversity, performance and adaptation to an architect's subjective choices in the sketching or design stages. The case study was based on three fitness related to the minimization of energy consumption, but not yet incorporated the comfort parameters or embodied energy.

From the evolutionary point of view, it is rare that an initial shape optimization leads to a qualitative jump when morphogenesis was not designed to do so. A promising line of research is the “proteome concept” (Lefort-Mathivet, 2007), which consists in separating the evolutionary process into three areas, as in biology : genes, proteins and protein products. This could be applied to a building as follows : proteins represent materials, assemblies, methods, construction rules or architectural techniques ; while an intelligent genes selection encode these proteins, and qualitatively new solutions can emerge from the GA

behaviour through the intermediary of the proteome.

From the interaction point of view, the ability to manually change the phenotype of a solution (e.g. blocking positions in CV) is desirable but has not yet been addressed in our IGA.

Finally, we have tried to increase the convergence speed without sacrificing the diversity needed to foster creativity. This delicate issue should be further investigated, although the use of highly parallelized GPU makes it possible to significantly accelerate the most consuming step of a GA : fitness evaluation. Depending on the case, (Master et al., 2009) speak of impressive factors ranging from 7 to 1000.

### ACKNOWLEDGEMENT

This work has been conducted in the framework of the project “*EcCoGen : Generative Eco-conception*” partly funded by French National Research Agency (ANR-10-Creation-012).

### REFERENCES

- Angus D., Woodward C., 2009. Multiple objective ant colony optimization, *Swarm Intelligence*, Springer, vol. 3, p. 69-85.
- Attia, S., Gratia, E., De Herde, A., Hensen, J., 2012. Simulation-based decision support tool for early stages of zero-energy building design, energy and buildings. ISSN 0378-7788.
- Baverel, O., Peupartier, B., Rivallain, M., 2012. Decision aiding & multi criteria genetic optimization for existing buildings holistic energy retrofit, Symposium *Materiality in its contemporary forms*, Ile d'Abeau, France.
- Carpentier, G., 2008. Approche computationnelle de l'orchestration musicale, optimisation multicritère sous contraintes de combinaisons instrumentales dans de grandes banques de sons, IRCAM, Thèse de l'Université Paris VI.
- Cerf, R., 1994. Une théorie asymptotique des algorithmes génétiques. PhD thesis, Université de Montpellier II, France.
- Coello, C.A., Van Veldhuizen, D.A., Lamont, G.B., 2007. Evolutionary Algorithms for Solving Multi-Objective Problems. Kluwer Academic Publishers, Springer.
- Deb, K., Agrawal, S., Pratap, A., Meyarivan, T., 2002. A fast and elitist Multi-objective Genetic Algorithm : NSGA-II, *IEEE transactions on evolutionary computation*, vol 6, n°2.
- Deb, K., 2011. Multi-Objective Optimization Using Evolutionary Algorithms: an Introduction, Department of Mechanical Engineering, KanGAL Report Number 2011003, Indian Institute of Technology Kanpur.
- Elaoud, S., Loukil, T., Teghem, J., 2007. The Pareto Fitness Genetic Algorithm : Test Function Study, *European Journal of Operational Research*, no. 177, 1703–1719.
- Goldberg, D.E., 1989. GA in Search, Optimization and Machine Learning. Addison Wesley.
- Hamdy M., Palonen M., Hasan A., 2012. Implementation of Pareto-Archive NSGA-II Algorithms to a nearly-Zero-Energy Building Optimization Problem. The first Simulation and Optimization Conference (BSO12). IBPSA-England, Loughborough University, UK.
- Ishibuchi, H., Murata, T., 1996. Multi-Objective Genetic Local Search Algorithm, *Proc. of 3rd IEEE International Conference on Evolutionary Computation*, pp. 119–124.
- Ishibuchi H., Yoshida, T., Murata, T., 2002. Balance between genetic and local search in hybrid evolutionary optimization algorithm.
- Jaszkiewicz, A., 2002. Genetic local search for multiple objective combinatorial optimization. *European Journal of Operational Research*.
- Kicinger R., Arciszewski T., 2006. Empirical analysis of memetic algorithms for conceptual design of steel structural systems in tall buildings, George Mason University, USA.
- Lefort-Mathivet, V., 2007. Evolution de second ordre et algorithmes évolutionnaires : l'algorithme RBF-Gene, Thèse de Doctorat de l'Université de Lyon, INSA de Lyon.
- Maitre, O., Lachiche, N., Clauss, P., Baumes, L., Corma, A., Collet, P., 2009. Efficient Parallel Implementation of Evolutionary Algorithms on GPGPU Cards, EuroPar Parallel Processing, Springer LNCS 5704.
- Mavromatidis, L.E., Bykalyuk, A., Lequay, H., 2013. Development of polynomial regression models for composite dynamic envelopes and thermal performance forecasting. *Applied Energy*, Vol. 104, April 2013, pp. 379-391.
- Mc Ginley, B., Maher, J., O'Riordan, C., Morgan, 2011. Maintaining Healthy Population Diversity Using Adaptive Crossover, Mutation and Selection (Acromuse), *IEEE Transactions on Evolutionary Computation Volume:15 Issue:5*.
- Quiroz, J.C., Sushil, J.L., Banerjee, J.L., 2008. A model of creative design using collaborative interactive genetic algorithms, *Design Computing and Cognition DCC'08*, Springer.
- Quiroz, J.C., Sushil, J.L., Banerjee, J.L., Dascalu, S.M., 2009. Reducing User Fatigue in Interactive Genetic Algorithms by Evaluation of Population Subsets, *Rapport IEEE*.
- Mourshed, D. Kelliher, M. Keane, 2003. ArDOT: a tool to optimize environmental design of buildings, Proc. of 8<sup>th</sup> IBPSA conference: Building Simulation, Eindhoven, Netherlands.
- Shi, X., Yang, W., 2013. Performance-driven architectural design and optimization technique from a perspective of architects, *Automation in Construction*, ISSN 0926-5805.