A NEW TOOL FOR DESIGNING COST EFFECTIVE LOW ENERGY HOUSES IN FRANCE

Serge Chardon[1*], Emmanuel Bozonnet[1], Robert Montecot[2] and Christian Inard[1]
[1]Laboratory of Engineering Sciences for the Environment, University of La Rochelle, France
[2]PROGEMI, Software development company, Saujon, France
*serge.chardon@univ-lr.fr

ABSTRACT

This paper introduces a new integrated building optimisation tool for low energy cost effective house design. At this stage it contains an optimisation module coupled to a building regulation program for energy needs calculation. The final tool will comprise the optimisation of both cost, assessed by a program already used by several stakeholders in France, and energy performance. The full methodology is described in this paper and potential developments of the tool are discussed. A first single objective genetic algorithm is analysed based on the optimisation of a typical building case.

INTRODUCTION

Constraints on energy supply, climate change and other environmental concerns drove the European Union to implement the so-called "20-20-20" targets requiring a decrease of greenhouse gas emissions by 20 % compared to 1990 levels, a 20 % increase in energy efficiency and raising the share of renewable energies to 20 % of the total final energy produced. Buildings are a key element in that respect since they account for approximately 40 % of the energy consumed in the EU and 36 % of its CO_2 emissions. Designing low energy buildings is hence a necessity for the success of its climate and energy policy.

The recent recast of the European Energy Performance of Buildings Directive (EPBD-Recast) requires new buildings to be Net Zero Energy Buildings (NZEB) by 2020, meaning that they should have a "very high energy performance" and that "the very low amount of energy required should be covered to a very significant extent by renewable energies" (Erhorn and Erhorn-Kluttig, 2013). Achieving such energy performance goals was demonstrated technically feasible in several studies but life cycle costs are not always advantageous as compared to less energy efficient buildings (Leckner and Zmeureanu, 2011; Marszal and Heiselberg, 2011).

In France, the previous EPBD has been fully implemented in 2013 through the new energy regulation for buildings called RT 2012. It is already quite challenging for house builders since it requires a primary energy consumption of 50 kWh/m²/year for all new buildings (Roger et al., 2012). Studies showed that such energy performance requirements entailed a final construction cost increase of 10 % to 20 % compared to previous state of the art buildings while resulting in life cycle cost savings after 20 years (ADEME, 2010; French Ministry of Ecology, Energy and Sustainable Development, 2010; Vergne, 2011). Designing cost effective low energy buildings hence increasingly draws house builders' attention because it limits cost increase due to regulations while adding value to their products thanks to lower spending on energy.

However, this reveals to be a complex issue. The number of parameters and products that can be varied is simply too vast to be entirely explored for each project by a trial and error procedure. Furthermore, this would require involving several tools from architectural sketches to cost calculation and energy simulation programs. A way to overcome these issues is to use automated optimisation and integrated design tools. In the research domain, automated optimisation for building design has been widely studied during the last two decades. It covers works with various goals such as "one shot" optimisation studies to give guidelines for building regulations (Hamdy et al., 2013), optimal system designs (Wright and Zhang, 2005) or real time optimisation for smart monitoring of buildings for instance (Kolokotsa et al., 2011). Evins (Evins, 2013) and Attia et al. (Attia et al., 2013) give comprehensive reviews of respectively 74 and 165 publications in this field.

In this paper study, we introduce a new optimisation tool for building design coupled to a building regulation program. The module is currently being integrated to the cost calculation tool ADR, developed by the local company PROGEMI (PROGEMI, 2014) and used in France by several stakeholders. The final aim of this work is to help house builders to find optimum compromises between energy performance and cost in the design phase of their projects. In the first part, the tools used are detailed. Their coupling is analysed in the methodology section and some first results of a single objective optimisation algorithm are analysed and discussed.

TOOLS

Building regulation program for energy performance calculation

The French building regulation (RT 2012) for energy performance requires three main indicators to be calculated: the building energy needs (called B_{bio}), the indoor temperature in summer during five consecutive days of hot weather (called T_{ic}) and the building primary energy consumption (called C_{ep}) for heating, cooling, domestic hot water, lighting and auxiliaries (pumps and fans). To comply with regulation, these values must not exceed thresholds that depend on regional climates. Because these parameters depend on many hypotheses such as occupancy, climate and temperature settings for instance, a specific building regulation program is provided by the authorities for their calculation. This RT 2012 program is a building simulation tool that runs with a hourly timestep where all regulatory hypotheses are fixed. It enables bioclimatic conception since parameters such as weather data, orientation and natural light are taken into account (Roger et al. 2012). It also ensures that different designers get the same results for a given project and hence makes all projects comparable. In practice, the RT 2012 program is a dynamic link library that can be ran by calling objects from a script or simply through an XML input file. This latter option was faster to implement and hence was chosen for this work. The input file contains all building envelope information, direct environment like trees, and systems. Each regulatory parameter can be calculated separately.

Cost calculation tool

Several studies use cost as an objective function in the optimisation process. Different types of costs can be used from construction cost, e.g. (Chantrelle et al., 2011; Hamdy et al., 2011), to life cycle cost, e.g. (Fesanghary et al., 2012; Verbeeck and Hens, 2007; Wang et al., 2005). The latter includes operation and maintenance cost which can be quite significant for buildings due to energy expenses. In any case the construction cost has to be calculated. In most studies, construction costs are determined using standard databases. This approach cannot be used here because the final aim is to make a practical tool for house designers. Different house builders will have different material suppliers, subcontractors and construction workers, which will entail different costs. The cost calculation tool developed by the local software development company PROGEMI is used. It enables house builders to assess the cost of their projects precisely in three phases.

First, a user interface enables house builders to input their own cost databases. These include material costs, subcontractors, artisans, margins, taxes etc. A second interface enables defining a building project through a selection of geometry, size, building components and systems. Finally the construction cost calculation can be processed. Several house builders have used this system for more than a decade and it has shown to be very efficient for precise cost analysis.

Optimisation module

Genetic algorithms have proven suitable and robust when applied to building design. Like other metaheuristics, they can tackle mixed integer non-linear optimisation problems with black-box objective functions and constraints. These problems belong to the NP-hard problem class which can only be handled by a few groups of methods, especially for those with large design spaces (Teghem, 2012). In their reviews, Evins and Attia et. al agreed on the genetic algorithms efficiency with regards to building design optimisation for the following reasons:

- They do not require any specific knowledge of the objective function, contrary to many heuristic or direct search methods.
- They are faster and more robust than other metaheuristic methods (Tuhus-Dubrow and Krarti, 2010), (Brownlee et al., 2011).
- They do not get trapped in local minima.
- They are relatively easy to implement.

They were therefore chosen for this study. The optimisation module contains a single objective genetic algorithm and a multi-objective Non-dominated Sorting Genetic Algorithm (NSGA-II) is currently being implemented. The NSGA-II was shown to perform better than other multi-objective evolutionary algorithms (Deb et al., 2002). The final aim of the optimisation module is to enable solving both single and multi-objective optimisation problems.

Evolutionary algorithms are all based on Darwin's principle of evolution. A population of individuals submitted to a hostile environment evolves through generations under the natural selection law. The general idea is that the fittest individuals have more chances to survive and reproduce, hence to transmit their genes to the next generation. After several generations, the population is fitter than the initial one. Gene mutation is added to the process and enables developing new individuals as well.

In practice, setting up a genetic algorithm first requires defining a way to represent design variables, also called encoding. Because most design parameters related to buildings are discrete, an integer representation was chosen. The principle is the following: if the search space contains four windows and three insulation thicknesses for instance, integers from 0 to 3 will represent windows, and integers from 0 to 2 will represent insulation thicknesses. With this encoding and a reference building, a list of two integer values enables representing a full building alternative design. For instance the list [2 0] represents a building where the

third window type and the first insulation thickness are used. Such a list is called individual. Each integer value is called a gene. In order to communicate with other programs, the optimisation module also includes a decoding phase where variables' values can be generated from an individual.

Then, just like in Darwin's theory, the evolution process is directed by the principles of selection, reproduction and mutation of a population of individuals. In the case of single objective genetic algorithms, selection is based on the objective function values also called fitness values. The fittest individuals have the highest chances of being selected for reproduction and can be selected more than once. Reproduction consists of generating a new population from the selected individuals. Their genes are combined to form a new population. Finally, some individuals are mutated by modifying some of their genes. After a given number of generations, the population is composed of fitter individuals than the initial one. For NSGA-II, a new fitness function is created. It sorts individuals with regard to their belonging to consecutive Pareto fronts. Individuals that belong to the first Pareto fronts have higher chances to be selected for reproduction (Deb et al., 2002).

A Python package called Deap was used to implement these algorithms. It contains a toolbox of all selection, mutation and reproduction operators needed for both single and multi-objective problems (Fortin et al., 2012). It was slightly modified in order to enable integer representation as opposed to binary. Designing a full genetic algorithm from scratch requires experience in the field since several parameters have to be set like probabilities of selection, mutation and reproduction. These parameters are problem dependant and no rule exists on how to fix them. The goal when setting them is to ensure that two principles are respected: diversification and intensification. The algorithm has to explore the design space as much as possible (diversification) while detailing zones with fit individuals (intensification). Here, values where set based on examples available in Deap documentation.

METHODOLOGY

In order to make multi objective optimisation possible, an integrated approach is required. Building data have to be inputted once and used both for cost and energy calculations. This ensures that an exact same project is assessed in both processes. Figure 1 shows how the different programs interact to achieve such integration. It contains three phases: initial design and configuration, optimisation loop and results. These are described in the following sections.

Initial design

The first phase consists of inputting building project data. This is done through the cost calculation tool user interface. It is modified to include not only databases and parameters related to costs, building geometry and components but also to energy performance assessment such as thermal transmittance or solar gain coefficients for instance. Once a project is inputted, a building file is generated. It corresponds to the reference building on which all modifications are made during the optimisation phase. Another piece of information has to be inputted at this stage: the design variables search space. It corresponds to a choice of parameters that will be varied during the optimisation process. Usually, they will correspond to libraries of components like windows, heating systems, insulation types, ceiling and floor types for instance. They are stored in a file containing a list of lists corresponding to each variable values e.g. windows, floors, heating systems, geometry etc.

Optimisation loop and final results

Once the reference building and design variables files are complete, the optimisation module starts. First, it reads the design variables search space file. The module encodes the variables by generating a list of integers that represent the number of alternatives possible per variable. This list is used as a mould to create individuals, i.e. lists of integers whose values are comprised between 0 and the corresponding integer values of the mould list.

The genetic algorithm generates an initial population. Each individual of that population follows the same process. The individual is decoded and a building design variable file is generated. An integrated program that contains both the cost assessment tool and a file format converter reads the file. It generates a new building project by merging a copy of the reference building file with the new building variables values. The cost of the new building project is assessed and written in an output file. In the meantime, the building project is written in a RT 2012 file format and processed through the RT 2012 building regulation program. Another output file is generated containing energy performance results. These results are read and saved in the optimisation module. This loop is repeated until all individuals of the population are evaluated.

At this stage, the population evolution can take place. A new population is generated by the algorithm using selection, reproduction and mutation operators. All individuals are evaluated following the same procedure than the initial population. After a predetermined number of generations the optimisation loop is stopped and results are written in a file. It contains all individuals and their cost and energy performance values. The results treatment has not been fully determined yet but it has to be a key element of the tool. It should enable representing Pareto fronts in case of multi-objective optimisation.

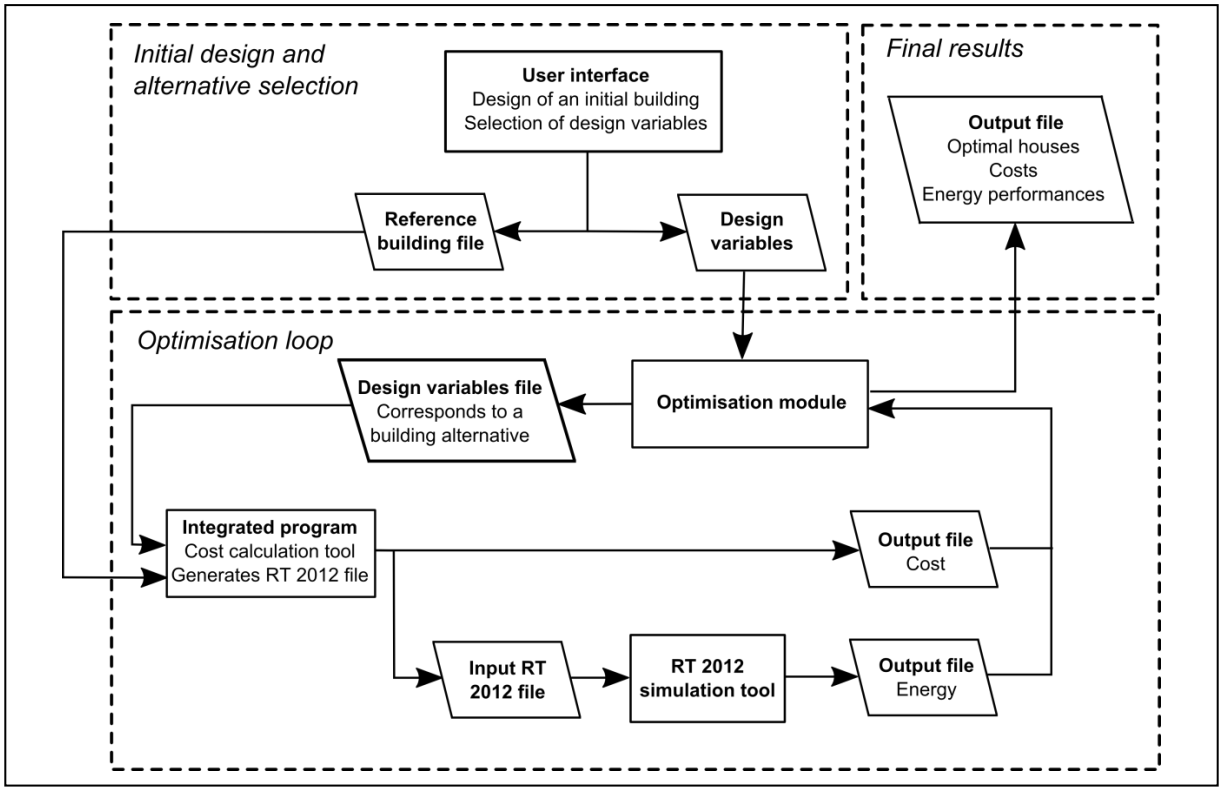

Figure 1 Methodology diagram

SINGLE OBJECTIVE OPTIMISATION STUDY

Most of the methodology represented on figure 1 has been implemented. The use of the cost assessment tool from a design variable file is the only part that remains to be done. It requires identifying what variables are likely to be modified and determining where they appear in the program. In order to validate the methodology, a first single objective optimisation algorithm was implemented. It focuses on energy performance and does not include the cost calculation tool. The differences with the diagram represented on figure 1 are that the integrated program only contains the module that generates the RT 2012 input file and that the output cost file is not generated.

Objective function and constraints

In this study, the B_{bio} regulatory indicator is set as the objective function to be minimised. It corresponds to a calculation of building energy needs, representative of the building envelope performance. It takes into account heat losses through walls, windows and thermal bridges, thermal inertia, solar and internal gains, natural lighting and heat losses due to air infiltration and mechanical ventilation. Then, several regulatory hypotheses are included in the program. It assumes for example that ventilation is provided by a mechanical ventilation system with a 50 % heat recovery efficiency. The B_{bio} value is therefore independent of the system choices. It is expressed without units and is a weighted indicator, similarly to primary energy factors: heating and cooling needs are multiplied by a factor of 2 while artificial lighting is multiplied by a factor of 5. Using the B_{bio} calculation as the objective function enables us not considering systems while still being meaningful in terms of energy efficient envelope design. Another advantage of this regulatory index is that all hypotheses are fixed. For this study, the only constraint set was the window area. It was required to be larger than one sixth of the total living area which is mandatory in the French building regulation. The constraint was directly handled in the optimisation module.

Case study and design variables

The building used is a 100 m² one storey house with a rectangular floor plan. Because the aim of this first work is to validate the method and algorithm, using such a simple building design seemed appropriate. A sketch of the house is shown on figure 2. Shadings are not represented on the sketch.

Figure 2 Reference house design

The RT 2012 file for this building contains around 200 input parameters (geometrical parameters,

thermal transmittances, thermal bridges values, shadings, control parameters etc.). Several building components variations are considered in this study. These design variables are listed in table 1. They include window surfaces, window and wall types and wall, floor and ceiling insulation thicknesses. The house compactness is also taken into account with the "wall area" variable. The smallest value corresponds to a square house while the highest one corresponds to a longer rectangular house. Several optimisation calculations were ran for two climates: north and south of France. The results are discussed in the following section.

Results

After several tests, using a population of 15 individuals and 60 generations always seemed to reach near-optimum solutions. Figure 3 shows the B_{bio} values of the best solutions found at each generation for the northern France simulation. As one would have expected the best designs were the ones with the highest levels of insulation i.e. 20 to 30 cm for the external wall, 50 cm for the ceiling and 20 cm for the floor. In all simulations, the wall type selected for the best building design was the aerated concrete blocks wall, most of the time with a resistance value of 2.27 m²K/W. The most compact alternative was also often found by the algorithm. Concerning windows, the algorithm chose the ones with the lowest thermal transmittances and minimised their surface while respecting the constraint. Windows facing south were kept large while others were set to minimum. In both cases, shadings were initially inputted and conservative values were used for solar gains. Had it not been the case, higher cooling needs would have been experienced and may have entailed a smaller south window surface, especially for the southern France case.

While the building designs obtained by the optimisation process seem rather obvious at this point, the algorithm performance results are promising. First, the building energy needs decrease substantially during the optimisation process. Only the design variables that have the least influence on the objective function may not always have optimal values. For instance, here, the compactness and window surfaces values chosen by the algorithm may not be the best but insulation thicknesses always are. In terms of number of simulations, the algorithm requires between 500 and 600 simulations, which takes around 90 minutes to complete. A brute force search algorithm where all combinations possible are simulated would require more than 3 million simulations (6x7x7x6x2x2x3x3x5x2x5 = 3 175 200 cases in table 1) and take months to complete (approximately 13s per simulation). The algorithm is hence rather robust and fast.

However, these results show the limitations of focusing only on energy needs when designing a building. The solutions obtained are not necessarily optimal with regard to other considerations such as thermal comfort, cost or architectural aspects. It is therefore crucial to include other objectives and constraints in the optimisation process. If costs were added for example, the building designs with the highest levels of insulation may not be located on the optimal Pareto front. Including cost in the objectives will also entail the use of real products in the optimisation process. Here, the optimisation was based on hypothetical window surface values but once the cost calculation tool is coupled to the

Table 1 Design variables

	Alternatives						
Design variables	Reference	No. 1	No. 2	No. 3	No. 4	No. 5	No. 6
Area windows N	2.25 m²	0 m²	1 m²	3 m²	4 m²	5 m²	
Area windows W	4.5 m²	0 m²	3 m²	4 m²	5 m²	6 m²	7 m²
Area windows E	4.5 m²	0 m²	3 m²	4 m²	5 m²	6 m²	7 m²
Area windows S	4 m²	0 m²	3 m²	5 m²	6 m²	7 m²	
***Uw* window N**	*Uw* = 1.3 W/m².K	*Uw* = 1.5 W/m².K					
***Uw* W, E and S**	*Uw* = 1.5 W/m².K	*Uw* = 2 W/m².K					
Wall area	104.5 m²	101.75 m²	110 m²				
Wall Composition	22 cm concrete blocks *R* = 0.25 m²K/W Exp. Polystyrene λ = 0.034 W/mK	Aerated concrete blocks *R* = 2.27 m²K/W Exp. Polystyrene λ = 0.034 W/mK	Aerated concrete blocks *R* = 1.82 m²K/W Exp. Polystyrene λ = 0.034 W/mK				
Wall insulation thickness	10 cm	15 cm	20 cm	25 cm	30 cm		
Floor insulation thickness	10 cm	20 cm					
Ceiling insulation thickness	10 cm	20 cm	30 cm	40 cm	50 cm		

algorithm, real product databases will be used.

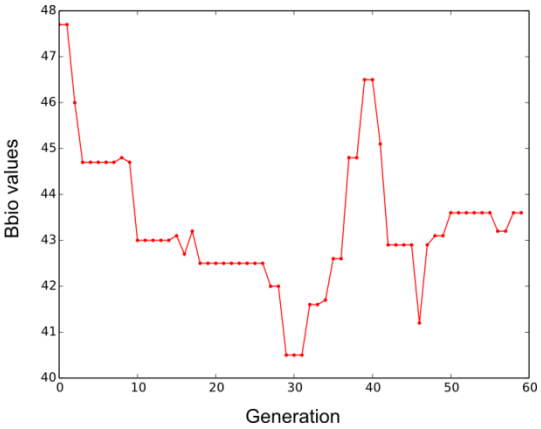

Figure 3 Minimum B_{bio} result per generation

Another important point to discuss from the simulation results is that genetic algorithms do not always find the optimal solution. The higher the numbers of generations and individuals are, the better the chances to reach the optimum solution. However, this is not necessarily the final purpose of such a tool. Instead, it should enable determining which parameters influence the most the objectives to help making design decisions. In order to do so several key post-processing programs have to be implemented. First, an interface should enable the user exploring the best solutions found by the algorithms and displaying the corresponding design variables values in a user-friendly way. This would help spotting the most influential parameters. Then, it should also enable modifying those and rerunning the objective functions in order to try to reach better results. This was done without such an interface for this study. The B_{bio} indicator was decreased by a further 2.7 % for the northern case by setting the wall insulation thickness to 30 cm instead of the 25 cm found by the algorithm. For the southern case an improvement of 2.8 % was achieved by setting a larger south facing window than the one selected by the algorithm (from 6 m² to 7 m²) and by selecting the most compact building design instead of the middle one. This clearly shows that the algorithm did not reach the minimum but that it was not far off.

The optimisation module can be improved in two ways: by decreasing computation time or improving its robustness, i.e. its capacity to always get close to the optimum. Decreasing the computation time can be achieved already simply by setting a lower number of generations and individuals per generation. In most cases, putting those values down to 40 and 10 would give similar results for a computational time of 30 minutes instead of 90. On figure 3 for instance, the minimum is reached at the 29th generation. Another way of reducing it would be to use multithreading computation. It is suited for this problem because the evaluations of individuals are independent from one another at each generation so they could all be ran on different computer processors. This should reduce the computing time by a factor proportional to the number of processors. Finally, several computers could be parallelised to decrease this time even further. Concerning robustness, the parameters of the genetic algorithm could be modified, like probabilities of selection and mutation for instance. The balance between diversification and intensification should be well managed though in order to give reasonable results in all cases. Implementing the user interface described previously is another way of improving the chances to get closer to the optimum. This can also be done automatically by adding a direct search algorithm to the optimisation module as described in Hamdy *et al.* paper (Hamdy et al., 2011). All these possibilities of improvement will be experimented at a later stage.

CONCLUSION

These first single objective optimisation results validated the overall methodology. Genetic algorithms seemed appropriate in terms of precision. The best individuals have fitness values less than 5 % higher than the true minimal fitness value. They also perform in a reasonable time. In this case using 40 generations and 10 individuals per generation seemed a good compromise between computation time, around thirty minutes on a desktop computer, and robustness. Several possible improvements were highlighted. These include multithreading to decrease computation time and developing a user interface to explore the solutions found and modify them directly. The need to include more objective functions and constraints was also pointed out. Costs and thermal comfort should be taken into account in the overall design process.

Even if the results seem promising, the use of the building regulation program as a simulation tool for energy performance assessment should be questioned. On the one hand using the simplifying hypotheses included in the program enables having standard energy performance indicators. On the other hand, these indicators may not precisely reflect reality. The B_{bio} index used in this study characterizes only the building envelope, but even the primary energy consumption C_{ep}, also calculated from the regulation program, is not as precise as the energy performance result that one could obtain from a dynamic building simulation tool. Some significant variations may be observed due to oversimplifying hypotheses included in the program. Comparisons with more precise building simulation tools will have to be carried out in order to assess the potential gaps between them. Finally, perspectives of the tool could be to include objective functions with broader environmental and energy considerations like embodied energy or other life cycle analyses indicators.

REFERENCES

ADEME, A. de l'Environnement et de la M. de l'Energie, 2010. Bilan du PREBAT : le bâtiment basse consommation, une obligation du Grenelle Environnement et déjà une réalité.

Attia, S., Hamdy, M., O'Brien, W., Carlucci, S., 2013. Assessing gaps and needs for integrating building performance optimization tools in net zero energy buildings design. Energy Build. 60, 110–124.

Brownlee, A.E.I., Wright, J.A., Mourshed, M., 2011. A multi-objective window optimisation problem.

Chantrelle, F.P., Lahmidi, H., Keilholz, W., Mankibi, M.E., Michel, P., 2011. Development of a multicriteria tool for optimizing the renovation of buildings. Appl. Energy 88, 1386–1394.

Deb, K., Pratap, A., Agarwal, S., Meyarivan, T., 2002. A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans. Evol. Comput. 6, 182–197.

Erhorn, H., Erhorn-Kluttig, H., 2013. Towards 2020 – Nearly Zero-Energy Buildings – Overview and Outcomes 2012, Concerted Action EPBD, in: Implementing the Energy Performance of Buildings Directive (EPBD). ADENE, pp. 47–56.

Evins, R., 2013. A review of computational optimisation methods applied to sustainable building design. Renew. Sustain. Energy Rev. 22, 230–245.

Fesanghary, M., Asadi, S., Geem, Z.W., 2012. Design of low-emission and energy-efficient residential buildings using a multi-objective optimization algorithm. Build. Environ. 49, 245–250.

Fortin, F.-A., De Rainville, F.-M., Gardner, M.-A., Parizeau, M., Gagné, C., 2012. DEAP: Evolutionary Algorithms Made Easy. Journal of Machine Learning Research 13, 2171–2175.

French Ministry of Ecology, Energy and Sustainable Development, 2010. Réglementation thermique « Grenelle Environnement 2012 ».

Hamdy, M., Hasan, A., Siren, K., 2011. Applying a multi-objective optimization approach for Design of low-emission cost-effective dwellings. Build. Environ. 46, 109–123.

Hamdy, M., Hasan, A., Siren, K., 2013. A multi-stage optimization method for cost-optimal and nearly-zero-energy building solutions in line with the EPBD-recast 2010. Energy Build. 56, 189–203.

Kolokotsa, D., Rovas, D., Kosmatopoulos, E., Kalaitzakis, K., 2011. A roadmap towards intelligent net zero- and positive-energy buildings. Sol. Energy 85, 3067–3084.

Leckner, M., Zmeureanu, R., 2011. Life cycle cost and energy analysis of a Net Zero Energy House with solar combisystem. Appl. Energy 88, 232–241.

Marszal, A.J., Heiselberg, P., 2011. Life cycle cost analysis of a multi-storey residential Net Zero Energy Building in Denmark. Energy 36, 5600–5609.

PROGEMI, 2014. PROGEMI [WWW Document]. URL http://www.progemi.fr/

Roger, M.-C., Remesy, R., Menager, Y., Le Guen, S., 2012. EPBD implementation in France - Statut at the end of 2012, in: Implementing the Energy Performance of Buildings Directive (EPBD). ADENE, pp. 181–190.

Teghem, J., 2012. Recherche Opérationnelle - Tome 1 : Méthodes d'optimisation. Ellipses Marketing.

Tuhus-Dubrow, D., Krarti, M., 2010. Genetic-algorithm based approach to optimize building envelope design for residential buildings. Build. Environ. 45, 1574–1581.

Verbeeck, G., Hens, H., 2007. Life Cycle Optimization of Extremely Low Energy Dwellings. J. Build. Phys. 31, 143–177.

Vergne, F., 2011. RT 2012 : un surcoût de construction de 15-20% [WWW Document]. lemoniteur.fr. URL http://www.lemoniteur.fr/190-metiers/article/actualite/775575-rt-2012-un-surcout-de-construction-de-15-20 (accessed 1.6.14).

Wang, W., Rivard, H., Zmeureanu, R., 2005. An object-oriented framework for simulation-based green building design optimization with genetic algorithms. Adv. Eng. Inform. 19, 5–23.

Wright, J.A., Zhang, Y., 2005. An "Ageing" Operator and Its Use in the Highly Constrained Topological Optimization of HVAC System Design. GECCO.