HYBRID DISCRET-CONTINUOUS MULTI-CRITERION OPTIMIZATION FOR BUILDING DESIGN

Abbass RAAD[1], Van-Binh Dinh[1], Jean-Louis Coulomb[1], Benoit Delinchant[1] and Frederic Wurtz[1]

[1] Univ. Grenoble Alps, CNRS, G2Elab, F-38000 Grenoble, France

ABSTRACT

This paper presents multi-physics modeling and optimization to design buildings by simultaneously treating the thermal comfort and the total cost (CAPES/OPEX). It focuses on real situations of the design of buildings in which it is required to use real components (Material Database) and to resolve the mixed continuous discrete optimization problem.

This optimization will be ensured, thanks to the optimization software of our lab (FGot[1] or the commercial version GOT-It[2]) using GMGA (Grid Multi-objective Genetic Algorithm). The coupling between optimization and the building model is integrated using a RESTful Web Service (HTTP protocol).

The aim of the optimization is to find the optimal settings for the design of the studied building, regarding insulation, windows, etc… while targeting energy performance, comfort and economic goals. It is thus a multi-criterion optimization that can be reached by a weighted mono-objective method or by finding optimal Pareto front.

So in this paper, we are dealing with global modelling for optimization purpose, model and optimization algorithm coupling, mixed discret-continuous optimization concerning database existing components.

INTRODUCTION

It is now common to use different models to simulate the behavior of systems. These systems become more and more complex because of the new environmental constraints, the new standards and regulations, the use of automation and the new technologies. All these make the choice of building design very complex.

A literature review on the topic of building design optimization, leads to the necessity of using multi-criteria approaches, such as those detailed in (Mela, 2012). Multi-criteria optimization is a first challenge. Moreover, many of the design parameters are discrete (e.g. window dimensions, wall thickness, insulation ...), whereas the physical model, depending from the same parameter is continuous. So it is interesting the compare the discrete and continuous optimization, and to see the respective advantages and difficulties of both approaches. Indeed, it known that discrete optimization can be very hard to solve, so it is our second challenge.

This paper first presents the building model (envelope and energy systems). Then, we develop the coupling with the optimization algorithm using our interoperability solution. Finally, we present the application of the multi-criterion optimization on a test case in which we are comparing continuous and discrete approach.

PROBLEM STATEMENT: MULTI-OBJECTIVE, MIXED DISCRET-CONTINUOUS OPTIMIZATION

First of all, we need a building model and an optimization algorithm dedicated to solve a multi-objective problem. Our objectives will be the thermal comfort and global cost (CAPEX/OPEX).

In this simulation, we consider the global cost of the life cycle which is composed of (1) the investment cost of the envelope, (2) the investment, maintenance, replacement costs of the heating system and the cooling system, (3) the cost of energy bought from the grid. Those design criterions should allow designers to have a good vision on components' cost in the early stage of the design process. The optimal solution of envelope parameters has to minimize the global cost and maximize thermal comfort (winter and summer).

Consequently, our approach will help to support the designer decision and avoid the oversizing. Regarding the envelope parameters, we will consider the internal and external insulation thickness of walls, roof, floor, solar heat gain coefficient (SHGC), window area and the inertia thickness as optimization parameters.

At first, we will work with continuous variables, and afterwards, discrete variables will be used in order to represent real components.

We use GMGA (Grid Multi-objective genetic algorithm) genetic algorithm (Delaforge T., 2015) to solve this multi-objective problem.

[1] http://forge-mage.g2elab.grenoble-inp.fr/project/got
[2] http://www.cedrat.com/en/software/got-it.html

In order to improve interoperability of modeling tools (written in Python) and optimization algorithms (written in Java) we have developed a web service based connector in order to solve this interoperability issue.

It is worth noting that our methodology can be integrated with any model written in any programming language. That is, users can benefit by optimizing their models using our approach.

METHODS AND SOFTWARE IMPLEMENTATION

Optimization tool

FGot (Featuring a Global Optimization Tool) is a platform developed at G2ELab used for model reduction, analysis and optimization. Below are its main features:

Model Reduction

- Screening (selection of the most influent parameters)
- Response surface (Polynomial and Radial basis function)

Analysis

- Evaluators (deterministic and stochastic, interval analysis)
- Plotters (Y(X), Z(X, Y), Isoval(X,Y), etc.)

Optimization

- Optimization problems (objectives, constraints, uncertainties on control or design parameters)
- Optimization algorithms such as SQP, GA, Niching, GMGA.
- Decision-making such as sensitivity/robustness of solutions, Pareto frontier.

FGot can be also used as an interface between an optimization algorithm and a web-service based model.

Grid Multi-Objective Genetic Algorithm

GMGA is a genetic algorithm to find the optimal Pareto-front of multi-objective problems (Delaforge T., 2015). For the continuous case, it discretizes design parameters to form a grid. For discrete parameters, the grid step is one and related to a data base where each number represents a component or material.
The first step defines the grid setting bounds for the parameters and their grid step. Then the first population is generated using the Latin Hypercube Sampling (Kent R. Davey 2008). A LHS selects points on a grid in order to be sure that all possible levels of parameters are tested one time (*Figure 1*). The points are uniformly distributed on the axes of the domain. It does not ensure the uniformity of the resulting solution on the domain.
For the next generation, elitism is used to select the parents. Every point on the grid close to the best solution is explored. Then selection, crossover and mutation are used. The Pareto-front is improved using vicinity mutation.
For the domination sorting, cells (i.e. the sizes of the grid step) are built for each parameter. Only one individual is authorized per cell. If the number of children after this sorting is higher than the required population, the cell size is increased. The best individual per cell is kept. Thus, the new population is built.

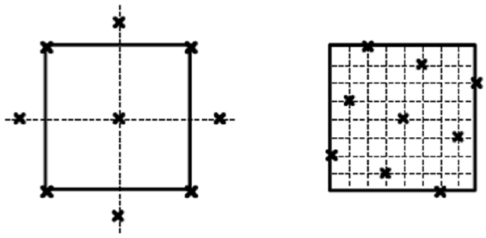


Figure 1: Grid selection, left standard sampling, right Hypercube Latin

Interoperability between model and optimization algorithm

For interoperability purpose, we have developed a generic connection between a building model and an optimization algorithm. For flexibility reasons, we have done it using web services. A web service is a unit of managed code that can be remotely invoked using Hypertext Transfer Protocol (HTTP). A web service allows the user to expose the functionality of existing code (such as a building model) over the network. Once it is exposed on the network, or locally on a same computer, another application can use the functionalities of the program. It allows various applications to communicate between each other and share data and services among themselves (RAAD A., 2015).
In our case, the building model (implemented in python) is the server, and the optimization software (FGot) is the client which sends requests to the server (*Figure 2*).

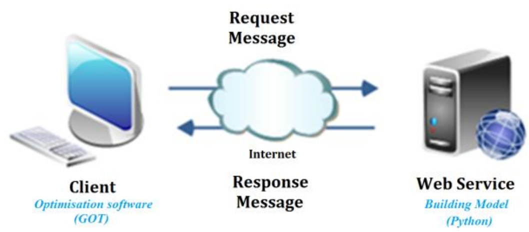


Figure 2 : Web-Server Exchange data

In the HTTP protocol, a method is a command that specifies a query type, that is to say, it asks the server to perform an action. In general the action concerns a

resource identified by the URL (uniform resource locators) that is followed by the method name, (e.g. http://localhost:8080/DoTest)
In our project we are using two types of methods:

- The GET method requests a representation of the specified resource. These requests should only retrieve data and should have no other effect.
- The PUT method requests storing the enclosed entity under the supplied URI (uniform resource identifier). If the URI refers to an already existing resource, it is modified. Otherwise, the server can create the resource with that URI.

Table 1 shows the main request actions to assure the communication between the model and the optimizer.

Table 1
API Client / Server

Requests	Methods	Parameters	Return
/getSessionId	GET	-	Session Id
/getRealScalar	GET	Id, Variable name	Variable value
/getRealVector	GET	Id, Variable name	Variable values
/setRealScalar	PUT	Id, Variable name & value	-
/setRealVector	PUT	Id, Variable name & values	-

Note that the building model simulation is done automatically when the client (optimization software) asks for outputs values.

APPLICATION

Building envelop model

The envelope model used in our test case is illustrated in the form of an electrical circuit with RC lumped parameters (Figure 3) (Dinh et al., 2015).

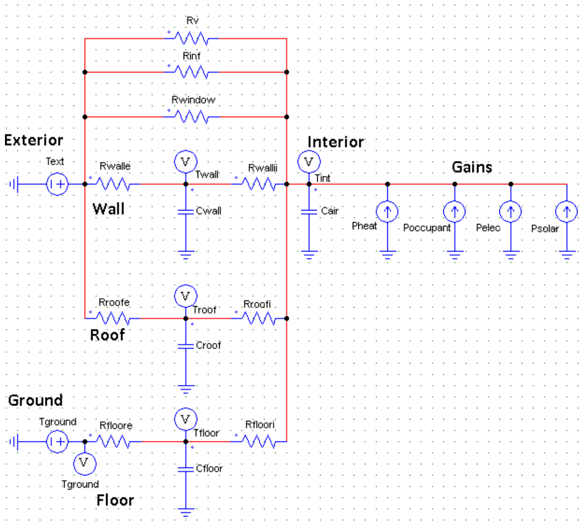


Figure 3 : Electrical equivalent circuit

The electrical equivalent circuit considers the resistances and capacitances as the insulations and inertias respectively of the building. These resistances and capacitances depend on the size and physical properties of the materials and can be analytically expressed by equations:

$$R = \frac{e_R}{\lambda.S} \qquad C = \rho.e_C.S.C_P \tag{1}$$

Where R, C are the thermal resistance (°K/W) and the thermal capacitance (J/°K) respectively; e_R and e_C are the insulation and inertia thickness (m); λ is the thermal conductivity (W/(m.°K)); S is the wall surface (m^2); ρ is the mass density (kg/m^3); C_P is the specific heat (J/(kg.K)).

Global cost model

In this study, the global cost represent the sum of the envelope investment cost, the cost of the thermal system (heating and cooling) and the present value of buying electricity from the grid.

$$COST_{tot} = C_{envelope} + C_{thermal} + C_{buy\_grid} \tag{2}$$

The envelope cost consists of the investment cost of the insulation, the inertia, and the windows.

For the thermal system, its cost function depends on the initial capital cost, the present value of the replacement cost, the present value of the maintenance cost. Therefore, the cost of thermal system Cost (€) is expressed as follows:

$$C_{thermal} = C_{inv\_thermal} + C_{rep\_thermal} + C_{M\_thermal} \tag{3}$$

Discomfort model

The thermal discomfort *discomf* expresses the sum of the winter thermal discomfort $discomf_{winter}$ (°C) and the summer thermal discomfort $discomf_{summer}$ (°C) which are defined as follows:

$$discomf_{winter} = \frac{1}{T_W}\sum_{t=1}^{T_W} e_T(t) \quad for\ e_T(t) > 0 \qquad (4)$$

$$discomf_{Summer} = \frac{1}{T_S}\sum_{t=1}^{T_S} e_T(t) \quad for\ e_T(t) < 0$$

With $e_T(t)$ is the difference between the interior temperature and the set point temperature at hour t:

$$e_T(t) = (T_{set}(t) - T_{\text{int}}(t)) \qquad (5)$$

T_S and T_W are the computing period in winter and summer respectively. The thermal discomfort is increasing when the building inside temperature is smaller, respectively greater, than the set point value in winter, respectively in summer.

Optimization criteria

In reality many buildings are built for more comfort in summer, the others are designed for more comfort in winter, all depending on the climate of the area in which they are localized.
Our study takes into account the comfort during winter and summer using their sum to get first criteria to minimize. The second objective is to minimize the global cost over the life cycle which is the addition of the capital expenditure (CAPEX) and the operating expenditure (OPEX).
So, in our optimization problem, we can differentiate four main objective functions to minimize: the CAPEX, the OPEX, the discomfort in winter and the discomfort in summer. In order to do this, a first bi-objective optimization is done in order to get the Pareto front trade-off between cost and discomfort.

Constraints are also taken into account, such as the total window area which must be greater than 1/6 of the living surface of building.

RESULT

Continuous Optimization

GMGA configuration is the following: 200 generations and 130 populations

Two objective functions: $COST_{tot}$ and *discomf*

At first, we choose two objective functions to minimize, the global cost ($COST_{tot}$) and the thermal discomfort (*discomf* - including both winter and summer).
Figure 4 shows the Pareto front of 103 solutions between the two objectives.

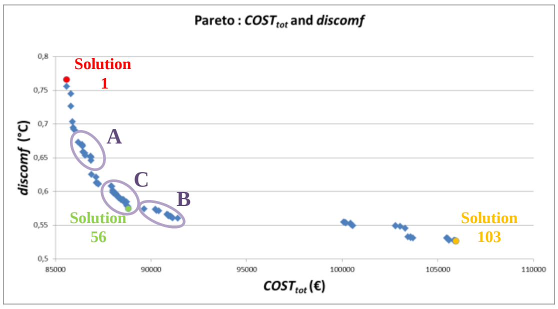


Figure 4 : Pareto $COST_{tot}$ and discomf - Continuous Optimization

In Figure 4:
Solution 1 (top left) is a building design with a minimum cost (approximately € 85 500), but it does not guarantee a good comfort. Indeed the sum of the discomforts in winter and summer, as defined in equation (4) is about 0.76 °C.
Solution 103 (bottom right) is a building design with the worst discomfort (0.52 °C), but this solution is too expensive comparing to others (approximately € 106 000).
Solution 56 (on the middle of the Pareto front) offers a good tradeoff between discomfort (0.56 °C) and overall cost (€ 88 800).
The Pareto front helps the designer making choice depending on it needs and financial capacities.
In addition, the designer can refine his choices by looking at the Pareto discomfort / Global Cost of each season separately, as shown in Figure 5 and Figure 6. For example, it can be noticed that even if solution 103 is very good for the global discomfort (as well as other with high cost), it is worst for summer discomfort that solutions near to solution 56.

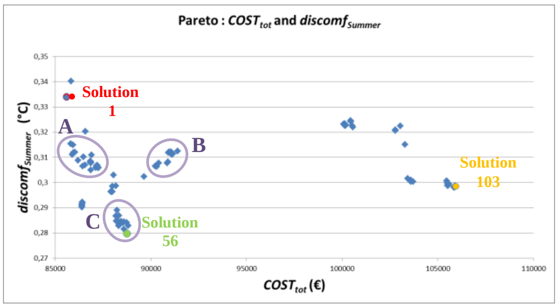


Figure 5: Summer discomfort depending on the global cost for the previous optimization results.

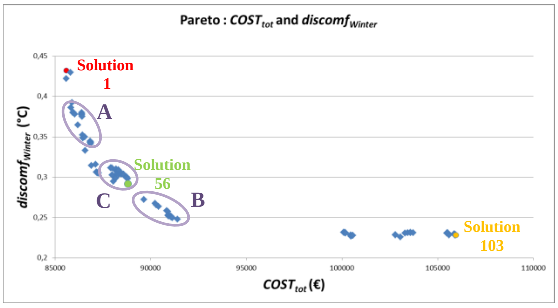


Figure 6: Winter discomfort depending on the global cost for the previous optimization results

In these pictures, it is possible to distinguish 3 groups of solutions: A, B and C.

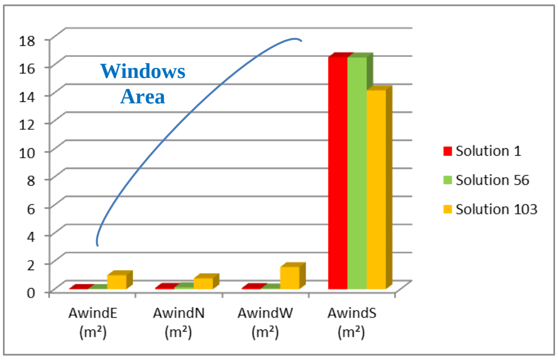
Without these pictures, group B was supposed to be good solution candidates. But if we look specifically for each discomfort (summer or winter), new information are appearing. For example, for group B in Figure 5, it is clear that summer comfort is not guarantee. In fact, its low winter discomfort compensates his high summer discomfort in the aggregated comfort objective.

Basically to have a better choice of solution requires a thorough analysis for each objective.

Figure 7 shows the design parameters values for these three solutions (1, 56 and 103).

This design provides a concrete vision on the choice of components. For example, the solution 103 (Figure 4) that offers the best overall comfort (summer and winter) is the highest cost. This cost is initially cause by the large thicknesses of the inertia and of the insulation (Figure 7).

These values are accepted physically, but the corresponding manufactured product may not exists, that is why a discrete approached which links the optimization algorithm to a database will be detailed in the next section.



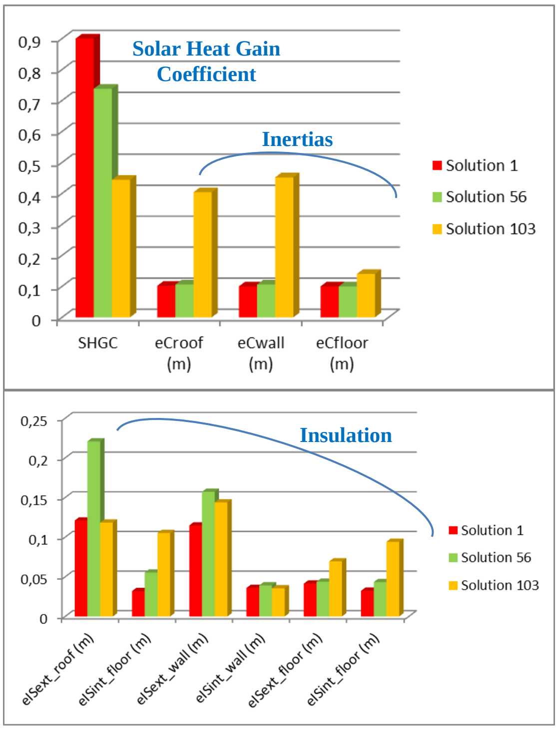


Figure 7: Parameters values for solutions 1, 28 and 103

<u>Three objective functions:</u> $COST_{tot}$, $discomf_{winter}$ and $discomf_{summer}$

The same problem can be treated in another approach, by considering three objective functions to minimize: the global cost ($COST_{tot}$), the thermal discomfort in summer ($discomf_{Summer}$) and the thermal discomfort in winter ($discomf_{Winter}$).

Figure 8 presents the 3D Pareto for the optimization with three objective functions. The optimization gives 26 solutions. Note that when we increase the number of objectives for the same problem and same physical configuration, the optimizer gives less solutions. To enrich the set of solutions, it is necessary to increase the population and generation numbers.

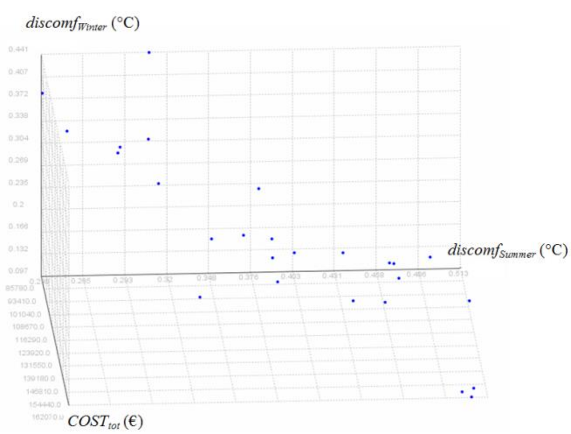


Figure 8: 3D Pareto $COST_{tot}$, $discomf_{Summer}$ and $discomf_{Winter}$ - Continuous Optimization

Having three different objective functions, is a very interesting analysis tool with can be analysed with 3D interactivity. Indeed, in this case (three objective functions) the optimizer tries to find a range of solutions that gives the expectations of each comfort function separately. This range of solutions will be more constrained if we have the comfort (summer and winter) as one aggregated objective function. Then, many interesting solutions founded in the case of three objective functions do not appear in the case of two objective functions.
This can also be done to optimize and analyze separately the cost objective functions CAPEX and OPEX. It allows the manufacturer to fix the choice of building components in a more specific study in term of capital or operating expenditures. The more the objectives increase, the more difficult the analysis is (3 or 4 dimensions). It also increases the computation time (twice as slow for the same number of generation and population).

Discrete Optimization

Now, we would like to adapt our optimization strategy using existing systems and materials. In order to do that, a discrete optimization is performed. What was expected is that continuous optimization will give best solutions than discrete optimization but with non-existing materials. On the other side, the discrete optimization will give real expected performances to the designer while choosing existing materials.
The same optimization (Building model, two objective functions, GMGA, 200 generations and 130 populations) with discrete parameters gives 25 optimal solutions in a Pareto front (Figure 9).
Note that the combinatorial of possibilities for 14 discrete parameters with their related discrete values leads to about 270 million combinations.
Figure 9 shows a comparison between discrete and continuous solutions.
The solutions proposed by the continuous optimization are more interesting at the physical level of the problem. In fact, these solutions offer a better compromise between comfort and overall cost. This optimization has found the Pareto front of best physically feasible solutions. This is very interesting to highlight what are the best components with a global overview of the building.

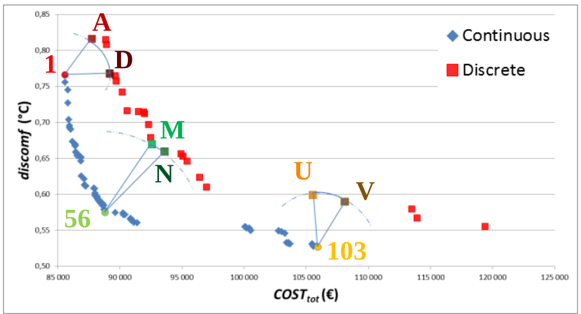
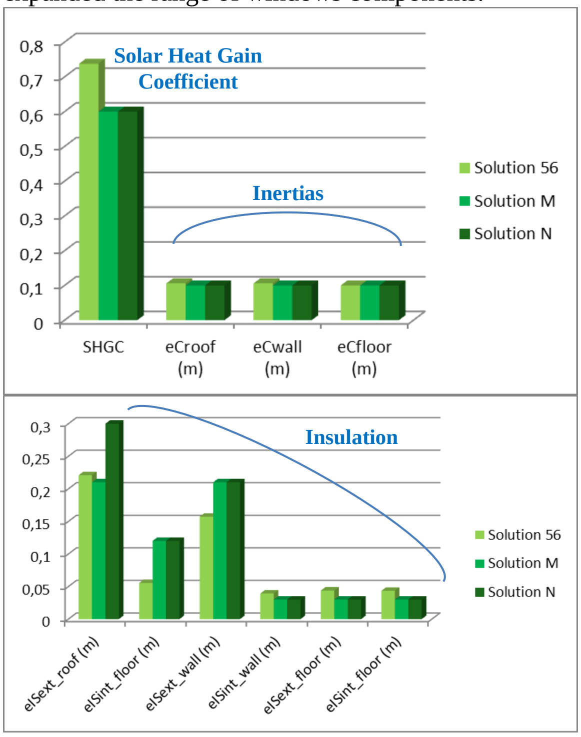


Figure 9: Pareto of continuous discrete optimization

For each continuous solution analyzed in the previous section (solution 1, 56 and 103) we searched for the nearest discrete solution. Discrete solutions (A and D), (M and N) and (U and V) are the closest solutions of continuous solutions 1, 56 and 103 respectively.
To deeply analyze the changes from continuous to discrete, we are comparing the construction parameters values for these set of solutions. As an example we compare the parameters value of the continuous solution 56 with that of discrete solutions M and N, (Figure 10).
The values for inertia and insulation are very closed.
The high value for the insulation thickness *eISext_roof* of the solution N may explain its better comfort with respect to the solution M, and its higher cost (Figure 9).
Regarding the window areas (Figure 10), there is a large gap between discrete and continuous optimal solutions. In fact the continuous solution 56 provides a concentrated window area on the south side, while the discrete solutions (M and N) offer a distributed area on the four sides but precisely on the east and south sides. This difference is probably due to what we are calling a “discrete system effect” which means that the best discrete solution is not the nearest of the continuous one for each parameters. This effect depends on the component library size. Indeed, if library size tends to the infinite (when the discretization step size tends to zero), then both discrete and continuous formulations are equals.
So, to increase the number of discrete solutions and to be near from the continuous solutions, we have expanded the range of windows components.



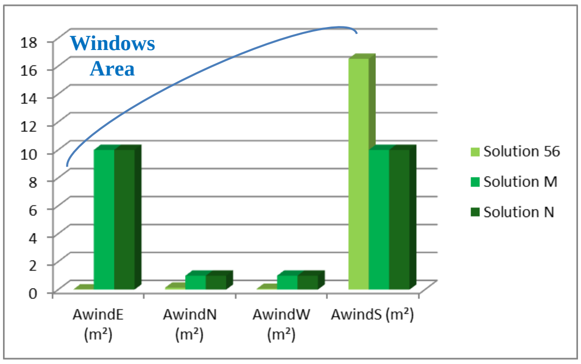


Figure 10 : Parameters values for continuous solution 1 and discrete solutions A and D

Note that the number of possible combination for this discrete optimization has become of the order of 4 billion combinations.
Figure 11 shows the new discrete solutions proposed by the optimizer by increasing the range of components available, especially for windows.

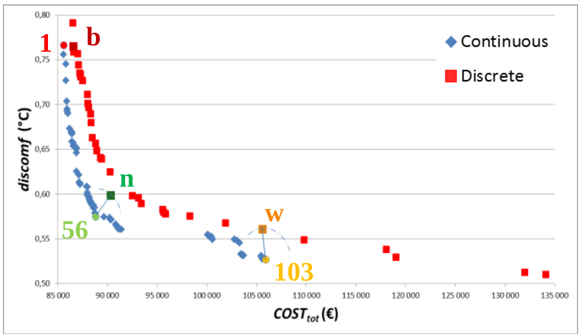


Figure 11: Pareto of continuous & discrete optimization – wide variety of component

With the new range of components, the solution "n" is the discrete solution closest to the continuous 56 solution which proposes the values of the window surfaces shown in Figure 12.
These surfaces are distributed on the four sides North, South, East and West, with a stronger weight in South side.

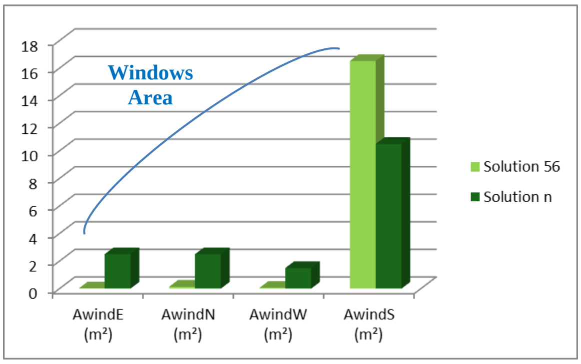


Figure 12 : Windows area values for continuous solution 56 and discrete solutions n

We find that for some other parameters, such as insulation or thickness, the optimal discrete values correspond to the nearest value of the continuous solutions. For other parameters (such as window area) the "discrete system effect" is still appearing. So it is possible to conclude that the optimal discrete solution is not necessarily the discrete solution closest to that obtained continuously.
As a result, it is important to realize the discrete optimization with real components database. But discrete optimization is still challenging. Indeed, the size of the list of components (discrete optimization) and the number of population selected affect proportionally the number of solutions and the computing time of the optimizer. In addition, whenever we increase the size of the optimization problem, we must increase the number of generation for a better convergence. Moreover, adding computation time issue and a small number of components available in the database, adding difficulties to define a good design problem formulation, it is frequent to have convergence issues and constraints that are not satisfied.
For these reasons, it is recommended first to solve the optimization problem by using continuous variables (if possible) with performing algorithms such as gradient based algorithm (Dinh V.B 2015). Once the optimization problem is well posed and constraints are well defined for "imaginary solutions" (Wurtz, F., 2012), discrete optimization can be tested.

CONCLUSION

In this article, a multi-objective optimization problem has been defined to find building parameters in order to minimize discomfort and global costs. Web service specifications as been defined in order to provide the communication and the data exchange between building models and optimization softwares. It has been implemented and used for a building model developed in python and a multi objectives genetic algorithm (GMGA) available in FGot software coded in Java Language.
Multi-objective optimization results have been analyzed regarding the number of objectives, mixing weighted sum of objectives and Pareto front.
It has been showed that discrete optimization is crucial to deal with real manufactured components and to avoid what we have called "discrete system effect". Nevertheless, a first continuous optimization, using gradient based algorithm is still an important task in order to help at defining the optimization problem.

NAMENCLATURE

HTTP,	Hypertext Transfer Protocol;
URL,	Uniform Resource Locators;
URI,	Uniform Resource Identifier;
API,	Application Programming Interface;
R,	Thermal Resistance;
C,	Thermal Capacitance;
C_P	Specific Heat;

ρ,	Mass Density;
λ,	Thermal Conductivity;
e_R,	Insulation;
e_C,	Inertia Thickness;
S,	Wall Surface;
$e_T(t)$,	Difference between set point and interior Temperature;
$T_{set}(t)$,	Set Point Temperature;
$T_{int}(t)$,	Interior Temperature;
T_W,	Computing Period in Winter;
T_S,	Computing Period in Summer;
$COST_{tot}$,	Global Cost;
$CAPEX$,	CApital EXPenditures;
$OPEX$,	OPerating EXpenses;
C_{inert_wall},	Envelope Investment Cost;
$C_{thermal}$,	Cost of thermal system;
C_{buy_grid},	Cost of buying electricity;
$C_{inv\_thermal}$,	Thermal Cost;
$C_{rep\_thermal}$,	Replacement Thermal Cost;
$C_{M\_thermal}$,	Maintenance Thermal Cost;
$discomf$	Thermal discomfort;
$discomf_{Winter}$	Thermal discomfort in winter;
$discomf_{Summer}$	Thermal discomfort in summer;
$eISext\_wall$,	External insulation thickness of wall ;
$eISint\_wall$,	*Internal insulation thickness of wall;*
$eISext\_roof$,	External insulation thickness of roof;
$eISint\_roof$,	Internal insulation thickness of roof;
$eISext\_floor$,	External insulation thickness of floor;
$eISint\_floor$,	Internal insulation thickness of floor;
$eCwall$,	Inertia thickness of wall;
$eCroof$,	Inertia thickness of roof;
$eCfloor$,	Inertia thickness of floor;
$AwindS$,	Window area in South;
$AwindN$,	Window area in North;
$AwindE$,	Window area in East;
$AwindW$,	Window area in West;
$SHGC$,	Solar Heat Gain Coefficient;

ACKNOWLEDGEMENT

French ANR project COSIMPHI (CO-Simulation Multi-Physique Interactive - ANR-13-VBDU-0002)

REFERENCES

Dinh V.B., Delinchant B., Wurtz F., “On the sizing of building enveloppe and energy system integrating management strategy in sketch phase”, IBPSA 2015.

Raad, A., Reinbold, V., Delinchant, B., Wurtz, F., “Energy building co-simulation based on the wrm algorithm for efficient simulation over fmu components of web service”, IBPSA 2015.

Nguyen-Xuan H., Coulomb J.L., Gerbaud L., Crébier J.C. and Rouger N., “Application of Progressive Quadratic Response Surface Methode for an Oscillation Problem Optimization”, The international journal for computation and mathematics in electrical and electronic engineering (COMPEL), Vol. 31 Iss: 3, pp.842 – 857

Delaforge T., SCHANEN J., “Optimal sizing of passive components in power converters using discrete methods”, 2015.

Wurtz, F., Kuo-Peng, P., De Carvalho, E. S., “The Concept of Imaginary Machines for Design and Setting of Optimization Problems: Application to a Synchronous Generator”, ICEM 2012.

Kent R. Davey, “Latin Hypercube Sampling and Pattern Search in Magnetic Field Optimization Problems”. IEEE Transactions on Magnetics, vol. 44, issue 6, pp 974-977, June 2008

Chardon, S., et al., “Développement d’un outil pratique d’optimisation utilisant des algorithmes génétiques pour l’aide à la conception économique et énergétique des maisons individuelles”, IBPSA France. 2014: Arras.

Chantrelle, F.P., et al., “Development of a multicriteria tool for optimizing the renovation of buildings”. Applied Energy, 2011. 88(4): p. 1386-1394.

Taillandier, F., A. Ndiaye, and C. Fernandez, “Optimisation par essaim particulaire de la planification des actions de meintenance d'un parc immobilier”, in 31ème Rencontres de l'AUGC. 2013: E.N.S. Cachan.

Bornatico, R., et al., “Optimal sizing of a solar thermal building installation using particle swarm optimization”. Energy, 2012. 41(1): p. 31-37.

Kirkpatrick, S., C.D. Gelatt, and M.P. Vecchi, “Optimization by Simulated Annealing”. Science, 1983. 220(4598): p. 671-680.

Goldberg, D.E., “Genetic Algorithms in Search, Optimization, and Machine Learning”. 1989: Addison-Wesley.

Mela K, Tiainen T, Heinisuo M. Comparative study of multiple criteria decision making methods for building design. Adv Eng Inf