Using self-adaptive optimisation methods to perform sequential optimisation for low-energy building design

A. P. Ramallo González [a, 1], D. A. Coley [b]

[a] *College of Enginnering, Mathematics and Physical Sciences, University of Exeter, UK*

[b] *Department of Architecture and Civil Engineering, University of Bath, UK*

Abstract

The use of software tools to aid building design, or to show compliance, is now commonplace. This has motivated investigations into the potential of optimisation algorithms, used in such software, to automatically optimise designs, or to generate a variety of near-optimal designs. Optimisation always requires the evaluation of a large number of possibilities, before a final selection is made. When using a building simulator to assess the quality of designs, all possible solutions in the early stages of optimisation (when there is a high volume of choices) are evaluated using the same tool, so that the computational time for the assessment of each of the possibilities is the same as the time required for the final, refined choice of solutions. This paper suggests using a method of evaluation which changes as the algorithm evolves: whereas accuracy is initially compromised to improve the speed of the algorithm, the process is subsequently altered to produce a more accurate, evaluation process. This is a case of dynamic optimisation that requires an algorithm able to cope with changes in the objective landscape. A self-adaptive evolutionary strategy has been chosen, for its ability to "learn" about changes, and the influence of the different decision variables, in the objective function as they arise. The results show that this method can reach the same optimal design, with substantially lower computational time.

Keywords: Optimisation, ES, dynamic optimisation, sequential optimisation, energy, building

1 Introduction

The new measures to reduce carbon emissions in the building sector have increased the interest on producing low-energy designs, being these, buildings that need a fraction of the energy needed for a traditional building to create the same levels of comfort. Several software packages have been developed to aid building professionals in the design of low-energy buildings. These software packages (getting more and more complex with time) are able to evaluate the building physics and its energy systems in a very comprehensive manner [1]. Among the phenomena that can be modelled with this kind of software one can found radiation, phase changing, humidity transfer, pollutant emissions, air movement, and many others. Although computers have become more powerful with time, this improvement does not overcome the growth on complexity of the building simulators, and the time needed to run a building annual simulation in a personal workstation is still substantial.

Those simulation times can be of the order of hours, making the process of investigating several designs slow and tedious.

The fact that current dynamic simulators can assess the quality of buildings in term of energy efficiency, has motivated some building scientists to use optimisation algorithms coupled with these software tools to find low-energy designs [2-4], but the long computational times needed to run these optimisations can make the process unfeasible.

Other authors had used simplified building models to be able to run the optimisation in relatively short times [5, 6]. Although the results of these research works are enlightening, one could argue that due to the use of a basic simulator, only approximated optimisation for the early stage can be performed, and a more complex simulator should be used for refining the design.

Another options, is to create Response Surface approximation models of the real objective function, and optimise that model (such as the work of [7]. This method allows using complex simulators; however, Magnier and Hagnier pointed out that the ANN may produce results with errors up to 3.9 %, therefore implying a drawback in the method. Apart from that, ANNs suffer of the *curse of dimensions* (explained bellow) as the number of points needed to train the ANNs grows exponentially with the number of decision variables of the optimisation problem.

The two cases above are examples of two ways of tackling problems that present unviable computational times: One, using a simple dynamic model to reduce the time of evaluating the objective function; or two, developing a surrogate model that will mimic the objective function and can be evaluated with short computational times.

In the one hand, the idea of creating a surrogate model for the whole decision space looks not ideal; in the other hand, using a basic building simulator may not provide with the accuracy needed for a given problem. In this thesis, a different method is suggested.

The methodology that is presented here uses an evolutionary algorithm as a core of the optimisation; as the algorithm evolves, the solutions are assessed with different assessment tools that require different computational times.

This way of evaluating the objective function, can make some decision variables to have no effect on the objective function when the simple assessment tools may not consider those decision variables (e.g. thermal mass of partitions in a steady-state calculation methodology). For this reason, the algorithm needs to be able to “catch up” on those decision variables when the assessment method becomes complex enough to interpret them. The algorithm should also maintain the values of the decision variables that were optimised in previous “cheaper” stages of the optimisation if the values are correct for further assessment tools.

The application of the methodology to a building design problem follows in Section 5, and the results are presented and described in 6, followed by conclusions and references in Section 7 and 8.

2 Previous work

The use of complex assessment tools for the evaluation of potential solutions can render optimisation unfeasible. There is a standard procedure in engineering to tackled this problem developed by Barthelemy and Haftka [8]:

1. Create a surrogate model of the objective function by:
    a. implementing a simpler model to assess the solutions, based on the physics of the problem
    b. creating an approximation of the objective function after evaluating a number of points within the objective function (meta-modelling)
2. Optimise the surrogate model
3. Verify the optimality of the solution of the surrogate model with the objective function

The creation of a surrogate model based on physical principles is normally challenging. A quantitative change has to be done in the way that the system is modelled to obtain a model that requires less computation. This can not be done in many cases due to the complexity of the system to be analysed or the nature of the problem (for example, search of natural modes of vibration).

Several works can be found in the literature where meta-modelling is used to create surrogate models for the optimisation, examples of these are [9-15] in mechanical engineering, and [7] in building design.

The work of Jin et al. summarised the strengths and weaknesses of four of the most popular meta-modelling techniques, namely polynomial regression, multivariate adaptive regression splines, radial basis functions and kriging [16].

One of the weaknesses of meta-models is that they suffer from the *curse of dimensions* [15]. This effect can be explained as follows: the number of points that are needed to create a realistic surrogate model of the decision space grows exponentially with the number of dimension of the objective function. As an example, if one wants to have 3 points per dimension in a decision space with 20 decision variables, one would need $3^{20} = 3486784401$ points, if the problem had 3 decision variables, one would need $3^3 = 27$ points. To create the surrogate model of the decision space the points need to be evaluated with the real objective function, and eventually be used to generate the surrogate model, having a large number of decision variables has therefore a clear impact on the computational time needed to create the meta-models: the computational time to create a meta-model grows exponentially with the number of decision variables.

Creating surrogate models was considered by Booker et al [9] as not ideal. In their report published by NASA, Booker et al. argues this violates a fundamental tenet of numerical optimisation: "*one should not work too hard until one nears the solution"*.

This was related to the need of constructing a surrogate model before knowing the shape of the decision space and performing an optimisation run of the surrogate model that might have been built with points that are not near the areas where the optimum is located and therefore does not represent the areas of high fitness well. Booker et al. suggested a more efficient way of performing the optimisation: he used an on-line surrogate model that improves during the optimisation as more "true" points are selected and calculated in areas with high fitness.

The work of Magnier and Haghighat [7], was the only one found in the literature that uses surrogate models to make a more efficient optimisation in building design. In the work of Magnier and Haghighat artificial neural networks and true points in the decision space are used to create a meta-model that is optimised.

The use of surrogate models has almost not been exploited in building design although Wetter and Wright mentioned in their work [17] that the use of optimisation with surrogate models in building design is a promising area of research. No research has been found that uses optimisation for building design and the surrogate models created by simplifying the models themselves (type a in the listing of Barthelemy and Haftka).

3 Method

In this paper, we show a way of combining modern optimisation techniques with different building modeling techniques to create an optimisation method for building design that is more efficient than the traditional methods.

The optimisation uses different assessment tools during the optimisation. This means that the objective landscape will change during the optimisation. That is called a Dynamic Optimisation Problem (DOP). This kind of problems will be described in Section 3.1.

The algorithm used has to be such that will be able to adapt to these changes in the objective landscape. We have selected the Covariance Matrix Adaptation Evolutionary Strategy (CMA-ES) for this. The algorithm is described in Section 3.2.

Three ways of evaluating the energy demand of the building had been chosen for this work. The specific tools used here are not an intrinsic part of the method. We described the three chosen in Section 3.3.

The implementation of the methodology to perform the application of Section 4 is shown in Section 3.4.

3.1 Dynamic optimisation problems

The objective function in some optimisation problems is not constant in time, but instead, it changes during the optimisation. The algorithm must track the optimum as it moves through the objective landscape to make sure the best option is always chosen. Such problems are called Dynamic Optimisation Problems (DOPs), and solving them has gained considerable popularity in the last few decades [18]. Examples are complex control in robotics or network management.

The formulation of a DOP is:

$$\text{optimise } f(\boldsymbol{x},t), \quad \text{subjected to } \boldsymbol{x} \in \Omega. \tag{Eq. 1}$$

where f is the objective function, $\boldsymbol{x}$ is the set of decision variables, t is the time at which the objective function is evaluated and Ω is the set of viable options (the decision space). If one compares the formulation of Eq. 8.1 to that of a traditional optimisation problem, the time parameter is the only difference. In DOPs, the objective function (or objective landscape for a more graphical understanding) changes as the optimisation is performed. This generates a landscape with one (or more) moving optimum.

The review by Cruz et al. [18] outlines the features that an optimisation algorithm should have in order to be efficient in tracking a movable optimum in a DOP. In most DOPs, the objective landscape does not change continuously in time. Instead, the optimisation algorithm is capable of performing i iterations with a static objective landscape before the next change occurs. During the i iterations, the algorithm has to look for the location of the peak. However, it is not intended that the algorithm will converge completely in that single point, as the algorithm should be able to move to other areas of the landscape efficiently if the objective landscape changes.

Population-based algorithms have proven to be a good way of maximising exploration of the decision space, and thus being able to recognize changes in the objective space [19]. Algorithms that use a set of solutions (populations), instead of a single one, are able to evaluate several points of the decision space in each iteration. If the spread of these points is maintained, the algorithm will always have some notion of the shape of the objective landscape. This spread of solutions is called diversity (following the evolutionary jargon).

Branke detailed the features that make a evolutionary algorithm suitable for solving DOPs [20]. These are:

1. Increasing diversity after a change,
2. Maintaining diversity throughout the run,
3. Use of memory, and
4. Multiple populations.

The review by Cruz et al. [18] pointed out that Evolutionary Algorithms (EAs) are one of the most popular and efficient ways of solving DOPs. EAs are population-based algorithms that have been used to solve optimisation problems for almost four decades now [21]. They have proven effective in several synthetic objective functions and real-world problems, and have produced satisfactory results (see [22], [23] , [24], [25] for general texts on EAs). These algorithms use crossover, mutation and selection to find the best individuals (solutions) when a population (set of solutions) evolves (converge) over generations (iterations).

The review by Cruz et al. and the one by Branke [20], describe how increasing diversity, following a change in the environment, or maintaining that diversity are ways of adapting a traditional EA to make it capable of solving a DOP. In this work, a specific EA has been selected that is able to enhance diversity intrinsically when needed.

3.2 The algorithm: CMA-ES

The Covariance Matrix Adaptation Evolutionary Strategy (CMA-ES) is an EA created by Hansen et al. [26] with the aim of reducing the stochastic nature of ESs by using the Covariance Matrix (CM) in each population of solutions. The CM provides information about the shape of the decision space to the algorithm.

The CMA-ES has been proven to be an efficient method for optimisation in building design [27], its main strength is that the algorithm is able to recognise changes in the objective landscape through the CM, and adapt to those new scenarios. Although the calculation of the CM could be considered as a computationally expensive calculation, if the time needed to calculate this matrix is compares with the time needed to perform an annual simulation of a building using a complex simulator (EnergyPlus in our case), this calculation results trivial (three order of magnitude smaller), especially using small population sizes (10 individuals in this work).

The algorithm has been used in this research in the form of the code provided in [28].

3.3 Building assessment tools

Three ways of evaluating the objective function were chosen in this work:

1. Simple calculation methodologies: sets of algebraic equations or chart-based tools that allow the calculation of energy demands for a given building.

2. RC-networks based simulators: simulators which use an RC-network to represent the thermo-dynamic response of the building.

3. Benchmark simulators: tools more commonly used by architects and building scientists, benchmark simulators are comprehensive simulator suites that consider a multitude of phenomena that occur in the building. Examples of these are IES - <VE> or EnergyPlus.

This separation into three groups is mainly based in complexity of each method. However, there is an assumption here that computational time is proportional to final accuracy, but this is not proven[1]. In Figure 1, the different kinds of calculations are compared qualitatively, against computational time and potential error.

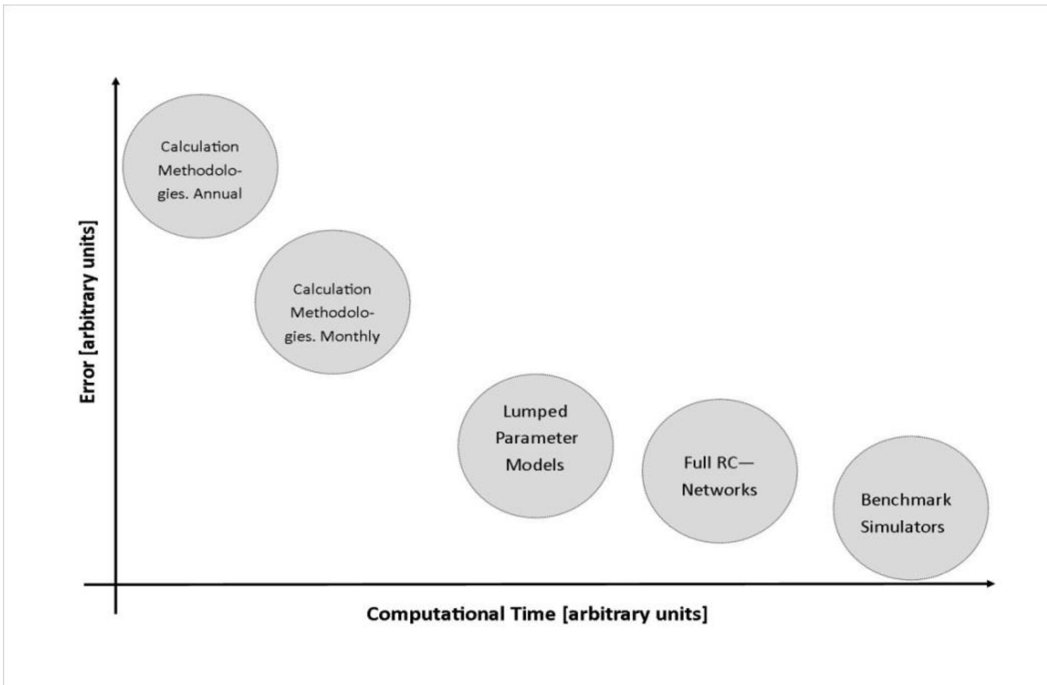

Figure 1 - Building assessment tools: Potential error vs computational time.

In this paper we suggest a procedure whereby multiple building assessment tools are used at different stages of the optimisation. Starting with those in the top-left of Figure 1, the process will follow the path of the graph, and it will end using a benchmark simulator (bottom-right of Figure 1). This is expected to be a more efficient way of using computational resources than maintaining the same levels of accuracy along the run as done in traditional optimisation. The algorithm has to be such that is able to update the search when a more complex simulator is used, so the variables that were left behind in the previous iterations get evaluated and optimised when it is needed.

[1] The higher complexity of a building model is due to a larger number of phenomena being represented by the code. However, simulating a larger number of phenomena does not imply having a better accuracy. The more phenomena represented the larger the number of inputs that are asked to the modeller. Even if those inputs are chosen under a fair judgment, they will be uncertainty. Examples of this is a weather file for a location that might come from observed data, but it is unlikely to happened again, or the consideration of solar gains, but considering an ideal horizon/sky. In opposition to this detailed simulators, simpler calculations are normally used with values that come from empirical correlations what can make the overall calculation more accurate than in the case than a complex simulation with the wrong inputs.

For this research, three stages using three different assessment tools are used. Although the selection of the number of stages is rather trivial, it is sufficient to use the main three building assessment tools represented in the previous classification.

One tool was selected for each of the stages listed previously. These tools are: the LT-method as the simple calculation methodology; an LPM simulator as the RC-Network simulator; and EnergyPlus for the benchmark simulator. Although there exist some weaknesses in some of these assessment methods, these tools were chosen to check the validity of the methodology, the reason for this will be explained in Section 4, where the application of the method is shown.

3.4 Implementation

The methodology developed in this research uses building assessment tools with different computational times at different stages of the optimisation for reducing computational times. The simpler calculation methods are considered surrogate models, and the benchmark simulator the closest computational assessment tool used to the real objective function.

Changing the assessment tool during the optimisation run will imply a change in the objective function, therefore will make the problem a dynamic optimization problem (DOP).

The method developed in this paper is a cross-over between optimisation using surrogate models and optimisation in dynamic environments. In this case, the change in the landscape is triggered by the algorithm itself once it has been seen that the run is reaching a termination criteria, with the changes being the selection of the next assessment method.

To implement the sequential optimisation algorithm, which has been called here CMA-ES with Sequential Assessment (CMA-ES-SA), the CMA-ES has been used with the default parameters suggested by the author in [28]. The algorithm is started with the simplest assessment tool, and it is run until the termination criteria are reached:

1. The scope of the search has become smaller than a given value; therefore the assessment method has to change to maintain diversity.
2. The algorithm gets stagnated: More than *N* generations without reaching Condition 1 (the algorithm is not converging)

When the algorithm reaches one of the termination criteria, the code changes to the next assessment tool and modifies the parameters of the algorithm to ensure the exploration of the new objective function. This is repeated for the third assessment tool (EnergyPlus in our application) and when the algorithm reaches a termination criteria exclusive for this stage the run is finished.

One of the mechanisms that can be read in [18] to improve the efficiency of algorithms in dynamic environments is to maintain diversity. In the methodology shown here, the algorithm will change to the next assessment method before the population has converged completely.

The CMA-ES recognise changes in the objective landscape through the CM, if the change on the assessment tool generates a substantial change in the objective landscape; it is expect that the algorithm would change the parameters of its operators to adapt to this change.

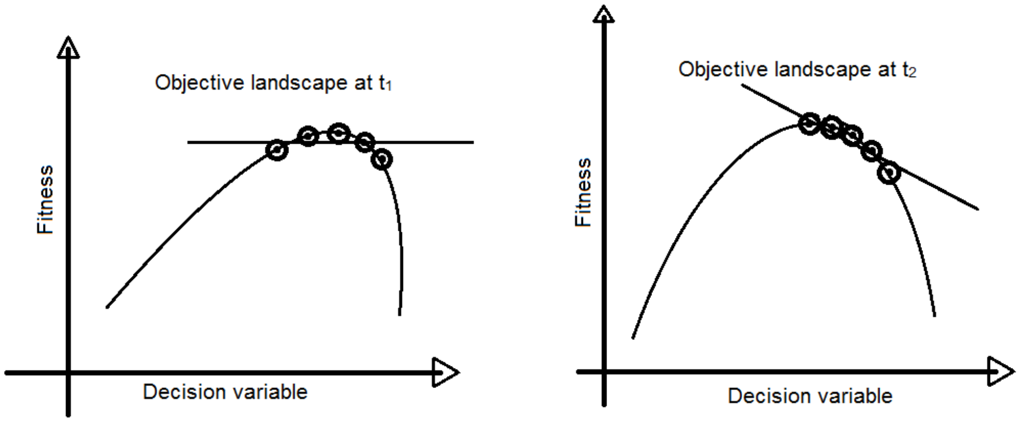

Figure 2 - Interpretation of the recognition of the landscape by the CM in one dimension: The straight line shows a linear regression of the solutions (circles). The same population that shows a linear regression with slope close to zero in the left, shows a linear regression with negative slope after a change in the objective landscape.

The CMA-ES is self-adaptive; this means that the algorithm calculate the next move depending on the population and on which direction the population should move. This is equivalent to calculating the gradient in continuous derivable functions, but CMA-ES calculates this on a discrete way (through the CM). As the variability of the objective function changes less with variations of the decision variables (equivalent to a gradient of zero) the algorithm makes smaller movements, this is equivalent to focusing on a smaller area. If the members of a population are over a peak in the objective function, the algorithm will see no substantial modification of the objective function with the covariance matrix of the individuals, if the landscape changes and they appear in the next iteration in an area with a substantial slope, the algorithm would detect that a movement needs to be done to the population and therefore will modify the scope of the mutation to make that happened. The CMA-ES, also internally modifies the probability of direction of the mutation to aim for the areas of the landscape likely to give high fitness individuals (illustrated in Figure 2).

Hansen et al. used the CMA-ES on the presence of uncertainties before in: [29]. However this was not a dynamic optimisation problem or optimisation with surrogate models.

4 Application

To evaluate the strengths and weaknesses of the method, it was applied to a building design problem.

This application tests the capability of an evolutionary optimisation algorithm (CMA-ES) to solve an optimisation problem in which the objective landscape varies due to changes in the technique for calculation of the objective function. The optimisation algorithm selected is the CMA-ES, and the three techniques for calculating the objective function are: an LT-calculation, an LPM simulation and EnergyPlus. The use of the CMA-ES with sequential assessment has been called CMA-ES-SA.

The problem was defined to emulate real optimisation problems that can be found in architecture. The optimisation was single objective, but the method can be easily adapted to run multi-objective problems. The objective was the energy demand (heating and cooling) as this thesis has been highly motivated by the current need of reducing energy use in buildings.

The optimisation problem at hand is the design of an office that is constrained to have 70 m2 and being only one storey, as it is supposed to be part of an office block. The gains have been extracted from [30] and represent the benchmark values and are shown in Table 1.

The decision variables include the properties of the fabrics of the envelope, the material properties of the internal partitions, the fenestration, and overhangs for windows in the south, east and west façade. Also, the aspect ratio and the infiltration levels were considered as decision variables. All the variables can be found in Table 2.

Table 1 - Benchmark values for internal heat gains for offices.

Density of occupation [person/m^2]	Heat gain [W/m^2]		
	People	Lighting	Equipment
1/8	10	12	20

Many decision variables have been selected in this problem because this methodology is trying to offer an alternative to optimisation methods that use meta-models with large computational time needed upfront to build the surrogates. With this application, it will be tested if the methodology performs well with large decision spaces (many decision variables), where the constructions of meta-models are not viable due to their long computational times.

The decision variables included in the problem control almost all thermally relevant elements of the building (see Table 2). Some of the elements are more relevant to the overall energy demand than others, these decision variables have been chosen in propose, to verify the capabilities of the algorithm to focus in the variables that are more than others along the optimisation.

Table 2 - Variables forming the decision space. IP are internal partitions.

	Variable	Type	Lower bound	Upper bound	Unit
	Infiltration	Real	0.021	0.6	*ach*
	Aspect Ratio	Real	0.3	3	*m/m*
	Fenestration, North (%	Real	12	80	*%*
	Fenestration, South (%	Real	12	80	*%*
	Fenestration, East (%	Real	12	80	*%*
	Fenestration, West (%	Real	12	80	*%*
	Wall Type	Symbol.d	Construction	Construction D	*n/a*
	Insulation	Real	100	500 (U-Value=0.1)	*mm*
	Conductivity of IP	Real	0.2	2.3	*W/(mK)*
	Capacity of IP	Real	200	3000	*J/(kgK)*
South Overhang	Depth	Real	0.0	2.0	*m*
	Left extension	Real	0.0	5.0	*m*
	Right extension	Real	0.0	5.0	*m*
East Overhang	Depth	Real	0.0	2.0	*m*
	Left extension	Real	0.0	5.0	*m*
	Right extension	Real	0.0	5.0	*m*
West Overhang	Depth	Real	0.0	2.0	*m*
	Left extension	Real	0.0	5.0	*m*
	Right extension	Real	0.0	5.0	*m*

Table 3 - Possible constructions of solutions.

Construction	Outside Layer	Intermediate	Inside Layer
A	200mm concrete	Insulation	25mm stucco
B	200mm ’’	’’	200mm concrete
C	25mm stucco	’’	200mm ’’
D	25mm ’’	’’	25mm stucco

4.1 The algorithms

Two algorithms have been used in this application, the one created in this thesis: the CMA-ES using sequential assessment (CMA-ES-SA) and a canonical form of GA. GAs are popular optimisation methods broadly used in the literature of building design and therefore, they seem as an adequate algorithm to take as the baseline. Several works can be found in the literature were GAs where used, examples of this in building design are: [2, 4-7, 31-37].

Both optimisation algorithms have been implemented in Octave (similar to Matlab). The genetic algorithm has been given the values recommended by Schaffer in [38] for crossover probability and mutation probability (Table 4), the algorithm has been run with 100 individuals, and uses the stochastic universal sampling as selection mechanism [39]. The CMA-ES has been given the default parameters provided by the author (Table 5) [28].

The decision variables have been normalised into the interval [0, 10] as recommended by Hansen [28] for both algorithms.

Table 4 - Default parameters of the GA, from [38].

Parameter	Value
Population size	100 individuals
Selection mechanism	Stochastic universal sampling
Crossover probability	0.75 crossovers per couple
Mutation probability	0.005 mutations per bit

Table 5 - Default parameters of the CMA-ES, from [28].

Parameter	Value
Population size	12 = (4+3*ln(dimensions))
Number of parents	6 = (population size / 2)

The algorithms have been provided with specific features to make sure that the optimisation is performed adequately. In the case of the CMA-ES-SA, the algorithm is able to detect that the population is converging into a few points (losing diversity) by checking the value of the step-size of mutation (called sigma 'σ'). When the step-size is smaller than a given value, the algorithm changes automatically the assessment method or terminates the run depending on the current stage. The actions are shown in Table 6.

Table 6 - Initial, final and intermediate steps of the CAM-ES-SA.

Assessment method	Action at the start of using the method	Termination/Change of assessment method
LT-Method	Initialisation of the algorithm	$\sigma < 0.5$ or simulations > 500
LPM	Nothing	$\sigma < 0.1$ & simulations > 1000
EnergyPlus	Nothing	$\sigma < 0.09$ or stagnation

When using the GA the decision space has to be discretised for every decision variable, in this problem, the decision space has been discretised in 50 values per variable. This means that, for the GA, the decision variables will have a precision of 10/50=0.2 units. This accuracy is translated differently in the physical values depending on the range of each one. This encoding represents 9.54×10^{33} possible solutions in the decision space.

Evolutionary Strategies (ESs) do not need discretization of the decision space as the variables are kept in real format. However, to make sure that advantage is not given to any of the methods, the CMA-ES-SA is allowed to evolve only until it reaches the same levels of accuracy of the discretised space of the GA (0.2 units). The GA can be run for as many generations as the operator decides, but the ES is self-stopped by definition when certain accuracy is reached. To make sure that the judgment of the computational times is fair, the GA has been stopped when the solution is similar to that obtained by the CMA-ES-SA. The details of discretisation and termination are summarised in Table 7.

Table 7 - Parameters of the optimisation algorithms.

	Precision of variables	Termination
GA	0.2 units (=10/50)	objective < solution of CMA-ES-SA
CMA-ES-SA	No predefined limit	σ < 0.09 (Table 6)

The CMA-ES-SA has its most substantial strength on using surrogate models to analyse the solutions along the optimisation. These surrogate models are different methods of calculating the heating and cooling demands. The following section shows how the building was modelled for each of these assessment methods.

4.2 The modelling

This section shows the way the building was modelled in each of the assessment tools.

1.1.1 Model for the LT-Method

The office modelled for this work, is located in London, therefore being mid European coastal, it is considered to have each of the façade orientated to each of the cardinal points. The fenestration percentage of each façade is an independent decision variable and it is considered that these windows are not affected by casted shadows from surrounding obstacles. The LT-Method suggests separating passive areas to non-passive areas and so was done for this application. The passive areas are delimited by external walls and the imaginary surface parallel to them 6 meters into the zone, this multi-zone configuration was used for the EnergyPlus model too.

1.1.2 Model for the LPM

The building created with the characteristics defined by the decision variables was created for each solution, and then the equivalent RC-network was obtained. The RC-network representing all the elements of the building was then reduced to a LPM with the methodology shown in [40]. The office is modelled as a single zone, to make possible its representation by the LPM. The model does not include ceiling and floor, as this surfaces are considered adiabatic because the office above and below the one studied were considered at the same temperature. The computational time needed to run a yearly simulation with this model was 0.29 seconds

To perform the simulation, the model has been solved with an ideal heating and cooling system, but also with natural ventilation.

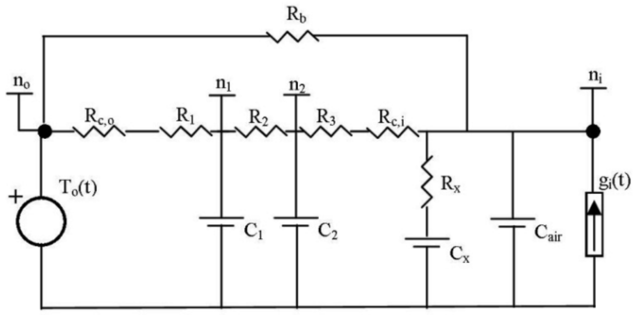

Figure 3. LPM of the building.

The software calculates the cooling load for each time step and it checks if that demand can be covered with natural ventilation by window opening (the software recognise how much air can be moved depending on the windows size and the external wind speed), if this load can be totally covered by natural ventilation, the windows will open the appropriate fraction to provide the adequate cooling. In the case that this is not enough, ventilation will happened first, and the cooling load will be calculated on top. With this the model is trying to mimic the capability of the more complex system used in EnergyPlus that is able to adjust the airflow taken from the outside to cool down the inside of the office.

The solar gains are calculated in a basic way for this simulator. The solar radiation per square meter in each time-step was calculated and reported for each of the surfaces of the building (North, South, East and West) using EnergyPlus before performing the optimisation and stored in a data base. For each solution, the fenestration area of each façade will be multiplied in each time-step by the appropriate value and the summation of those will give the total solar gain for each time-step. The gains due to metabolic, light and electric appliances were the same as in the run with EnergyPlus.

This model has been solved using state-space equations, and boundary conditions read from a weather data file of London ([41] as with EnergyPlus). The integration time-step of the equations was of 0.1 hours (6 minutes).

1.1.3 Model for EnergyPlus

The simulation in EnergyPlus is the most complex of all three. The way the building was modelled in this simulator is as follows.

The office is modelled in EnergyPlus as a single story office in a multi-storey block, with no exchange of heat through floor and ceiling celling and floor (adiabatic).

The conditioning equipment is an air-based system that uses an electric chiller and a gas furnace to deliver cold and warm water respectively that circulates through coils on the outlets of the air ducts for each area. The conditioning system is able to use un-conditioned outside air to cool down the office, what is equivalent to the natural ventilation modelled in the LPM simulator.

The model is multi-zone as the one described for the LT method with the addition of a plenum zone that gathers the air flow from the zones and returns it to the conditioning system. The geometry of the office with an aspect ratio of 1.0 and with arbitrary windows and overhangs can be seen in Figure 4.

The office has been surrounded by four surfaces with the same height of its walls but located at 30 meters away of each façade, with this; the shadow of potential buildings is represented.

The objective function is calculated after summing up the heating demand and the cooling demand multiplied by a constant factor. This factor accounts for the different price of primary energy use for the chiller (electricity) compare with the furnace (gas). The factor was the same in the three assessment methods.

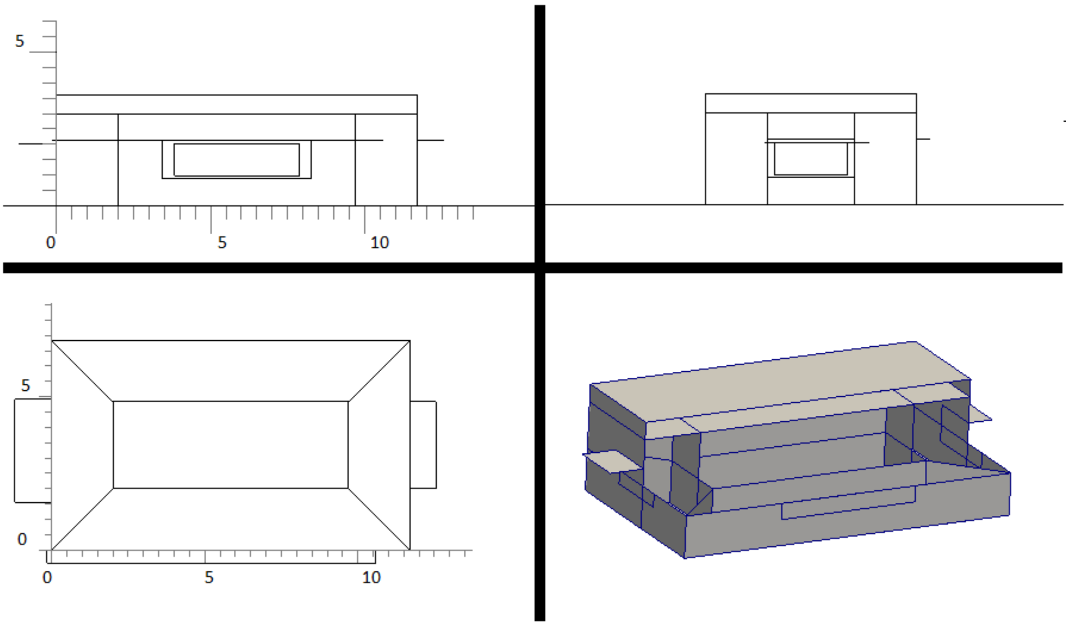

Figure 4 – Example of the 3D geometrical model of the office used for this work, the overhangs, windows and aspect ratio vary. The units are metres.

The next section shows the results of using the CMA-ES-SA to minimise the annual energy demand (heating and cooling) for the application described.

5 Results and discussion

The sequential optimisation methodology, CAM-ES-SA, was applied to the problem described before: the optimisation of an office in a block located in London with 70 m^2 floor area and with the decision variables as shown in Table 2. In order to study the benefits of the approach, the same optimisation problem was solved using a traditional GA and the CMA-ES-SA. To verify the solutions that can be obtained using only the LPM and the LT-method, two more runs were performed.

To evaluate properly the accuracy of the method, the two approaches, the GA and the CMA-ES-SA have been applied several times and the results of these runs studied.

Firstly, 20 runs were done using the CMA-ES-SA. The solutions of this set are represented in Figure 5b.

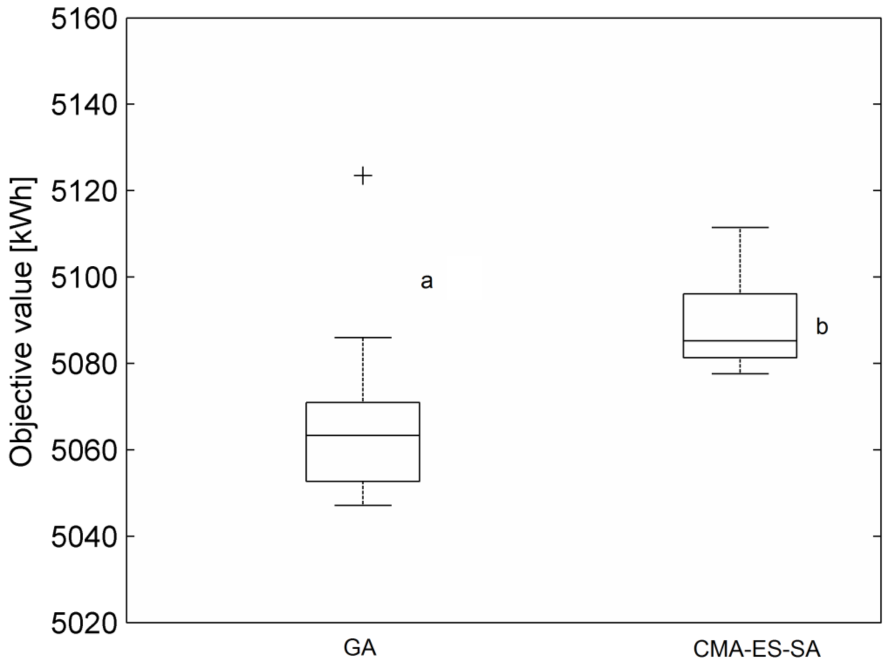

Figure 5 - Boxplot of the results obtained with the traditional GA (a) and the CMA-ES-SA (b). The single point represent an outliner in both cases: Points are drawn as outliers if they are larger than q3 + w(q3 – q1) or smaller than q1 – w(q3 – q1), where q1 and q3 are the 25th and 75th percentiles, respectively.

The same optimisation problem was solved using the GA; in this case, 9 runs were performed. The parameters needed to achieve solutions of the order of those that were obtained with the CMA-ES-SA were investigated. A population size of 100 individuals and a number of generations of 140 came out as the most efficient combination; the results of the nine runs using the GA are shown in Figure 5a.

The variability of the solutions obtained with the CMA-ES is of 30.40 kWh, around 0.6% of the average value of the objective function. This variation between solutions, although substantial in numerical algorithms theory, it is negligible in building energy calculations[2]. Several authors have shown that energy calculations have much larger variability due to uncertainties [42] [43] [44] [45] [46].

[2] The difference between using a desktop computer, with a nominal power of 100W, and using a laptop with a nominal power of 50W, 8 hours a day 5 days a week, sums up a total difference in energy demand of 104 kWh over the year.

Figure 5 shows that the two optimisation methods were run until reaching solutions with similar solutions as imposed by the termination criteria. After seeing that both algorithms were run in a way in which they delivered the same accuracy in the solution, the computational times were studied. The result is shown in Figure 6.

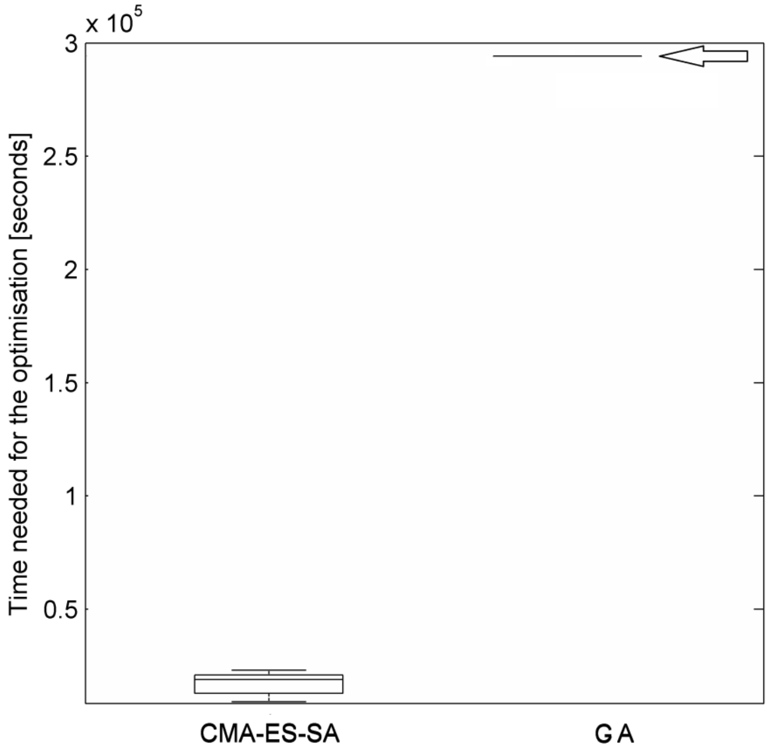

Figure 6 - Time differences in running the two optimisation methods. The time for the GA is always the same as the generations and population size has been fixed to achieve the same accuracy as the CMA-ES-SA, the time of the CMA-ES-SA varies.

The CMA-ES-SA used different times for every run, as the algorithm is stochastic and this makes the times at which the assessment methods are changed different, also the speed of converging in each of the steps is different. In opposition, the GA was always run with 100 individuals and 140 generations, and therefore makes always 14000 evaluations of the objective functions, the time needed by the CMA-ES-SA is a fraction of the time needed by the GA to get to almost the same solutions.

Figure 6 shows that the differences on times are substantial, even in the worst case scenario the CMA-ES-SA is much faster than the GA.

To understand better the solution to the optimisation problem in both cases, one solution from each of the set of the runs was studied in more detail.

One of the results from the optimisation using the CMA-ES-SA with and objective value of 5080.97 kWh was chosen, the solution using the GA with an objective value of 5086.00 kWh was selected for this comparison. The value of the decision variables for both of these solutions are shown in Table 8.

Table 8 - Results of the optimisation runs using specific assessment tools, and sequential optimisation.

				Genetic Algorithm			CMA-ES-SA
	Variables		Units	LT	LPM	e+	Sequential
	1	Infiltration	ach	n/a	0.021	0.021	0.021
	2	Aspect ratio	m/m	0.374	1.098	1.326	1.673
	3	U-Value wind.	W/(mK)	n/a	1.966	1.890	1.921
	4	North Window	%	12.7	22.1	12.01	12.56
	5	South Window	%	14.0	12.1	33.78	23.31
	6	East Window	%	21.9	12.0	12.10	12.60
	7	West Window	%	47.0	12.0	12.05	12.18
	8	Wall Type	symbolic	n/a	C	C	C
	9	Insulation	mm	n/a	499	499.7	500.0
	10	k - partitions	W/(mK)	n/a	1.85	2.39	2.398
	11	c_p – partitions	J/(kgK)	n/a	2976	2999	2989
South Overhang	12	Depth	m	n/a	n/a	1.050	0.774
	13	Left extension	m	n/a	n/a	1.614	2.485
	14	Right extension	m	n/a	n/a	1.549	2.060
East Overhang	15	Depth	m	n/a	n/a	0.665	0.742
	16	Left extension	m	n/a	n/a	0.172	3.660
	17	Right extension	m	n/a	n/a	0.935	4.740
West Overhang	18	Depth	m	n/a	n/a	1.828	0.610
	19	Left extension	m	n/a	n/a	1.433	4.435
	20	Right extension	m	n/a	n/a	0.026	2.790
Best objective			kWh	28014	5197	5086.00	5080.97
Evaluations				7000	14000	14000	1009+ 1716+ 1356
Time per simulation			s	0.001	0.29	16.70	0.001, 0.29, 16.7
Total time			s	7	40600	**238,000**	**23,551**

Two other runs using the GA and only the LT-method and only the LPMs have been also included in Table 8 for illustrative proposes.

Table 8 shows the values of the decision variables together with other parameters of the optimisation of the solutions found by different methods. The first set of decision variables (from 1 to 11), are the interpreted by the LPM simulation, and are supposed to be the most influential in the energy demand of the office. Most of the values for these variables are very similar using the GA and the CMA-ES-SA, and the ones that are not similar, show an interesting fact: the aspect ratio and the windows size lead to similar sizes of windows. It can be seen that the aspect ratio in the solutions from the GA and the CMA-ES-SA are different, this ratio, is going to have an impact in window size, as the last one is defined as a percentage of the area of the façade. If one observes the aspect ratio and the size of the window in the south façade, one can see that although the window in the solution of the CMA-ES-SA is smaller, the aspect ratio is larger, and therefore, the effect of both decision variables together make the solution more similar to the solution in the GA.

Another interesting fact to note from Table 8 is the failure of implementing the overhangs properly in the optimisation. For the assessment tool, an overhang that extends 4 meters over the window is as valid as one that extends 0.5 metres; no penalisation was put in place for solutions that have overhangs that are too large. Also, if an overhang needs to be at least, let it be said 1 meter, and the simulator recognise that having an overhang of 5 meters is as good in the energy demands, then any value between 1 and 5 will be chosen by the algorithm and therefore, the user will not be able to know what is best, as he/she would not be able to recognise from the solutions the “at least x meters” condition. Including this type of decision variables in the algorithms in the future can be challenging and would need to be done properly in the future if this method is applied.

It can be seen in the number of function evaluations, that the CMA-ES-SA need less than the GA even without differentiating between the stages with different assessment tools. The GA needed 14,000 function evaluations, and the CMA-ES-SA needed 4,081 in this example. This could be because using surrogate models allow the algorithm to concentrate in the variables that mater the most in each stage; because the efficiency of the CMA-ES itself is higher than the efficiency of the GA or because a combination of the two. Observing the comparisons done in [47] one could assume that CMA-ES would always be more efficient.

The number of function evaluations that were needed using the computationally expensive simulator (EnergyPlus) are few when using the sequential optimisation (CMA-ES-SA). This will have the greatest impact on the total computational time. Only 33.23% of the function evaluations are carried out with EnergyPlus, the rest, 66.77% of the function evaluations, are done with the other two assessment tools (surrogate models), although this is a large percentage of the evaluations, it only accounts for 2.15% of the total computational time of the optimisation as these are assessments that are “cheap” to run. The benefits of using this pre-processing in the optimisation using the surrogates models are clearly shown in this case; the extra computational time of this pre-processing with surrogates is negligible (~8 minutes) compared with the savings in computational time that produced when compared with a traditional GA (~two and a half days, or 214450 seconds). A representation of how the computational time is spread in the optimisation using the CMA-ES-SA can be seen in Figure 7.

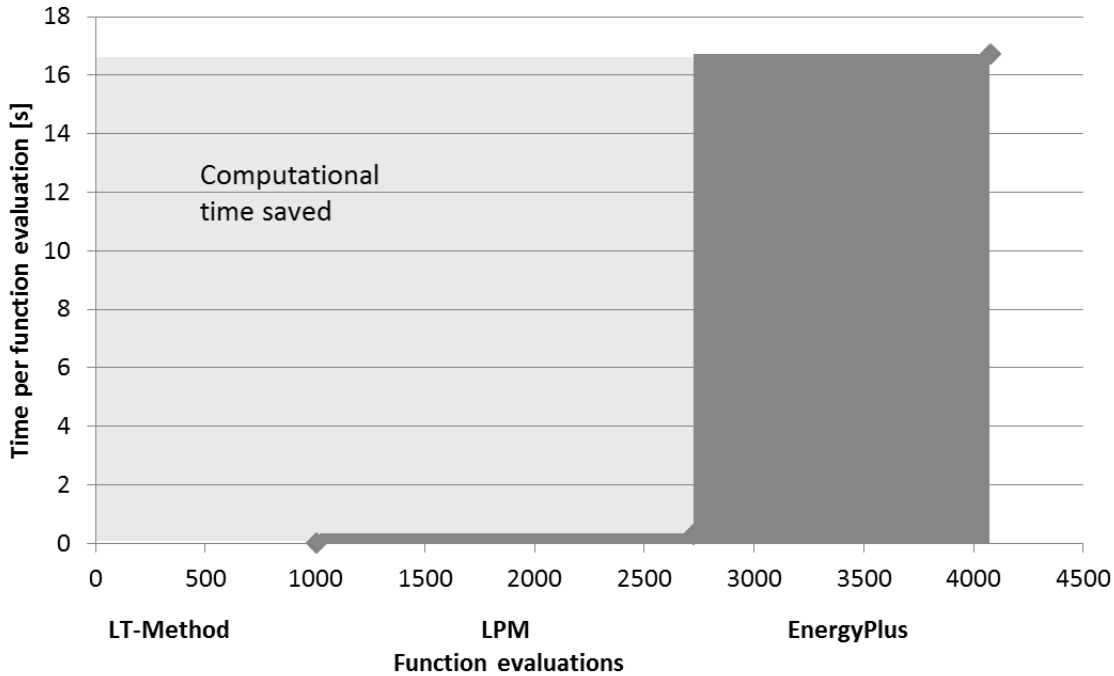

Figure 7 - Graph showing the time spent in each stage of the optimisation using the CMA-ES-SA for the case shown in Table 8. Time is represented by areas in this graphs and measured in seconds. The computational time saved has been shown as the light grey area.

If one observes the results of the optimisations performed by using only one of the surrogates models (either the LT-method or the LPMs) in Table 8, one can see that the runs carried out using the LT-method as the assessment tool, although much faster, could not reach a low-energy design. This was expected as the LT method is a very basic tool that does not even interpret the majority of the variables. In the case of the solution found using only LPMs, the optimum found is close to the minimum found when using GA and EnergyPlus (around 2%). This justifies the selection of these simulators for optimisation problems as was done by [5] and [6]. One could think after seeing this result that one should go straight to the use of LPMs for optimisation; however, it should be remembered that the problem at hand here is rather simplistic. There is no detailed study of the air flows, illumination or other components. In other cases, where real buildings are being created, a much higher level of accuracy would be needed. It is believed that the CMA-ES-SA has a great potential for those problems.

Figure 9 and Figure 8 show the evolution of the optimisation using the GA and the CMA-ES-SA respectively for the examples shown in Table 8.

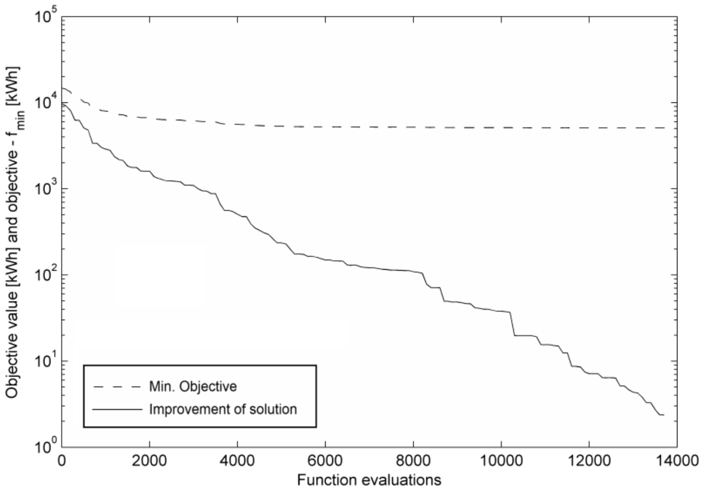

Figure 8 - Evolution of the GA using EnergyPlus. Minimum value of the objective function found (dashed line). Relative improvement of the objective value found until then (solid).

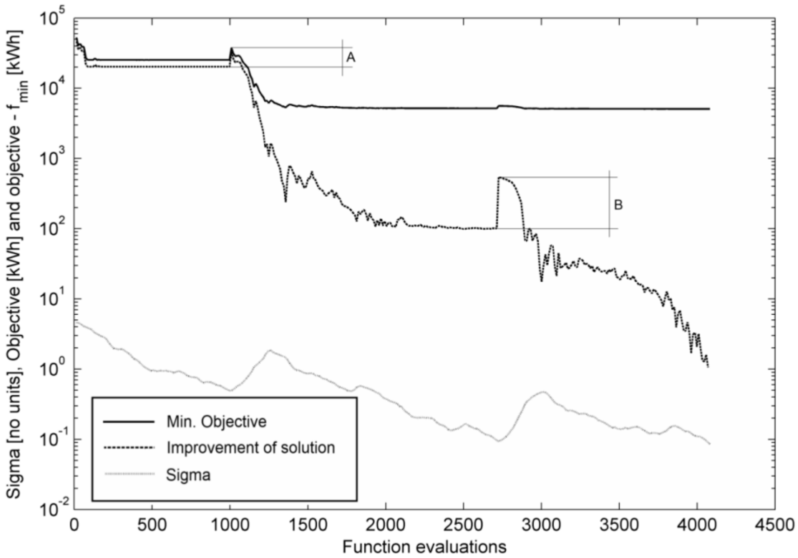

Figure 9 - Evolution of the CMA-ES-SA. "A" shows the differences between the objective values of the best solution at the moment of changing the assessment tool from the LT-Method to the LPM. "B" shows the differences between the objective values of the best solutions at the moment of changing the assessment tool from the LPM to EnergyPlus.

Figure 8 shows that, in the GA, the solution get improved with a relative change of 10^4 at the beginning (the difference in objective value in the best individual from a generation and to the best individual of the next is of that order), and after 14000 function evaluations refines to relative improvements of the order of 2 kWh. In the case of the CMA-ES-SA, the improvement on the objective value of the solutions is steeper. It can be seen that in the stage of the optimisation where LPMs are used (from the point at the optimisation where "A" is marked to the point where "B" is marked) the accuracy of the search grows rapidly getting an improvement of the solution of the order of 100kWh in about two thousand function evaluations. This can also be seen in the stage where EnergyPlus is used, in which the improvement of the solutions goes from around 500 kWh to 1 kWh in around 1300 evaluations.

This rapid convergency of the method explains the higher efficiency of the CMA-ES-SA, it can be seen that the GA gives a constant logarithmic improvement of the solution. However the CMA-ES-SA shows that at the beginning of each change of the assessment tool, the algorithm improves the solution rapidly, and making the assessment sequential provides 3 such stages.

Finally it should be said that two of the 20 optimisations done with CMA-ES-SA failed as they converged immediately after changing to EnergyPlus. The similarities in the objective landscape when the assessment tool was the LPM and when the assessment tool was EnergyPlus, made the algorithm not able to recognise the change and therefore diversity was not augmented. The termination criteria (sigma <0.09) was reached before the CMA-ES-SA was able to explore the new landscape provided by EnergyPlus. This problem is easily solvable by imposing a minimum number of function evaluations for EnergyPlus. In this case a minimum of 100 function evaluations would be enough, as this would allow the algorithm to increase the step-size and eliminate the potential premature convergency.

The CMA-ES is effective in adapting the parameters of the optimisation to new landscapes, and therefore in solving this kind of optimisation problem. If one observes the step-size (sigma) of the CMA-ES-SA in Figure 9, one can see that at the points where "A" and "B" are located (i.e. where the assessment tool changes), sigma starts getting larger. This is because the algorithm is able to recognise the changes in the landscape, and starts increasing the step-size to make sure that the new landscape is explored properly. However, it would appear that the algorithm should be forced to run a minimum of function evaluations at each stage to ensure that the internal mechanisms of the algorithm are able to recognise the new objective landscape. The premature convergency is likely to happen when the sigma required to change the assessment tool is similar to the sigma required to terminate the optimisation, as happened in this case.

The algorithm was able to cope well with the symbolic variable, providing the same results in the CAM-ES-SA run and the GA run.

6 Conclusions

The methodology suggested in this paper has been seen to need less computational time in this application than a GA that uses only one assessment tool to evaluate the solutions.

It is seen that the optimum found using the LPM simulator is similar to that found using EnergyPlus. This justifies the use of this kind of model in previous publications [5] or [6].

Several software developers, such as IES-<VE>, and EnergyPlus are integrating optimisation modules into their packages[3,4]. These new modules can have long computational times if the objective functions are always evaluated using the whole comprehensive dynamic simulator; this paper shows that there are more efficient ways of searching for optimal solutions in building design and those are using self-adaptive optimisation methods and surrogate models. This method is similar to the natural method of building design, where simplified tools are used in the early stages when many variables have to be determined, and only at later stages more complex tools are used. There are also clear analogies to product development or scientific thought, where a series of ever more complex models are created with piece meal optimisation in between.

The results of this work are in general satisfactory. However, improvements are possible; for example, choosing a different number of building assessment tools, or changing the criteria that make the algorithm jump from one assessment tool to the next, would increase efficiency further. It is though that in the case of running an optimisation, where the maximum level of accuracy requires running simulations that take the order of hours, the CMA-ES-SA could show a much better reduction in the computational times compared with the traditional method.

7 References

[1] Crawley DB, Hand JW, Kurnmert M, Griffith BT. Contrasting the capabilities of building energy performance simulation programs. Build Environ. 2008;43:661-73.

[2] Tuhus-Dubrow D, Krarti M. Genetic-algorithm based approach to optimize building envelope design for residential buildings. Build Environ. 2010;45:1574-81.

[3] GAMA. Consumers Directory of certified efficiency ratings for heating and water heating equipment. Gas Appliance Manufacturers Association; 2005.

[4] Chantrelle FP, Lahmidi H, Keilholz W, El Mankibi M, Michel P. Development of a multicriteria tool for optimizing the renovation of buildings. Appl Energ. 2011;88:1386-94.

[5] Coley DA, Schukat S. Low-energy design: combining computer-based optimisation and human judgement. Build Environ. 2002;37:1241-7.

[6] Wright JA, Loosemore HA, Farmani R. Optimization of building thermal design and control by multi-criterion genetic algorithm. Energ Buildings. 2002;34:959-72.

[7] Magnier L, Haghighat F. Multiobjective optimization of building design using TRNSYS simulations, genetic algorithm, and Artificial Neural Network. Build Environ. 2010;45:739-46.

[8] Barthelemy JFM, Haftka RT. Approximation Concepts for Optimum Structural Design - a Review. Struct Optimization. 1993;5:129-44.

[3] see the optimisation tool for Design Builder (EnergyPlus), developed in the research project ADOPT with de Monfort University with Yi Zhang as Principal Investigator (PI). (Not launched yet).

[4] see the optimisation tool for IES-Virtual Environment, developed in the research project OPTIMISE with Loughborough University among other industrial partners. (Launched 5 September 2012)

[9] Booker AJ, Dennis JE, Frank PD, Serafini DB, Torczon V, Trosset MW. A rigorous framework for optimization of expensive functions by surrogates. Struct Optimization. 1999;17:1-13.

[10] El-Beltagy MA, Keane AJ. Metalmodeling techniques for evolutionary optimization of computationally expensive problems: promises and limitations. Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-99). Orlando, USA1999. p. 196-203.

[11] Forrester AIJ, Sobester A, Keane AJ. Multi-fidelity optimization via surrogate modelling. P Roy Soc a-Math Phy. 2007;463:3251-69.

[12] Jansson T, Nilsson L, Redhe M. Using surrogate models and response surfaces in structural optimization - with application to crashworthiness design and sheet metal forming. Struct Multidiscip O. 2003;25:129-40.

[13] Koziel S, Bandler JW, Madsen K. Space Mapping With Adaptive Response Correction for Microwave Design Optimization. Ieee T Microw Theory. 2009;57:478-86.

[14] Queipo NV, Haftka RT, Shyy W, Goel T, Vaidyanathan R, Tucker PK. Surrogate-based analysis and optimization. Prog Aerosp Sci. 2005;41:1-28.

[15] Ong YS, Nair PB, Keane AJ, Wong KW. Surrogate-Assisted Evolutionary Optimization Frameworks for High-Fidelity Engineering Design Problems. In: Jin Y., editor. Knowledge Incorporation in Evolutionary Computation. Berlin: Springer; 2004.

[16] Jin R, Chen W, Simpson TW. Comparative studies of metamodelling techniques under multiple modelling criteria. Struct Multidiscip O. 2001;23:1-13.

[17] Wetter M, Wright J. A comparison of deterministic and probabilistic optimization algorithms for nonsmooth simulation-based optimization. Build Environ. 2004;39:989-99.

[18] Cruz C, Gonzalez JR, Pelta DA. Optimization in dynamic environments: a survey on problems, methods and measures. Soft Comput. 2011;15:1427-48.

[19] Goldberg DE. Genetic algorithms in search, optimisation, and machine learning. New York: Addison-Wesley Publishing Company; 1989.

[20] Branke J. Evolutionary Approaches to Dynamic Optimization Problems-Updated Survey. GECCO WORKSHOP on Evolutionary Algorithms for Dynamic Optimization Problems2001. p. pp. 27-30.

[21] Bäck T, Fogel DB, Michalewicz Z. Handbook of Evolutionary Computation. Oxon: Taylor & Francis Group; 1997.

[22] Coley DA. An Introduction to Genetic Algorithms for Scientists and Engineers. Singapore: World Scientific; 1999.

[23] Beyer H-G, Schwefel H-P. Evolution strategies – A comprehensive introduction. Natural Computing. 2002;1:3-52.

[24] Bäck T, Schwefel H-P. An Overview of Evolutionary Algorithms for Parameter Optimization. Evolutionary Computation. 1993;1:1-23.

[25] Deb K. Multi-Objective Optimization Using Evolutionary Algorithms. Chichester: John Wiley & Sons; 2001.

[26] Hansen N, Müller SD, Koumoutsakos P. Reducing the Time Complexity of the Derandomized Evolution Strategy with Covariance Matrix Adaptation (CMA-ES). Evolutionary Computation. 2003;11:1-18.

[27] Kampf JH, Robinson D. A hybrid CMA-ES and HDE optimisation algorithm with application to solar energy potential. Appl Soft Comput. 2009;9:738-45.

[28] Hansen N. CMA-ES for Octave/Matlab. v3.0 ed. http://www.lri.fr2012.

[29] Hansen N, Niederberger ASP, Guzzella L, Koumoutsakos P. A Method for Handling Uncertainty in Evolutionary Optimization With an Application to Feedback Control of Combustion. Ieee T Evolut Comput. 2009;13:180-97.

[30] CIBSE. Guide A: Environmental design. London: The Chartered Institution of Building Services Engineers 2006.

[31] Ooka R, Komamura K. Optimal design method for building energy systems using genetic algorithms. Build Environ. 2009;44:1538-44.

[32] Kayo G, Ooka R. Building energy system optimizations with utilization of waste heat from cogenerations by means of genetic algorithm. Energ Buildings. 2010;42:985-91.

[33] Caldas LG, Norford LK. A design optimization tool based on a genetic algorithm. Automation in Construction. 2002;11:173-84.

[34] Suga K, Kato S, Hiyama K. Structural analysis of Pareto-optimal solution sets for multi-objective optimization: An application to outer window design problems using Multiple Objective Genetic Algorithms. Build Environ. 2010;45:1144-52.

[35] Wang WM, Zmeureanu R, Rivard H. Applying multi-objective genetic algorithms in green building design optimization. Build Environ. 2005;40:1512-25.

[36] Caldas LG, Norford LK. Genetic algorithms for optimization of building envelopes and the design and control of HVAC systems. J Sol Energ-T Asme. 2003;125:343-51.

[37] Juan YK, Kim JH, Roper K, Castro-Lacouture D. GA-based decision support system for housing condition assessment and refurbishment strategies. Automation in Construction. 2009;18:394-401.

[38] Schaffer JD, Caruana RA, Eshelman LJ, Das R. A Study of Control Parameters Affecting Online Performance of Genetic Algorithms for Function Optimization. Proceedings of the Third International Conference on Genetic Algorithms. Firefax, VA1989.

[39] Baker JE. Reducing bias and inefficiency in the selection algorithm. In: J Grefenstette (Hillsdale N.E., editor. Proc 2nd Int Conf on Genetic Algorithms. Cambridge, MA1987. p. pp 57–62.

[40] Ramallo-González AP, Eames ME, Coley DA. Lumped Parameter Models for Building Thermal Modelling: An Analytic approach to simplifying complex multi-layered constructions. Energ Buildings. 2013;60:174-84.

[41] DoE. London-Gatwick, weather datafile. In: Energy U.S.D.o., editor.2011.

[42] Eames M, Kershaw T, Coley D. The appropriate spatial resolution of future weather files for building simulation. J Build Perform Simu. 2011;5:1-12.

[43] de Wilde P, Tian W. Identification of key factors for uncertainty in the prediction of the thermal performance of an office building under climate change. Building Simulation. 2009;2:157-74.

[44] Wang L, Mathew P, Pang X. Uncertainties in energy consumption introduced by building operations and weather for a medium-size office building. Energ Buildings. 2012;53:152-8.

[45] Pettersen TD. Variation of Energy-Consumption in Dwellings Due to Climate, Building and Inhabitants. Energ Buildings. 1994;21:209-18.

[46] Hopfe CJ, Hensen JLM. Uncertainty analysis in building performance simulation for design support. Energ Buildings. 2011;43:2798-805.

[47] Bäck T. Evolutionary Algorithms in Theory and Practice. New York: Oxford Univerity Press; 1996.