Energy and Buildings 81 (2014) 18–29

Contents lists available at ScienceDirect

Energy and Buildings

journal homepage: www.elsevier.com/locate/enbuild

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

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

[a] College of Engineering, Mathematics and Physical Sciences, University of Exeter, UK
[b] Department of Architecture and Civil Engineering, University of Bath, UK

ARTICLE INFO

Article history:
Received 31 March 2013
Received in revised form 19 May 2014
Accepted 27 May 2014
Available online 4 June 2014

Keywords:
Optimisation
Evolutionary strategy
Dynamic
Sequential
Energy
Building

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. Normally 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 than the optimisation methods found in the literature.

1. Introduction

The new measures to reduce carbon emissions in the building sector have increased the interest in producing low-energy designs, these being 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 (which are getting more and more complex) are able to represent the building physics and any energy systems in a very comprehensive manner [1]. Among the phenomena that can be modelled with this kind of software one could find heat transfer, phase change, moisture transfer, pollutant emissions, air movement and others. Although computers have become more powerful with time, this improvement does not overcome the growth in complexity of building simulators, this leads to the substantial run times found when running an annual simulation on a personal workstation. These simulation times can be of the order of hours, making the process of investigating several designs slow and tedious.

Also, with the new requirements from clients and regulations on the design of low-energy buildings, architects and engineers have been passed the obligation of designing buildings that have to perform in several aspects. Assisted design helps the building professionals to produce these designs, as building simulators and virtual prototyping allow them to evaluate the quality of the buildings at a very low cost and with the possibility of exploring a large number of options. Assisted design has opened the door to automatic design, where the computer tools themselves explore the options for a design, but also select the best candidates. These techniques, which are already popular in research [2,3], are starting to be implemented in commercial software.[1,2] It is intended that with

* Corresponding author at: The Innovation Centre (Phase 1), University of Exeter, Rennes Drive, Exeter, Devon EX4 4RN, UK. Tel.: +44 01392 724144.
E-mail addresses: ar321@exeter.ac.uk, aprg20@bath.ac.uk (A.P. Ramallo-González).

[1] 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 at the time of writing).

[2] 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).

http://dx.doi.org/10.1016/j.enbuild.2014.05.037
0378-7788/

these tools the time used by the building professionals will be optimised, as the exploration of options will be done internally in the software tools, and they will be able to use higher level knowledge into assessing the filtered options provided for optimisation algorithms.

These techniques are very powerful, but one should keep in mind, that the time required to perform a building simulation is not at all negligible, and if one wants to explore designs where many options can be changed, the permutations make the number of possibilities immense. To explore and find solutions in such a large domain of options, the optimisation (even the most efficient ones) need to perform a large number of trials to recognise the best designs. This has been seen to be a technical barrier that makes the implementation of automatic or semi-automatic design difficult.

Optimisation (or automatic design) has largely been tested as a tool for building design by the research community, and is not commonly used by practicing engineers [5]. This situation may be changing with the introduction of optimisation tools for building design within "off the shelf" computational packages. There have been two key publications that give information about the status-quo of automatic design (optimisation) of buildings. These are the literature review of Evins [5], and the work of Attia et al. [6]. The latter includes a survey of building professionals that covers the use of optimisation for building design.

Attia et al. stated that most of the optimisation runs in the building design process are done during the early stages of the design (when there are still many degrees of freedom). The work of Attia et al. also showed that the two main technical obstacles for the deployment of optimisation in building design are:

1. Uncertainties in simulation model input, and
2. Long computational times.

The first point is still to be fully examined although there have been some preliminary works exploring the issue [7,8]; the second point is the main motivation for the development of the methodology described in this paper.

Some authors have used simplified building models to be able to run the optimisation in relatively short times [9,10]. Although the results 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 option is to create a response surface, i.e. an approximation model of the real objective function, and optimise that model (such as the work of [11]). Magnier and Haghighat used artificial neural networks (ANNs) to create an approximation of the building thermal model. An inconvenient of this is that, ANNs suffer of the *curse of dimensions* (explained below) 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.

On one hand, the idea of creating a surrogate model for the whole decision space looks less than ideal; on the other hand, using a basic building simulator may not provide the accuracy needed for a given problem.

We present in this paper, a different approach. We show a methodology that uses an evolutionary algorithm as the core of the optimisation, however as the algorithm evolves, the solutions are assessed with different assessment tools that require different computational times. With this we reduce the computational time of running an optimisation algorithm for building design to a fraction, and therefore, diminish the barrier for the deployment of these methods. It also represents a method more akin to human-based design, with only the key elements being considered early on, and more detailed aspects later.

The application of the methodology to a building design problem follows in Section 4, and the results are presented and described in Section 5, followed by conclusions (Section 6), acknowledgments and references.

2. Previous work

As discussed above, the use of complex assessment tools for the evaluation of potential solutions can render optimisation unfeasible. There is a standard procedure in engineering to tackle this problem developed by Barthelemy and Haftka [12]:

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 quite 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 cannot be done in many cases due to the complexity of the system to be analysed or the nature of the problem (for example, the search for 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 [13–19] in mechanical engineering, and [11] 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 [20].

One of the weaknesses of meta-models is that they suffer from the *curse of dimensionality* [19]. 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 $3^{20} = 3{,}486{,}784{,}401$ points, whereas, if the problem had 3 decision variables, one would need $3^3 = 27$ points. In the case of having a large number of decision variables (as in the first case) the use of a surrogate has to be combined with a Monte Carlo method that could benefit from a sampling method such as the Latin hypercube; but the order of growth of the number points that have to be analysed still stands. 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 meta-models.

Creating surrogate models for the whole decision space was considered by Booker et al. [13] as not ideal. In their report published by NASA, Booker et al. argued that 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 to construct a surrogate model before knowing the shape of the decision space and performing an optimisation run of the surrogate model that might