

OPTIMISING BUILDING RETROFITS AT THE STOCK LEVEL: THE DEVELOPMENT OF A METHODOLOGY

S. Amrith¹, P. Ruyssevelt¹, I. Korolija² and P. Fennell¹

¹UCL Energy Institute, University College London, London, United Kingdom

²UCL Institute for Environmental Design and Engineering, University College London, London, United Kingdom

ABSTRACT

Optimisation has been widely applied to individual building energy models, however there exists a gap in the literature around stock-level optimisation approaches. This paper describes the early development of a novel methodology which aims to allow multi-objective optimisation to be performed over a stock of buildings. This would be valuable to parties such as local authorities who require decision support when mapping retrofit pathways for their stock.

An early-stage workflow is described which is facilitated by a custom developed Python module and applied to an EnergyPlus stock model. An optimisation problem can be defined for the area, which influences how the module handles the parametric setup and optimisation. The setup involves specifying restrictions (e.g. listed buildings) and desires for the recommendations (e.g. if a standardised retrofit package should be applied to a certain dwelling type). According to this formulation, the module assists with parameter creation. The EnergyPlus file is automatically altered so that each parameter can be independent. Optimisation objectives and constraints can be derived from the model outputs. The module links EnergyPlus to the NSGA-II algorithm. Results are output in a format that allows easy segmentation, plotting and exporting.

The workflow is demonstrated on an area of housing stock to showcase how the methods are informed by an example problem definition. The paper concludes with a discussion around advantages, challenges and future development. The intention is to build on this work and include stages which handle simplification of the stock model, sensitivity analysis and calibration.

INTRODUCTION

The building stock is responsible for roughly 40% of the EU's energy consumption and 36% of greenhouse gas emissions (European Commission, 2020b). It is estimated that 97% of buildings in Europe are not energy efficient, and that 85-95% of the buildings which exist today will still be standing in 2050 (European Commission, 2020a; Buildings Performance Institute Europe, 2022). There is thus a clear need to improve the energy efficiency of the current stock, however the annual energy-renovation rate is estimated to be a mere 1% (European Commission Directorate-General for Energy, 2019).

There is thus a clear need to retrofit the building stock at a faster pace in order to meet the emission targets that have been set. However planning a retrofit, even with the aid of powerful building energy models, often proves extremely difficult in reality due to the complexities of conflicting criteria. Diakaki, Grigoroudis and Kolokotsa (2008) note that when planning a retrofit strategy that aims to maximise energy savings, the decision maker must also compensate energy, environmental, social and financial factors in order to satisfy all parties and criteria involved. Hashempour, Taherkhani and Mahdikhani (2020) also add that several renovation or retrofit techniques exist and are available, making the selection process even more complicated. Thus, energy retrofitting is often considered to be a multi-objective optimisation problem.

Optimisation has been widely applied at the building level, however this is not the case for the stock level. Carli *et al.* (2017) state that there appears to be a gap in the literature with regards the development of methodologies to optimise the retrofit of a portfolio of buildings with respect to a number of conflicting criteria. The paper also notes that there is a necessity to develop this area of research which has significant applications in real-world scenarios; for example in assisting organisations and public administrations to effectively allocate limited budget and resources among different buildings in the stock. Currently, these parties are only able to apply crude method such as scenario analysis when planning a large-scale retrofit, meaning it can be difficult to assess each case against the multiple criteria of interest.

This paper describes the early development of a novel methodology which aims to allow multi-objective optimisation to be applied over an EnergyPlus model of a stock of buildings. The setup of the problem is facilitated by a custom Python module which then tailors the setup of shared parameters based on the initial problem definition. The module passes the parameter and objective values between EnergyPlus and the NSGA-II algorithm. The optimisation results can be segmented based on the initial problem definition. In this way, the methodology seeks to address a number of the challenges which arise when setting up a stock-level optimisation problem. The methods described are applied to an area of the residential stock, with three problem scenarios demonstrated. The paper concludes with a discussion around advantages of the approach, the challenges of stock-level optimisation and further development of the methodology in the future.

METHODOLOGY

The early-stage methodology is described here. A full process from input data to results is presented. It is expected that this process will receive additional stages in future iterations based on the investigations that are described later in this paper.

Input data

Buildings data is fed into Simstock (Coffey *et al.*, 2015), a tool developed at UCL which automatically generates EnergyPlus model(s) for a stock of buildings. These models can then be set up for optimisation. An existing EnergyPlus model can also be chosen instead of generating one using Simstock. EnergyPlus is a whole building energy simulation software which can be used to model the heating, cooling, ventilation, lighting and plug loads in one or many buildings (Crawley *et al.*, 2001). Shading blocks can also be included in the model. The module interfaces with the EnergyPlus idf model file partially via the Eppy Python package (Philip, 2022).

Definition of the optimisation problem

As an illustrative example, we consider a local authority (LA) who must decide on the best strategy to retrofit their housing stock given certain criteria that must be fulfilled. In this formulation, the parameters represent the possible retrofits and the optimisation objectives/constraints represent criteria to minimise/maximise, for example CO₂ emissions/indoor air quality.

Considering this example of an LA's housing stock, there are many approaches that can be taken to apply optimisation on this. In a real-world scenario such as this one, identifying optimal individual retrofit strategies for each dwelling in their stock may not be of great use to the LA. Instead, it is possible that the LA would like to apply a standardised retrofit package to a pre-defined group of buildings in their stock, for example to all terraced houses. Whilst this may not be the most effective solution for each individual dwelling, it may be much more feasible from the perspective of coordinating and delivering such a large-scale retrofit programme. Creating standardised retrofit packages may also be desirable when treating a group of buildings with a similar construction, for example cavity walled dwellings. In addition, any conservation areas and listed buildings must either be excluded from the optimisation or given special attention.

Translating all of these factors into an optimisation problem inevitably influences both a) which buildings must be optimised together, and b) how the parameters are set up. A parameter refers to a type of

retrofit (e.g. window replacement) and its values refer to the possible choices (e.g. double glazing, triple glazing etc.). From a mathematical perspective, optimising at the individual building level and at the stock level is equivalent, except if:

1. The buildings have energy flows between them (e.g. a party wall).
2. The buildings are required to share parameters, in which case optimising at the building level is impossible (a shared parameter means that all buildings which share this parameter will be given the same parameter value).

In the absence of the above two conditions, optimising each building separately is sufficient, yet can be time consuming to set up. In most real-world cases, however, both of these conditions would be present. The optimisation framework presented in this paper aims to automate the process of setting up stock-level optimisation problems which take into account these factors.

In order to set up shared parameters, the attributes required are those which will be used to group the dwellings. There are then three options for how to segment the stock:

1. All dwellings have their own set of independent parameters (each dwelling will receive unique recommendations).
2. Dwelling groups are formed and parameters are shared within each group (each group will receive unique recommendations).
3. The entire stock shares a single set of parameters (all dwellings will receive the same recommendations).

It is also at this stage that any listed or heritage buildings can be specified. These can either form their own group, be treated individually, or be excluded from the optimisation altogether. Once the choice of segmentation has been established, the actual parameters can be defined for each group.

Parameters in EnergyPlus

EnergyPlus is capable of containing many buildings in one model file. However in the usual setup, a single set of model objects (e.g. materials, constructions) are shared amongst many of the buildings in the model. This poses a problem when setting up a stock-level optimisation problem since this configuration means it is usually not possible to set up independent parameters. Without prior setup, parameters which are set up on a given model object (e.g. a material's thickness) will be shared amongst all buildings which use that object.

To set up a parameter, the following must be specified: the model object, a list of values that the parameter can take, and the group of dwellings which the parameter applies to. The module duplicates any model objects which have been specified as parameters to ensure that they can be fully independent. It also duplicates any other model objects which rely on these (e.g. constructions which contain the selected material). It then updates any references to these objects.

If we take a material's thickness as an example parameter, the following model objects will be edited:

- The material object itself – duplicated and renamed to reference the specified dwelling group to which it belongs
- The construction object which contains the original material object – duplicated and renamed to reference the specified dwelling group to which it belongs
- Any building surface belonging to the specified dwelling group which uses this construction – the field which refers to the construction name is updated to the new name

By following this procedure, we leave the original material and construction objects untouched. This means any objects which refer to these will not be affected.

Link to the NSGA-II optimisation algorithm

The module links EnergyPlus to the NSGA-II algorithm which is implemented in the Platypus Python library (Deb *et al.*, 2002; Hadka, 2022). The NSGA-II is a widely used genetic algorithm (GA); Huang and Niu (2016) state that “the GA and its modifications are considered to be the best choices for solving building design optimisation problems”. GAs take inspiration from biological processes to iteratively generate a group of solutions which improve with each ‘generation’. The ‘fitness’ of the current

generation is assessed in each iteration, which is a function of the chosen optimisation objectives. Since GAs introduce some stochasticity in the form of genetic ‘mutations’, they are effective when working with objectives containing many local minima. They are also favoured for their ability to handle non-linear problems with discontinuities (Machairas, Tsangrassoulis and Axarli, 2014).

Optimisation objectives & constraints

The optimisation objectives and constraints can be derived from any of the EnergyPlus simulation outputs. This can either be the raw output value, or a manipulation of this via a pre-defined mathematical function. Each time a generation is completed, the resulting objective values are fed to the NSGA-II algorithm which in turn determines the parameter values for the next generation. These are inserted into the EnergyPlus model and then the simulations are executed. This process is repeated until the target number of generations has been reached. The optimisation process is able to make use of multiple CPU cores and therefore can run the EnergyPlus simulations in parallel.

Results and visualisation

A results archive is created in csv format which records the parameter values for each simulation along with the associated objective values. Results are appended to the archive after each generation completes; in the case of a crash, results for all generations up to the point of failure can still be retrieved. The module can restructure a given result of interest to list each individual dwelling against the parameter values it received. Restructuring the results in this way is useful since this information is not easy to determine when parameters are shared amongst a number of dwellings. This can then be visualised in external software such as QGIS – an example is provided at the end of the case study section.

CASE STUDY

In this section, the methodology is demonstrated on a simple model of a residential area of the stock. The purpose of this is to demonstrate how the module assists with the setup of an optimisation study, given different input problems.

The area contains a mix of dwelling types: detached, semi-detached and terraced. The building footprints are shown in Figure 1; 23 of the dwellings (in yellow) were modelled and the rest (in blue) were included as shading blocks. The polygons and building heights were sourced from Ordnance Survey MasterMap (OS MasterMap, 2019). This data was fed into Simstock which generated the EnergyPlus stock model. According to the building heights, each modelled dwelling was horizontally sliced into a number of thermal zones to achieve a minimum floor-to-floor height of 3m. The resulting model contained 49 thermal zones. The constructions were assumed to be a typical cavity wall with a base case insulation thickness of 5cm. Typical occupancy schedules, setpoints and loads were sourced from Fennell, Ruyssevelt and Korolija (2020). The electricity consumption and heating loads were roughly calibrated against Ofgem’s Typical Domestic Consumption Values (TDCVs) (Ofgem, 2020).

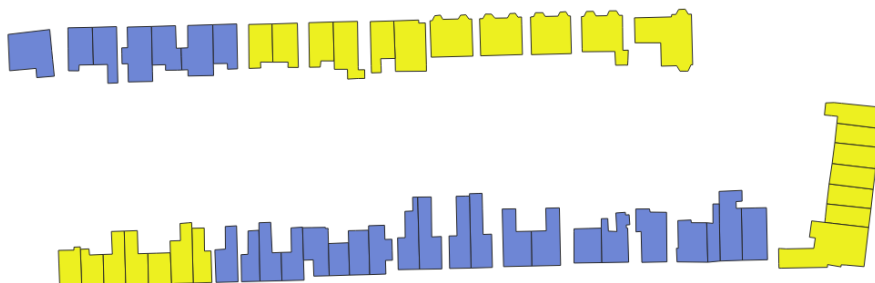


Figure 1: Building footprints of the selected area. Contains a mix of dwelling types. Dwellings in yellow (23) were modelled whilst those in blue were included as shading geometries.

Retrofits

The retrofits chosen for this study are wall and roof insulation thicknesses and glazing type; these can be found in Table 1 with their allowed values. Details on the constructions of the window types can be found in Table 2.

Retrofit	Values
Thickness of wool quilt for wall	{0.04, 0.06, 0.08, 0.10, 0.12, 0.14} metres
Thickness of wool quilt for roof	{0.04, 0.06, 0.08, 0.10, 0.12, 0.14} metres
Glazing type	{Single, Double, Double xenon}

Table 1: The retrofits included in the study with their possible values.

Window type	Construction (out -> in)	Modelled U-value [W/(m ² ·K)]
Single glazing	4mm clear glass	5.871
Double glazing	4mm clear glass 12mm air gap 4mm clear glass	2.725
Double glazing xenon	4mm clear glass 16mm xenon gap 4mm low-e glass	1.782

Table 2: Constructions for each glazing type with their modelled U-values.

Parameters and groupings

Each row in Table 1 represents a parameter template and its allowed values. The next step is to determine which dwellings must share parameters, after which the actual parameters can be created. To demonstrate the abilities of the methodology, three scenarios are considered:

1. Scenario 1 (S1): every individual dwelling can receive their own retrofit configuration (no shared parameters)
2. Scenario 2 (S2): a standardised retrofit package must be given to each dwelling type (parameters should be shared by dwellings of the same type)
3. Scenario 3 (S3): a standardised retrofit package must be given to all dwellings (parameters should be shared by all dwellings)

To set up the optimisation, the module firstly expects information about how to create the dwelling groups over which the parameters will be applied. Scenario S1 is a simple case since no parameters are shared, so no further information was required as input (in actuality, a group is formed for every dwelling, meaning the number of dwelling groups = number of dwellings). The module then set up an individual set of parameters for each dwelling.

For scenario S2, information needed to be specified regarding the dwelling type of each dwelling. Given this information, the module created three groups (for detached, semi-detached and terraced dwellings) and automatically set up shared parameters for each of these groups.

For scenario S3, similarly to S1, no extra information was required for this case. A single group was formed containing all dwellings.

The resulting number of dwelling groups and parameters are shown in Table 3.

Note that for simplicity, only one set of retrofits are considered for this demonstration and these were applied to every dwelling group. However, this is not a requirement; specific sets of retrofits can be assigned to specific dwelling groups if required – refer to the section “beyond this simple example” for a discussion around this.

	S1	S2	S3
No. of dwelling groups	23 (one per dwelling)	3 (one per dwelling type)	1 (with all dwellings)
No. of parameters	69 (= 3 x 23)	9 (= 3 x 3)	3 (=3 x 1)

Table 3: The number of dwelling groups and parameters for each scenario. The number of parameters in each scenario is calculated by multiplying the number of retrofits considered (3 in all cases) by the number of dwelling groups.

Once the parameters were defined, the module automatically edited the EnergyPlus model files to facilitate them as follows. It is the two insulation material objects, wall construction, roof construction and window construction model objects that required duplication. The module duplicated these to match the number of dwelling groups in each case, and named the new objects appropriately. Finally, each zone group's building surface objects were edited to reference the group's newly created material and construction objects. For S3, no edits to the model file were required.

Optimisation objectives

For the optimisation objectives, heating and cooling loads were chosen for this demonstration as they are opposing objectives meaning they should lead to a Pareto front. The location of the study is the UK which is a heating-dominated climate; improving the building fabric can help to reduce the heating loads but can also increase the risk of overheating in the summer since it is more difficult for the heat to escape whilst the solar and internal gains remain high. These objectives were automatically retrieved directly from the EnergyPlus simulation results for each thermal zone and summed to arrive at the total loads for the whole area.

Simulation

For each scenario, the GA was run for 120 generations, each with a population size of 10. This was run on UCL's high performance computing cluster, using 16 cores of an Intel(R) Xeon(R) Gold 6240 CPU @ 2.60GHz – each scenario took around 9 hours to complete.

RESULTS AND DISCUSSION

Running the GA for 120 generations proved sufficient to achieve good convergence in scenarios S2 and S3, whilst S1 failed to explore the full space due to the very high number of parameters. Plots of showing the optimisation objectives for each scenario are shown in Figure 2. A Pareto front can be seen in all three scenarios, however they differ in shape and clarity. A Pareto front consists of Pareto-optimal solutions; this marks the boundary beyond which none of the objective values can be improved upon without sacrificing the other objective. S3 exhibits a very sharp and defined Pareto front which is likely to be due to the very small number of parameters; the exploration space is limited and there are in fact only 108 possible permutations meaning it is highly likely that the entire space was explored.

The plots of the objective values do not provide any information about the retrofit configurations themselves except for their associated heating and cooling demands. Since it is impossible to analyse every solution, a representative point is picked from each scenario to be broken down and analysed. It was aimed for these points to be as close as possible in terms of their objective values. The points selected: S1 – heating = 125,821 kWh, cooling = 69,298 kWh; S2 – heating = 128,566 kWh, cooling = 67,441 kWh; S3 – heating = 126,548 kWh, cooling = 67,221 kWh. After selecting the points of interest, the module functions were used to restructure the results in order to match the parameter values to the associated dwellings. This was then visualised in QGIS and the plots are discussed below.

The results for each retrofit are plotted separately in Figure 3. To illustrate how the results differ in each scenario, three maps of the dwellings are shown within each box; one for each scenario. Viewing the results in this way highlights the effect that shared parameters have on the optimisation. Since there were no shared parameters in S1, each dwelling has received its own retrofit configuration; this can be seen in the left-most maps. This kind of result takes into account the individual features of each dwelling, for example the floor area, constructions, façade orientation and so on. In S2 (middle maps), each dwelling type has received its own retrofit configuration. This kind of result has the effect of “averaging out” the individual features within a given dwelling group. It will also give more weighting to the dwellings which achieve a larger drop in heating/cooling loads since these provide a larger contribution to the overall minimisation of the objective values. In S3 (right-most maps), all dwellings have received exactly the same retrofit configuration. The same principles apply here as in S2 except now this includes all dwellings. Since this was a demonstration of setting up an optimisation problem, the retrofit

recommendations themselves will not be discussed here (see future work section).

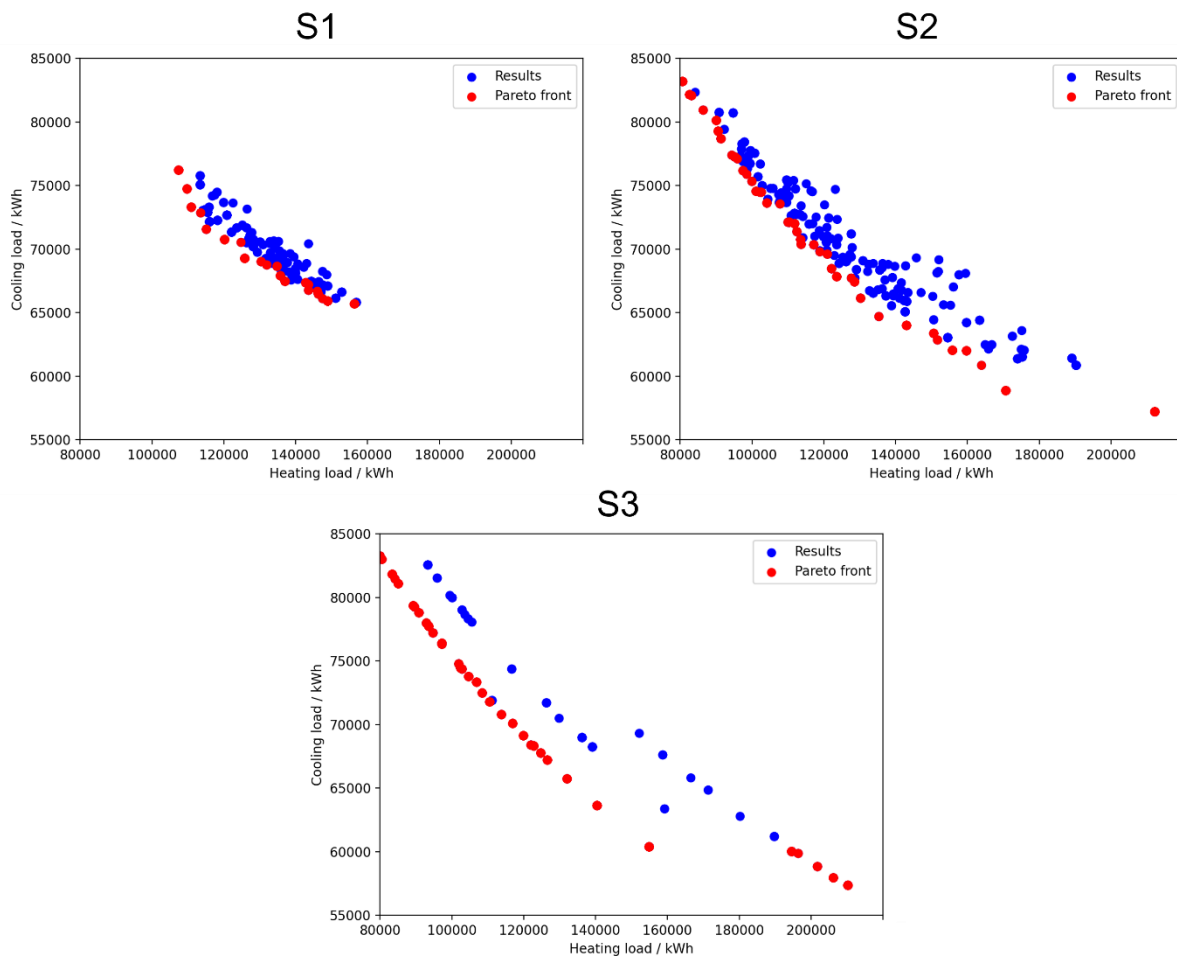


Figure 2: Optimisation results showing cooling load against heating load for each scenario. All plots are to the same scale. The Pareto-optimal solutions in each scenario are denoted in red.

Beyond this simple example

This example study was intended to demonstrate the formation of dwelling groups, creation of associated parameters, and the retrieval of group results. However, more complex setups are also possible.

Firstly, any provided attribute can be used to divide the stock into dwelling groups. In the demonstration dwelling type was chosen, but other examples include dwelling age, floor area, current construction and so on.

Secondly, it is not necessary that the same set of retrofits are considered for each group. Each group can receive its own set of retrofits if required. This is likely to be necessary in a real-world scenario as some retrofits may not be appropriate for certain dwellings. For example, separate groups could be formed for cavity-walled and solid-walled dwellings, and the insulation parameters created for these groups could differ in the type of insulation considered.

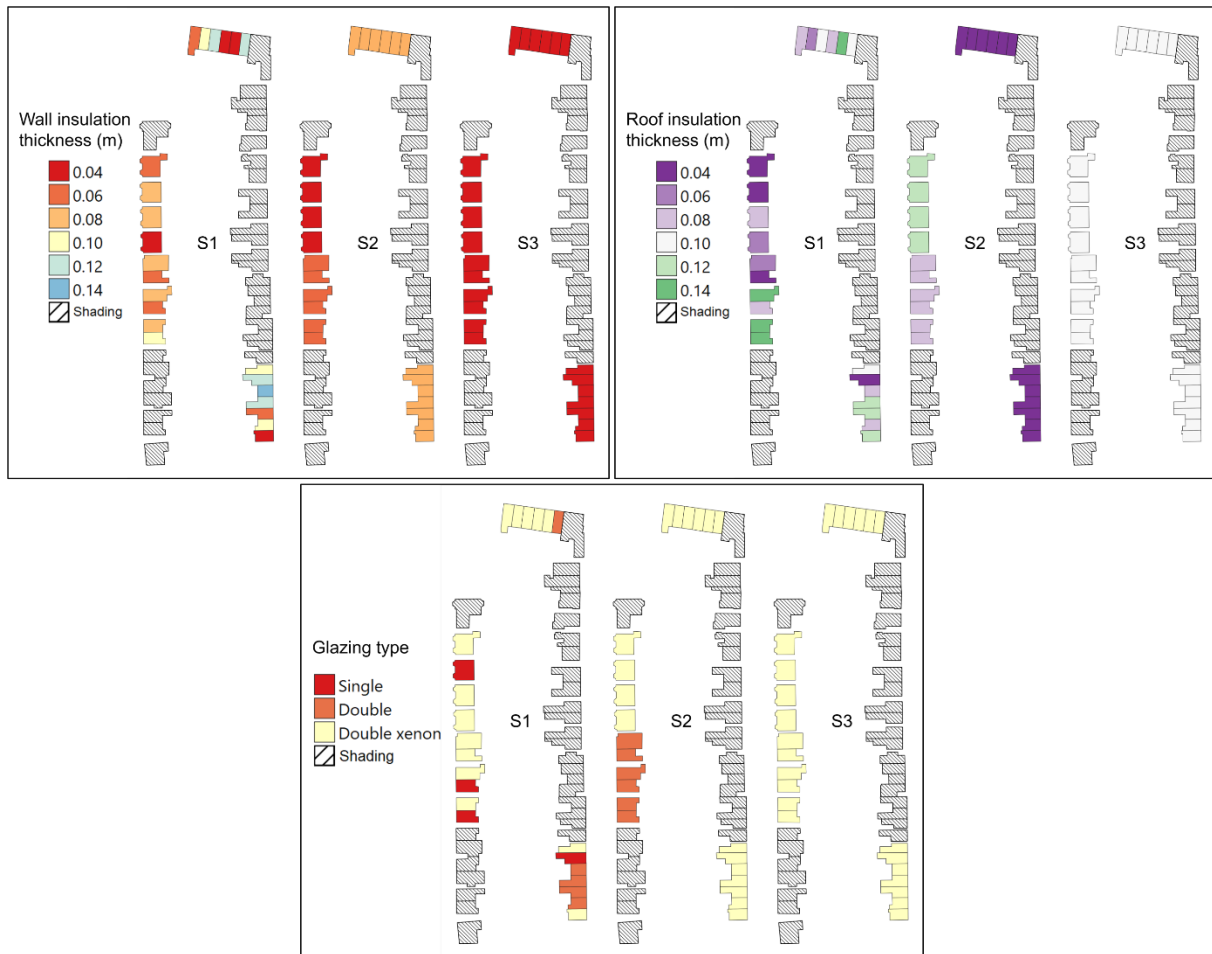


Figure 3: Each box contains the results for a given retrofit, and a map of the dwellings is shown for each scenario within this. The parameter value result for each dwelling is indicated by the colour of the dwelling.

Reducing the number of permutations

The total number of possible parametric permutations in S1 is extremely high, and it is clear that this negatively affected the optimisation results. As previously mentioned, only buildings which either share parameters or have some energy flows between them must be optimised together. In S1, there are many buildings which are not physically connected and therefore do not share any energy flows. In addition, none of the buildings share parameters. As a result, the area could be subdivided into “built islands” (i.e. collections of buildings which are touching), and a separate optimisation could be performed on each of these units. For S1, this would reduce the total number of possible parametric permutations by an order of 10^{32} . This is an extremely significant reduction and therefore would give the optimisation algorithm a much better chance of exploring the full extent of the parameter space. This is already possible using the module since Simstock is capable of subdividing the area into built islands before creating the EnergyPlus models. The main question that remains here is how to re-aggregate the results after performing the separate optimisations. This is a future objective for this project and is discussed further in the future work section.

Further challenges at the stock-level

In addition to the aforementioned challenges faced when implementing stock-level optimisation, some more are discussed here. The work that will be done to try to address these challenges is described in the future work section.

One of the major challenges around stock-level optimisation is the computational time and power required to simulate such a large model hundreds or thousands of times. Applying an algorithm still drastically reduces the number of simulations required when compared to the brute force method,

however without adjustments to the model, this is still likely to be insufficient to allow the optimisation to converge in a reasonable time-frame on modern hardware.

Since there is a gap in the literature with regards to stock-level optimisation, there is also no established format with which to present the results of such a study. The Pareto front is composed of solutions of equal importance in terms of the chosen objectives, so it is insufficient to focus on just one of these points. On the other hand, it would be impossible to perform an in-depth analysis of every single Pareto-optimal result. These considerations will be key in making the results as valuable as possible to parties who would benefit from stock optimisation methodologies.

Future work

As mentioned, the module is able to identify built islands of buildings which have energy transfer via party walls. It is intended to modify this process to also take into account formed dwelling groups which share parameters. Since these two considerations are what determines which buildings must be optimised together, the smallest possible optimisation units can ultimately be determined from this information. Ensuring that optimisation is performed on the smallest possible units ensures that the maximum number of parametric permutations is minimised.

There are several ways of reducing the stock model simulation time, with the most obvious being via model simplification. In the next stages of the project, the impact of a selection of simplification techniques will be assessed on the simulation time and model accuracy.

The limits of the parameter space will be investigated before the NSGA-II algorithm starts failing to explore the full space. The effect of tuning the hyperparameters (i.e. crossover and mutation) will also be assessed with regards to convergence. Alternative optimisation algorithms will be tested to see if their performance can improve over that of the NSGA-II when the parameter space is especially large. It is a known problem that genetic algorithms can get “stuck” in a small region of parameter space without exploring the full extent of it, as was the case in scenario S1. This risks the set of Pareto-optimal solutions being very far from the “true” Pareto front (i.e. the front that would be achieved if every parametric permutation were simulated). However, there are also a variety of approaches to counter this, which will be explored in future work (Pandey, Chaudhary and Mehrotra, 2014).

Once the methodology has been developed further, it will be applied to various test cases. To this end, it is intended to apply more thorough calibration on the stock model. Uncertainty and sensitivity analysis will also be applied to quantify and potentially reduce uncertainties in the model outputs.

CONCLUSION

A gap in the literature has been noted with regards to stock-level optimisation. There is a clear need for further research in this area as it would provide local authorities with much needed holistic decision support as they try to decide on the best strategies to decarbonise their housing stocks.

This paper has presented a novel early-stage approach to setting up stock-level optimisation. A custom developed Python module automates the setup of an optimisation study on an EnergyPlus model of a stock of buildings. Based on a specified input problem, the module determines how to segment the stock and which buildings require shared parameters. This allows buildings to be grouped and standardised retrofit packages to be determined for each group if desired. This formulation then informs the automatic parameter creation, setup of the EnergyPlus model, and results format. The approach was applied to three scenarios in order to demonstrate the workflow, and to show how each problem definition affected the results of the optimisation. The methodology is in its early stages and will be developed further in future iterations.

This paper has also highlighted a number of the challenges around stock-level optimisation. There is work to be done around minimising the dimensionality of the parameter space, ensuring the optimisation

algorithm is used in the most appropriate way, guaranteeing convergence, model simplification and exploring different approaches to present the results.

ACKNOWLEDGMENT

This research was made possible by support from the EPSRC Centre for Doctoral Training in Energy Resilience and the Built Environment (ERBE), grant number EP/S021671/1 and with in-kind and/or financial support from Bentley Systems.

REFERENCES

- Buildings Performance Institute Europe (2022) *Report on the evolution of the European regulatory framework for buildings efficiency*.
- Carli, R. *et al.* (2017) 'A Decision Making Technique to Optimize a Buildings' Stock Energy Efficiency', *IEEE Transactions on Systems, Man, and Cybernetics: Systems*. Institute of Electrical and Electronics Engineers Inc., 47(5), pp. 794–807. doi: 10.1109/TSMC.2016.2521836.
- Coffey, B. *et al.* (2015) 'An Epidemiological Approach To Simulation-based Analysis Of Large Building Stocks', in *14th Conference of International Building Performance Simulation Association, Hyderabad, India*.
- Crawley, D. B. *et al.* (2001) 'EnergyPlus: A new-generation building energy simulation program', *Energy and Buildings*, 33(4), pp. 319–331. doi: 10.1016/S0378-7788(00)00114-6.
- Deb, K. *et al.* (2002) 'A fast and elitist multi-objective genetic algorithm: NSGAII', 6(2), pp. 182–197.
- Diakaki, C., Grigoroudis, E. and Kolokotsa, D. (2008) 'Towards a multi-objective optimization approach for improving energy efficiency in buildings', *Energy and Buildings*, 40(9), pp. 1747–1754. doi: 10.1016/j.enbuild.2008.03.002.
- European Commission (2020a) 'A Renovation Wave for Europe - greening our buildings, creating jobs, improving lives', *European Commission*, COM(2020), pp. 1–26.
- European Commission (2020b) *In focus: Energy efficiency in buildings*. Available at: https://ec.europa.eu/info/news/focus-energy-efficiency-buildings-2020-lut-17_en (Accessed: 22 August 2022).
- European Commission Directorate-General for Energy (2019) *Comprehensive study of building energy renovation activities and the uptake of nearly zero-energy buildings in the EU*.
- Fennell, P. J., Ruyssevelt, P. and Korolija, I. (2020) 'Balancing accuracy and computation burden - an evaluation of different sensitivity analysis methods for urban scale building energy models', in *uSIM2020 - Building to Buildings: Urban and Community Energy Modelling*.
- Hadka, D. (2022) *Platypus: A Free and Open Source Python Library for Multiobjective Optimization*. Available at: <https://github.com/Project-Platypus/Platypus> (Accessed: 25 August 2022).
- Hashempour, N., Taherkhani, R. and Mahdikhani, M. (2020) 'Energy performance optimization of existing buildings: A literature review', *Sustainable Cities and Society*. Elsevier Ltd, 54, p. 101967. doi: 10.1016/j.scs.2019.101967.
- Huang, Y. and Niu, J. L. (2016) 'Optimal building envelope design based on simulated performance: History, current status and new potentials', *Energy and Buildings*. Elsevier Ltd, 117, pp. 387–398. doi: 10.1016/j.enbuild.2015.09.025.
- Li, T. *et al.* (2017) 'Genetic Algorithm for Building Optimization: State-of-the-Art Survey', in *Proceedings of the 9th International Conference on Machine Learning and Computing - ICMLC 2017*. Singapore: ACM Press, pp. 205–210. doi: 10.1145/3055635.3056591.
- Machairas, V., Tsangrassoulis, A. and Axarli, K. (2014) 'Algorithms for optimization of building design: A review', *Renewable and Sustainable Energy Reviews*. Elsevier Ltd, 31, pp. 101–112. doi: 10.1016/j.rser.2013.11.036.
- Ofgem (2020) *Typical Domestic Consumption Values*. Available at: <https://www.ofgem.gov.uk/gas/retail-market/monitoring-data-and-statistics/typical-domestic-consumption-values> (Accessed: 12 August 2020).

OS MasterMap® Topography Layer [GML2 geospatial data], Scale 1:1250, Tiles: GB, Updated: 28 November 2019, Ordnance Survey (GB), Using: EDINA Digimap Ordnance Survey Service, <<https://digimap.edina.ac.uk>>, Downloaded: 2020-07-23 11:43:50.135

OS MasterMap Building Height Attribute [CSV geospatial data], Scale 1:2500, Updated: 30 August 2019, Ordnance Survey (GB), Using: EDINA Digimap Ordnance Survey Service, <<https://digimap.edina.ac.uk>>, Downloaded: 2020-07-23 11:43:50.135

Pandey, H. M., Chaudhary, A. and Mehrotra, D. (2014) 'A comparative review of approaches to prevent premature convergence in GA', *Applied Soft Computing*. Elsevier, 24, pp. 1047–1077. doi: 10.1016/J.ASOC.2014.08.025.

Philip, S. (2022) *eppy*. Available at: <https://pypi.org/project/eppy/> (Accessed: 25 August 2022).