THE USE OF OPTIMISATION IN THE CALIBRATION OF BUILDING SIMULATION MODELS

Shadi Basurra, Ljubomir Jankovic, Halla Huws

Zero Carbon Lab
Birmingham School of Architecture
Birmingham City University, Birmingham, UK
Shadi.Basurra@bcu.ac.uk; Lubo.Jankovic@bcu.ac.uk; Halla.Huws@mail.bcu.ac.uk

ABSTRACT

In this paper, a model-assisted calibration is proposed in order to reduce performance gap of simulation models, being the estimated differences between the predicted performance of the simulation model and the actual performance of the building. We use the nearly unbiased k Nearest Neighbour (KNN) algorithm to classify the solution that exhibits the minimum performance gap based on a set of reference points. Density avoidance algorithm is used to further refine the solutions by finding regions in the space of input factors for which the model output is either maximum or minimum to meet optimum criterion, hence fine tuning the model to establish one-to-one relationship between the simulated and actual performance.

INTRODUCTION

Model calibration is an important process to ensure that the building thermal performance is represented accurately, in relation to the architecture, mechanical systems, internal gains, and the building fabric. During the calibration the input values of parameters are varied and tested until the difference between the real performance of the building and the model performance are minimised, reducing the performance gap.

Even for an experienced modeller, the calibration process could be labour intensive and time consuming. The aim of this paper is to examine the use of KNN to calibrate the model. The Birmingham Zero Carbon House is used as experimental evidence base for this investigation. It is a retrofitted Victorian house that has achieved a carbon negative performance, and it has been under detailed instrumental monitoring over the past four years. The data collected from the monitoring are utilised in the calibration process.

Energy retrofit of existing buildings helps to increase the building energy efficiency through a mix of design solutions with a reasonable cost and payback time frame. Building simulation tools have been used to explore possible alternatives to achieve better energy performance with a shorter payback period.

Allocation of risks requires uncertainty quantification of projected cost effectiveness of design options for a given retrofit project. Hence, reducing the performance gap between the actual building and the model encourages building owners to invest in retrofit with confidence, and facilitates realistic ambitions towards energy saving and payback. This is vital for retrofit schemes such as recently discontinued Green Deal (Energy Saving Trust, 2014) that involved retrofits of existing buildings through performance based contract.

Using building simulation is somewhat easier for new built projects, were building properties and parameters are given using the engineering design specification. However, designing a model to represent an actual building is not trivial. It is difficult know how the building's internal/external components perform, and whether or not the building materials and technology used have the same theoretical properties after the building is built. Hence, simulation models of existing buildings need some form of verification or calibration to minimise their performance gap, which will help to build knowledge base for new designs.

The paper will demonstrate that calibration via KNN and density avoidance algorithm is achievable, and that it can be implemented on existing data generated from standard simulation tools such as DesignBuilder and IES. This process ensures that all options are considered and tested, which leads to better and more accurate models, whilst also reducing the time and effort required to calibrate a model.

Related work

Calibration is used to identify the cause of poor model performance in comparison with actual data obtained from building surveys, expert knowledge, industry standards, sensor information etc. Real information can be used to explore possible solutions via refinement or 'justified' tweaking of the model input. However, this can be complicated due to the issues identified by Clarke (1993); 1- the model range is constrained due to the lack of experimental evidence; 2- hidden assumptions performed by various software implementations; 3- energy models can be complex with many interactions; 4- uncertainties with basic properties of existing building. These issues have been dealt with in various ways, for example using manual evidence based calibration (Raftery, 2011). That research

describes a systematic evidence-based methodology for the calibration of building simulation models. It aims to improve the accuracy of the final model by allowing only verifiable information about the model to be used. In addition, to improving the reproducibility, all previous calibrated modes are stored in version control repository as supporting evidence to understand the assumptions made and minimise tuning input parameters. However, this evidence based process can be lengthy, and requires detailed information about the existing building which may not be available for old and historical buildings. Furthermore, entire calibration process should be preferably automated to ensure efficiency and consistency (Tahmasebi, 2012).

Zoning strategy is another example of model calibration that has been used in studies such as by Yiqun et al, (2007). This is carried out by combining thermal zones in the building, using up to five zones per occupied floor. According to (Raftery, 2011) zoning technique can be appropriate for complex floor plans, but it reduces the model accuracy by simplifying the model. For example, the model cannot accurately represent situations where opposing cooling and heating loads in one floor, and it does not allow accurate representation of different occupancy profiles and internal loads. The research (Heo, 2012) presented a probabilistic methodology based on Baysian calibration of normative energy models to quantify uncertainties in the model that are translated to quantify risks associated with underperformance of retrofit designs. However, the method depends heavily on experts judgment in the choice of calibration parameters and their distribution. Moreover, the current method is based on the complicated statistical formulation of Kennedy and O'Hagan's framework, which can cause computation burdens for large-scale analysis, hence, this method is limited to the cases in which the source of measured data is at one building level. The use of optimisation–aided model calibration to select best results in relation to cost/fitness function is described in (Taheri et al, 2013). Calibration was used to minimise the error rate between the measured air temperature and simulated temperature. The main issue in that work is that parameters and their range values were selected in advance based on the basis of author's experience, which may lead to inconsistent results.

METHODOLOGY

The k-nearest neighbours' algorithm (KNN) (Alt, 2001) is a widely used technique for clustering and classification of data in data mining, and pattern recognition. It is a basic approach to find the most similar k number of points as nearest neighbour to a given reference point on a solution space. In this study, we suggest KNN methods as an alternative to solve the problem of simulation model calibration in order to improve the correspondence between actual and monitored values towards a one-to-one line with an intercept of zero in the ideal case. In spite of their simplicity, KNN methods are among the best performers in a large number of classification problems. This is because KNN is non-parametric which means the algorithm works without presumption of the primary data distribution. Thus, the algorithm requires no training phase before being used on a solution space. This is useful for calibration of simulation models since the real monitored data do not usually obey the typical theoretical assumptions made in the simulation model. Moreover, the algorithm is fast to perform, despite the fact that KNN bases its decision after calculating the entire solution space.

KNN is used for classification and regression. Classification is performed using the instance-based classifier by locating the nearest neighbour in the instance space and labelling the unknown instance with the same class label as that of the located classified (known) neighbour. One of the classification rule for KNN is to find the nearest neighbour using distance and majority voting which is calculated as a weighted KNN where each point has a weight that is typically calculated using the inverse distance and majority voting, which allows those neighbours where $k > 1$ to decide the outcome of the class labelling.

KNN assumes that the data points are in a metric space, hence the data can be multidimensional vectors. Each of the training data consists of a sect of vectors and class label associated with each vector. Positive (+) or Negative (–) are the easiest form of classes, however, KNN can effectively work well with various numbers of classes.

The process starts by measuring distances between the query points to the rest of the solution points. One of the most popular choices to measure the distances is to use the Euclidean function. Given $x = (x_1, \dots, x_n)$ and $y = (y_1, \dots, y_n)$, the distance is calculated as

$$d_E(x, y) = \sqrt{\sum_{n=1}^{N}(x_n - y_n)^2}. \quad (1)$$

KNN regression is related to predict the outcome of a dependent variable given a set of independent variables. This is useful since it enables the prediction of the regions in which future candidate solutions will be populated.

The algorithm function

k is the number of nearest neighbours in the solution space $S := (p_1, \dots, p_n)$ where p_n is the solution sample in the form $p_1 = (x_i, c_i)$, where x_i solution entry with all parameter values of the point p_i. c_i is the class that p_i belongs to (see Figure 1).

- Start:
  - For each $\boldsymbol{p'} = (\boldsymbol{x'}, \boldsymbol{c'})$
  - Calculate the distanced $\boldsymbol{d}(\boldsymbol{x'}, \boldsymbol{x_i})$ between $\boldsymbol{p'}$ and all $\boldsymbol{p_i}$ belonging to $\boldsymbol{S}$
  - Re-organise all $\boldsymbol{p_i}$ in accordance to their distance
  - Select the first $\boldsymbol{k}$ points from the sorted list, those are the $\boldsymbol{k}$ closest training samples to $\boldsymbol{p'}$
  - Allocate a class to $\boldsymbol{p'}$ based on majority vote: $\boldsymbol{c'} = \boldsymbol{argmax} \sum(\boldsymbol{x_i}, \boldsymbol{c_i})$ belonging to $\boldsymbol{S}$, $\boldsymbol{I}(\boldsymbol{y} = \boldsymbol{c_i})$. For $\boldsymbol{p_i}$, where $\boldsymbol{i}$ = 1,2, .. number of pints in $\boldsymbol{c_i}$
- End:

Figure 1: KNN algorithm steps

The selection of k is critical. This is because a small value of k means that the results will be increasingly influenced by noise. However, a large value of k can make it computationally expensive, but also defeats the concept behind the KNN that solution 'points' that are near are likely to have similar density factor. One simple approach suggested by Richard et al. (2000) is to set k as $k = \sqrt{n}$ where n is the total number of points in the solution space.

Calibration assistant

KNN works well when the solution points are scattered around the reference point while covering all regions in the graph. Each neighbour encapsulates all values for the parameters used during the simulation. The algorithm then calculates the maximum and minimum values for each parameter from all discovered neighbour solutions. Using the minimum and maximum values, we can break the range further into smaller steps to be used again as inputs variables for the model parameters in the simulation. This iterative process helps to bring the simulation solutions closer or even intercept with the reference point retrieved from the monitored data. This will also help identify the least sensitive data, with the same maximum and minimum values.

Density estimation for KNN

As discussed above, KNN ideally identify neighbour solutions scattered evenly in the solution space while covering various regions on the graph. However, if the reference point is adjacent to highly densely populated area of solutions, the algorithm only selects the solutions from the dense area, especially if the number of nodes located in that area exceeds the calculated k neighbours. This will worsen if all detected nearest neighbours hold the same solution values, which exist on the same location in the graph.

Various extensions have been performed to the KNN algorithm to consider density. Although classification is the primary application of KNN, density estimation can also be used in KNN. Density estimation is a non-parametric method for constructing a density estimate of results. This is very similar to Parzen-window which is essentially a data-interpolation technique (Richard et al, 2000). For example, to estimate density at a point *x*, by placing a circle centered at *x* and keep increasing its size until k neighbours are captured, the density estimation uses the following formula:

$$p(x) = \frac{k/n}{a} \qquad (2)$$

In the formula above, n is the total number of solutions, and a is the area of the circle. The numerator is constant and the density is influenced by its value. For example, if density at a point *x* is high, KNN finds the k points near *x*, and these points turn to be close to *x*. This shows that the area of the circle is small, and the resultant density is quite high. However, if the density around the point *x* is low, the area of the hypercube that is needed to encompass k nearest neighbours will become large, since the density ratio is low.

We use similar technique to KNN, but differ in the sense that instead of using density to classify neighbours, we use density calculation to select a fewer neighbours located in high dense areas, hence, the algorithm looks for other solutions that cover all areas as long as they are within reasonable distance from the reference point.

Let us consider the simplified example in Figure 2a. Given the total number of solutions in the solution space is 9, k becomes 3 according to the selection rule of k explained above. When KNN is implemented, based on the weighted distance only, the solutions *B*, *C* and *D* are selected as the best neighbours to the reference point shown as a cross in the graph. Although these solutions are close in terms of distance, they all exist in one region covering a smaller range of parameters. Hence, the maximum and minimum ranges will be small when trying to fine-tune the model to the reference point. However, when KNN is implemented with density avoidance enabled, *A*, *E* and *C* solutions are selected instead since they are positioned at a sensible distance to the reference point. *B* and *D* will be ignored, as *C* will provide the range that covers the highly dense area formed by the solutions *B*, *C* and *D* (see Figure 2b).

Density avoidance for KNN

As mentioned above, KNN density estimation is used for regression and classification; however, we use density detection concept to avoid the high-density regions dilemma. We are proposing a density avoidance algorithm which has been tested against various cases for this purpose. Our proposed density avoidance algorithm is explained below.

Starting from a solution close by from the reference point, each solution will form a circular region with a constant radius R to capture all surrounding nodes in the solution space. For example, let us consider a solution *X* of *N* solutions in the graph. *X* will perform the density estimation and calculate the density using Equation (2). Also see Figure 3 for Pseudocode describing the steps of the density avoidance algorithm.

If density is above a threshold, the node closest to *X* (not the reference point), will be tagged as idle. The whole process repeats again, and *X* becomes the second closest node to the reference point. In subsequent iterations, idle nodes are not selected to perform the density calculation, and will not be considered in the density check if they fall within the range within a circle area of another valid 'non idle' node. Following these rules, all nodes in the solution space will be tagged as either idle or valid.

Then we implement the KNN algorithm that selects the closest k neighbours, but also selects only those valid solutions. This was successfully implemented, and is shown in Figure 9.

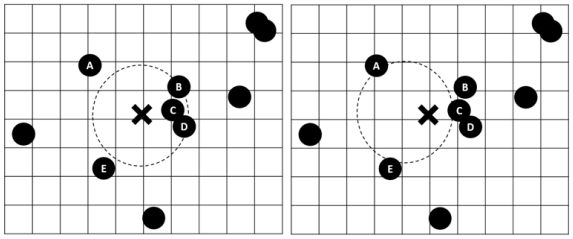

Figure 2:KNN in operation without (a) / with (b) density avoidance technique

- PROGRAM DensityExclusionAlgorithm:
  - Using **KNN**, CALCULATE distances to all Nsolutions from Reference Point.
  - Store the N neighbours with their distances in a list L
  - Sort list L in a ascendant order putting least distant solutions at the top of L.
  - LOOP through L starting from the top, and select X solution
    - X Identify nearby none-idle neighbours using a predefined radius R, and store them in a new list L_2.
    - X calculates density L_2
    - If $(Density > Threshold)$
      - THEN from L_2, set "IDLE" to Neighbour closest to X
      - ELSE DO NOTHING;
  - ENDLOOP
  - CALCULATE neighbours of Reference Point with K number of neighbours (See Figure 2).
- End

Figure 3: Pseudocode describing the steps of the density avoidance algorithm.

Zero Carbon House model

For the purpose of this research, we have selected the Birmingham Zero Carbon house (Christophers, 2014), which was originally built in 1840, and has been retrofitted recently to achieve zero carbon performance. It has been selected based on availability of information for the energy model, good quality observations and easy access to the site for operational adjustments. There have been various studies that have conducted on the house, which focus on its performance and occupants' thermal comfort (Jankovic and Huws, 2012; Huws and Jankovic, 2014). The house has been comprehensively simulated in parallel with detailed instrumental monitoring, which is another reason for choosing it for this research.

Various models have been constructed to simulate the Zero Carbon House using tools such as IES-VE and DesignBuilder (DesignBuilder, 2015), the latter being preferred for the optimisation technique it provides. Optimisation refers to the selection process that looks for the best solution in relation to certain criteria from a solution space that contains a set of available alternatives (George, 2014). Performing multi-objective optimisation minimises the number of candidate solutions, while searching for the range of possible solutions and trade-offs that fulfil environmental, social and economic criteria for zero carbon design. Geometry of the DesignBuilder model is shown in Figure 4.

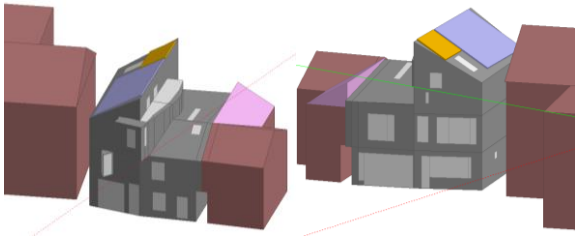

Figure 4: Birmingham Zero Carbon House model geometry - front and rear view

A comprehensive data monitoring system was installed in the house, which consists of internal temperatures, relative humidity, and energy flow sensors, as well as external air temperature sensor and solar radiation instrument. Hence, accurate monitored data were collected and used for the calibration purpose discussed in this paper.

Another challenge to perform calibration of a real building is to obtain real weather data relevant to the building location. Actual weather data was collected from The Centre for Environmental Data Archival (CEDA) (ceda.ac.uk, 2015), which represent the closest viable data to zero carbon house site. This weather data file was subsequently modified, to include site-specific measurements obtained from the instrumentation system in the Zero Carbon House. This weather data was converted into '.epw' format used by EnergyPlus (Crawley, 2001) since DesignBuilder uses EnergyPlus as core engine to perform simulation. We have run the optimisation with two objectives: discomfort hours and carbon emissions produced by the building.

We calculated the first objective by generating temperature distribution scatter graphs showing the relative humidity and operative temperature intervals during the occupied period. As Zero Carbon House is heavily insulated, air temperature and internal surface temperatures are quite similar, and therefore air temperature was considered to be a reasonable approximation of the operative temperature. Subsequently, we used ANSI/ASHRAE Standard 55-

2004, illustrated in Figure 5, for thermal environmental conditions for human occupancy in order to calculate the total discomfort hours for one year as explained below.

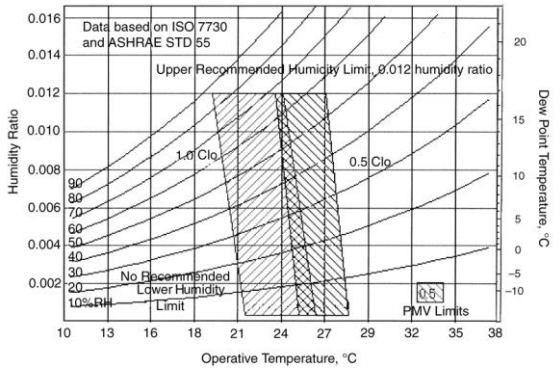

Figure: 5 Acceptable range of operative temperature and humidity for spaces that meet the criteria specified in ASHRAE 55-2004.

We have developed a simple programming script in Java that determines whether a point or a set of points are inside the comfort polygon or not. Using this script, we were able to calculate the number of comfort hours that fall within or intersect with the boundary of comfort hours. The boundaries of the polygon are defined by the upper and lower recommended humidity ratio and operative temperature as shown in Figure 5. Figure 6 shows the results of mapping of monitored values to ASHRAE-55 diagram expressed as Relative Humidity/Operative temperature relationship.

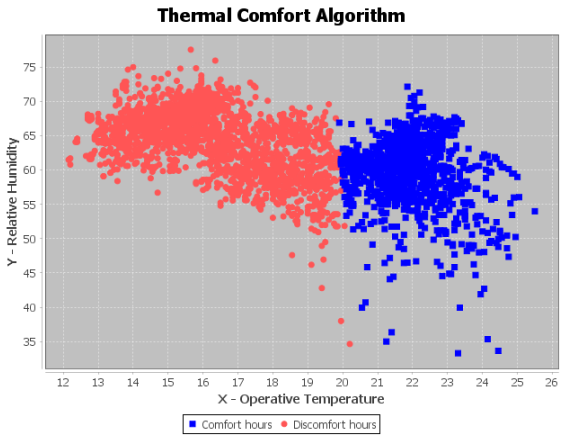

Figure 6: Comfort hours (in blue) and discomfort hours (in red) derived using ASHRAE 55-2004.

RESULTS

Using the thermal comfort algorithm, we realised that the total number of comfort hours is 2128 in year 2012. From related studies (Jankovic, 2012) we obtained CO_2 emissions produced by the building during the same year, showing carbon negative performance of -661.60 kgCO2 in the same year.

We used these results to form a reference point in the solution space, and use the KNN and the proposed density avoidance algorithm to find the closest neighbours to the reference point, hence finding the closest solutions between the measured and simulated results. With this knowledge we were able to explore possible solutions - in the form of theoretical extensions or refinements to the input values of the model parameters.

Table 1 below shows the parameters used to calibrate the model such as heating set point temperature, natural ventliation, infiltration and all heat gains representing the lumped gains into the space from people, equipment, lights etc. These most significant parameters were identified in earlier studies (Jankovic and Huws, 2012; Huws and Jankovic, 2014).

Table 1:
Optimisation / parametric analysis settings used for the building model

Name	Min Value	Max Value	Step
Heating set point temperature (°C)	15	20.00	0.10
Natural ventilation rate (ac/h)	0.00	12.00	2.00
Nat vent set-point temp (°C)	21.00	30.00	5.00
Infiltration (ac/h)	0.40	4.00	0.20
All heat gains (W/m2)	0.00	80.00	20.00
External window operation schedule	-	-	3 Options of window operations
External wall construction	-	-	3 Options of wall constructions

Optimisation was performed remotely using the ENSIMS X3200 Simulation Server located at the University. This allowed quick simulation and optimisation, minimising the number of results in the solution space while finding a trade-off between the input design parameters according discomfort hours and CO_2 emissions. When optimisation was completed, the results were exported as '.csv' files, and the reference point was subsequently embedded in the results, making the file ready to be processed by the KNN with the density avoidance algorithm.

Figure 7 shows the initial results from the optimisation process using DesignBuilder. This Figure also shows the reference point as a blue diagonal cross in the solution space. The dark grey solutions represents the Pareto fronts from re-

optimising the building model with various sets of parameter ranges.

Figure 8 shows a zoomed in view towards the reference point. It shows the neighbour solutions being identified by the KNN without using the density avoidance algorithm. Given the total number of points in the solution space demonstrated in Figure 7 that is equal to 2189 solutions, using the square root, the number of the k neighbours appears to be equal to 46 solutions in Figure 8.

KNN automatically identified the closest solutions to the reference point, and reduced the results to 46, hence minimising the time needed to calibrate the results further towards the reference point. However, those neighbours do not cover other reasonably close candidates located in different regions of the solution space.

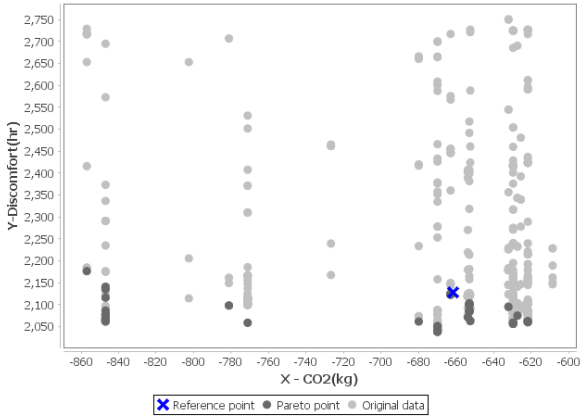

Figure 7: Optimisation results and the imbedded reference point as a blue diagonal cross in the solution space

In contrast, Figure 9 shows the results of implementing KNN with density avoidance algorithm. This Figure shows clearly that our approach manages to select neighbours located in reasonable distance from the reference point, but also covers all solutions in all four regions around that reference point. In fact, the density avoidance algorithm manages to produce this solution with one third of number of k neighbours, namely 12 neighbours, as shown in Figure 9. More details are shown in Table 2, where the solution entries are organised by distances from the reference point. The maximum and minimum of each output is subsequently calculated to find the values within which the reference point resides.

DISCUSSION

The followings are some of the benefits of the use of KNN with density avoidance algorithm as a calibration method of building simulation:

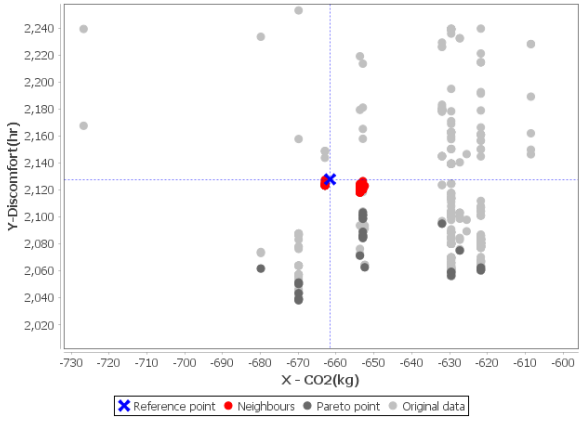

Figure 8: Results of KNN without the use of density avoidance algorithm

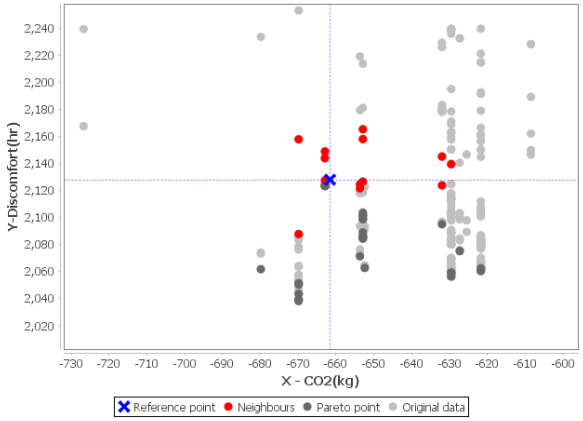

Figure 9: KNN in operation while using the density avoidance algorithm

1- This is standalone application that is easy to use on any existing results generated from various building optimisation/simulation tools such as DesignBuilder, IES, JEPlus+EA, EnergyPlus; 2- it is quick to execute and can deal with multiple number of reference points in a large solution spaces. For each reference point the algorithm performance is equivalent to $O(N)$ as it will grow linearly in direct proportion to the size of the solution space; 3- the tool identifies the parameters and the value range led to the solutions range. The building model can be fine-tuned further by breaking the solution range into smaller values to be used as input for further simulations/optimisation while aiming for one-to-one relationship between the monitored and simulated data; 4- it reasonably avoids the dilemma of dense areas in the solution space; 5- more importantly, while density detection effectively selects the closest solutions around the reference point, it also minimises the number of k neighbours that typically resulted from using KNN without the density avoidance algorithm.

This further reduces the time and computation cost required to reach the one-to-one calibration. Moreover, it minimises the range of solutions if recursive simulation is required for fine-tuned calibration.

Table 2: Detailed parametric settings of the K neighbour solutions (displayed in Red in Figure 9)

Iteration	Generation	Heating set-point temperature (°C)	Natural ventilation rate (ac/h)	Nat vent set-point temp (°C)	Mechanical ventilation rate (ac/h)	All gains (W/m2)	External window operation schedule	External wall construction	CO2 (kg)	Discomfort (hr)
93	23	17.0	9.0	22.3	3.2	64.9	1	3	-663	2128
754	10	16.8	11.1	23.6	1.1	11.3	1	3	-654	2125
159	41	16.8	7.5	22.4	3.3	64.5	1	3	-653	2127
754	10	16.8	11.1	23.6	1.1	11.3	1	3	-654	2122
50	12	17.3	8.4	22.4	3.3	64.5	1	3	-663	2144
21	5	17.1	8.3	22.4	3.4	41.1	1	3	-663	2149
138	37	16.6	10.4	21.0	2.1	36.2	1	2	-632	2124
61	15	18.7	6.6	22.3	2.0	46.0	1	3	-670	2158
62	15	17.3	6.6	22.1	3.2	65.5	1	3	-653	2158
722	2	16.8	11.1	24.6	1.1	11.3	1	3	-630	2140
41	10	15.9	10.4	23.0	2.2	37.5	1	2	-632	2145
59	15	17.3	6.6	23.0	3.2	64.5	1	3	-653	2165
11	2	20.0	9.6	23.2	1.9	34.0	1	3	-670	2088
516	143	19.7	11.9	23.2	2.7	28.3	1	1	-627	2103
578	162	19.7	11.8	21.1	2.9	20.5	1	2	-632	2097
524	146	19.7	11.9	21.1	2.7	28.2	1	3	-622	2112

However, there are some challenges, which appeared when implementing the KNN with density avoidance technique. For example, for the density avoidance to work well, the user needs to identify the best value for the radius R, density and the k neighbours. Despite this, it is important to mention that the density avoidance work best with lower number of k recommended for KNN. This means the outcome will be fewer results to deal with but only those which are the best in the solution space. Therefore, it is a good idea to start with $k = \sqrt{n}$ and minimise it appropriately. We used $\frac{1}{3}k$ in the study discussed in this paper. For radius R and density threshold, we calculated the average of both from the solution space as a starting point, and then minimised their values further if necessary.

FUTURE WORK

The optimisation method used in DesignBuilder uses the Non-dominated Sorting Genetic Algorithm II (NSGA II) (Deb, 2002). NSGA II searches through the solution space to find a set of optimal trade-offs, while treating all objectives as being equally important (i.e. non-dominated solutions) and the output set contains the optimal solutions. For example, finding the design that cost the least but also produces the least carbon dioxide to the environment. These solutions are called Pareto fronts. We attempted to adjust the optimisation function in DesignBuilder to find optimal solutions that are the closest to the reference point instead of the default optimisation objectives. Due to software limitations, it was not possible to adjust the objective functions in DesignBuilder in this way in order to suit the calibration requirements. For future work, we are planning to export the IDF file from DesignBuilder and perform optimisation using RetrofitPlus web Application (Basurra, 2014) that uses JEPlus +EA as core engine. The optimisation through NSGA II will search the solution space to find a set of optimal trade-offs while targeting to the reference point. Unlike KNN which required density avoidance mechanism to find the results around the reference point, NSGA II has a built in density function to estimate density of dominant solutions around the optimal solutions. This is performed by calculating the average distance to other points on either side of the solution. This density value is the so-called crowding distance, and is used to prioritise non-dominant solutions when they have similar ranks. In this case, NSGA-II chooses the solution that exists in the less dense area in the graph (Deb, 2002). Our aim will be to compare KNN with density avoidance and optimisation through NSGA-II for calibration of building simulation, and evaluate both approaches.

CONCLUSION

A study of calibration method for a thermal performance model of a building was presented. The starting point was the creation of the base model. Data from architectural drawings and site measurements were used to build the exact geometry of the house, and all building systems, fabric, lighting and equipment are specified to closely correspond to the actual. Occupancy and usage of the house were based on observations, questionnaires and data from the monitoring. In a typical simulation/optimisation analysis, the usual aim is to search for the optimum performance points. In this analysis, the aim was to locate the performance points of the simulation model that are the closest to the actual performance, and these optimum performance points were then

used to find out the corresponding model parameters that resulted in the smallest performance gap.

A k Nearest Neighbour (KNN) algorithm was used to identify the solutions with the lowest performance gap based on a set of reference points that represented the actual performance of the real building. Density avoidance algorithm was used to further refine the solutions by finding regions in the space of input factors for which the model output is either maximum or minimum to meet optimum criterion of one-to-one relationship between the simulated and actual data.

The results suggest that the predictive performance of simulation models can be calibrated quickly and accurately using the monitored performance data of the real building. Automating such process increases its efficiency while reducing the time and effort required for calibration.

ACKNOWLEDGEMENT

The authors acknowledge the financial support provided by the Innovate UK (formerly Technology Strategy Board) through the Retrofit Plus project funding, Grant Reference 101614.

REFERENCES

Alt, H. (2001). The nearest neighbor. Computational Discrete Mathematics: Advanced Lectures, 2122, 13-24.

Basurra, S and Jankovic, L. (2014) “Development of an Expert System for Zero Cabron Design and Retrofit of Buildgs”. Zero Carbon Buildings Today and in the Future - Birmingham City University, UK, 11-12 September 2014.

Ceda.ac.uk, (2015) Centre for Environmental Data Archival. [online] Available at: http://www.ceda.ac.uk/ [Accessed May 2015].

Christophers J., (2014) " ZERO CARBON HOUSE, BIRMINGHAM: SUN, LIGHT AND MATERIAL". In Proceedings of the 1st International Conference on Zero Carbon Buildings Today and in the Future, Birmingham City University, 11-12 September 2014.

Clarke J A, Strachan P and Pernot C (1993) `An Approach to the Calibration of Building Energy Simulation Models', ASHRAE Transactions, 99(2), pp917-927, 1993.

D. Crawley, L. Lawrie, F. Winkelmann, W. Buhl, J.Y. Huang, C. Pedersen et al. EnergyPlus: creating a new-generation building energy simulation program, Energy Build, 33 (2001), p. 443.

Dantzig, G. B. (2014) The Nature of Mathematical Programming. [Online] Available from:http://glossary.computing.society.informs.org/index.php?page=nature.html. [Accessed: 12 May 2015]

Deb, K. et al., 2002. A Fast and Elitist Multiobjective Genetic Algorithm :, 6(2), pp.182–197.

Designbuilder.co.uk, (2015). DesignBuilder - building simulation made easy. [online] Available at: http://www.designbuilder.co.uk/ [Accessed 12 May 2015].

Energy Saving Trust. (2014) Green Deal and ECO. [Online] Available from:http://www.energysavingtrust.org.uk/Take-action/Find-a-grant/Green-Deal-and-ECO#apply. [Accessed: 14 April 2015]

Heo, Y., Choudhary, R. & Augenbroe, G. a., 2012. Calibration of building energy models for retrofit analysis under uncertainty. Energy and Buildings, 47, pp.550–560.

Huws, H. and Jankovic, L. (2014). "Birmingham Zero Carbon House - Making It Stay Zero Carbon Through Climate Change". In Jankovic, Ljubomir, Ed. (2014) Zero Carbon Buildings Today and in the Future -Proceedings of a conference held at Birmingham City University, 11-12 September 2014.

Jankovic, L. and H. Huws (2012) "Simulation experiments with Birmingham Zero Carbon House and optimisation in the context of climate change". In proceedings of BSO 12 - Building Simulation and Optimization, 11 - 12 September 2012, University of Loughborough.

Jankovic. L. (2012) Designing Zero Carbon Buildings Using Dynamic Simulation Methods. Routledge. London and New York.

Paul Raftery, Marcus Keane, James O’Donnell, Calibrating whole building energy models: An evidence-based methodology, Energy and Buildings, Volume 43, Issue 9, September 2011, Pages 2356-2364, ISSN 0378-7788.

Richard O. Duda, Peter E. Hart, and David G. Stork. 2000. Pattern Classification (2nd Edition). Wiley-Interscience.

Taheri, M., Tahmasebi, F. & Mahdavi, A., 2013. A case study of optimization-aided thermal building performance simulation calibration. Proceedings of BS2013, Chambéry, France, pp.603–607.

Tahmasebi, F., Mahdavi, A., 2012. Optimization-based simulation model calibration using sensitivity analysis, in: Simulace Budov a Techniky Prostredi, O. Sikula, J. Hirs (ed.); Ceska Technika - nakladatelstvi CVUT, 1 (2012), ISBN: 978-80-260-3392-9; Paper ID 71.

Yiqun Pan, Zhizhong Huang, Gang Wu, Calibrated building energy simulation and its application in a high-rise commercial building in Shanghai, Energy and Buildings, Volume 39, Issue 6, June 2007, Pages 651-657, ISSN 0378-7788.