Optimisation as a Tool for Gaining Insight: An Application to the Built Environment

Eric S Fraga, Abdellah Salhi, Di Zhang, and
Lazaros G Papageorgiou

reprinted from

Journal of
Algorithms & Computational Technology

Volume 9 · Number 1

March 2015

Multi-Science Publishing Co. Ltd

Optimisation as a Tool for Gaining Insight: An Application to the Built Environment

Eric S Fraga,[1] Abdellah Salhi,[2] Di Zhang,[1] and Lazaros G Papageorgiou[1]
[1]Centre for Process Systems Engineering, Department of Chemical Engineering, University College London (UCL)
[2]Department of Mathematical Sciences, University of Essex

Received 30/11/2013; Accepted 01/08/2014

ABSTRACT
The design of heating systems for dwellings using new technologies, or new versions of old technologies, requires the ability to predict the temperatures in a dwelling. The temperature behaviour can be modelled, typically by differential equations which incorporate thermal driving forces and the thermal inertia of a dwelling. The development and characterisation of these models is usually based on fitting data accumulated over sufficient time to capture the behaviour of the dwelling under different conditions (summer, winter, etc.). Model fitting relies on assumptions about the behaviour of the system.

Optimisation can be used to examine these assumptions and gain insight into this behaviour. This paper describes the application of a nature inspired algorithm, known as the *Plant Propagation Algorithm,* a variant of a *Variable Neighbourhood Search* algorithm, to the problem of modelling a dwelling heated by an air source heat pump. The algorithm is evaluated using different population evolution strategies and implemented using a simple parallel computing paradigm on a multi-core desktop system. The results are used to identify potential sources of missing data which could explain the observed behaviour of the dwelling.

Keywords: built environment; dwelling modelling; model fitting; variable neighbourhood search

---

Corresponding author: e.fraga@ucl.ac.uk
This paper was presented at the EGH2013 Workshop, University of Essex, June 2013.

1. INTRODUCTION

Model fitting is the problem of identifying the best values for a set of unknown parameters in a model to achieve the best *fit* to a set of data. A wide range of methods exists for model fitting but, at its core, model fitting is about optimisation. Depending on the underlying model chosen to fit to the data, this optimisation problem can be challenging for reasons of nonlinearity, convexity and scale. Further, the quality of the model derived is dependent not only on the optimisation method employed but also on the assumptions inherent in the choice of model characteristics to fit. Optimisation can be a useful tool to fit a model; however, it may also be used to identify the best model features. Through interactive and iterative use of optimisation, insight into the behaviour of the system being modelled can be gained. This insight can be used to further improve the model by, for instance, the generation of more data, possibly from different sources of measurement.

In the built environment, models are required to describe the thermal behaviour of dwellings. In the case study presented below, a model is required to determine the temperature of a room in a house. This temperature is affected by direct space, heating and by thermal gains from and losses to the environment. The dynamic nature of the temperature in a room leads to a differential equation based model. Evaluating the fit of a differential equation over a time domain is computationally expensive and the amount of data required for a good representation may be large.

This paper describes the use of a Plant Propagation Algorithm [5] for fitting a complex dynamic model to a typical 4 bedroom dwelling in the UK. The implementation of the algorithm together with computational aspects of the performance are discussed. The evolution of the model, from a standard model used by practitioners in the built environment discipline to a model which captures unknown sources and sinks of heat, is presented.

2. MODELLING A DWELLING

The dynamic model most frequently used [2] for predicting the temperature in a given room, $T_r(t)$ in a dwelling is

$$C\frac{\mathrm{d}}{\mathrm{d}t}T_r(t) = W_{\mathrm{hp}}(t) - L\Delta T(t) \tag{1}$$

where $\Delta T(t)$ is the *driving force,*

$$\Delta T(t) \overset{\mathrm{def}}{\equiv} T_r(t) - T_e(t) \tag{2}$$

$T_e(t)$ is the temperature in the outside *environment*. All temperatures are in °C.

The unknown parameters in this model, which we wish to fit using data collected, are *C,* the *thermal capacity* of the house in kWh °C^{-1}, and *L,* the *heat loss* due to the driving force (eq. 2), in kW°C^{-1}. W_{hp} is the output of the heating equipment in kW; for the case study below, this will be the output of an *air source heat pump*. The model assumes that there are no other sources or sinks of heat; we will come back to this assumption later below.

Fitting data to a model consists of identifying the values of the free parameters which minimise some estimate of the error in the fit. Mathematically, this problem can be described as:

$$\min_{C,L} z = \| T_r - \hat{T}_r \| \tag{3}$$

subject to Eq. 1 and where $\hat{T}_r$ represents temperature data, $||\cdot||$ is an appropriate norm and $t \in [t_1, t_n]$ is the time period over which the dwelling's behaviour is simulated.

This is a dynamic nonlinear optimisation problem (DNLP) to minimise z by manipulating *C* and *L* subject to the dynamic model for the behaviour of the room temperature.

The solution obtained by solving the problem defined in Eq. 3 is shown graphically in Figure 1. For this paper, we consider a time period of 1 day with data available in 5 minute intervals. This problem was modelled in GAMS [3] through a simple first order discretisation of the derivative term and solved using the combination of BARON and CPLEX. Although the fit follows the actual behaviour in a general sense, i.e. qualitatively, there are periods in the day during which the fit is not good. This same behaviour is found when analysing other days in the year. The quality of the solutions obtained motivates us to consider the underlying model and whether it is a good representation of how the dwelling behaves.

3. AN IMPROVED MODEL OF THE DWELLING

There are three assumptions inherent in the simple model (Eq. 1):

1. heat transfer is the result of a linear driving force,
2. this heat transfer takes place instantaneously upon a change in that driving force, and
3. the data available for fitting the model are correct and comprehensive.

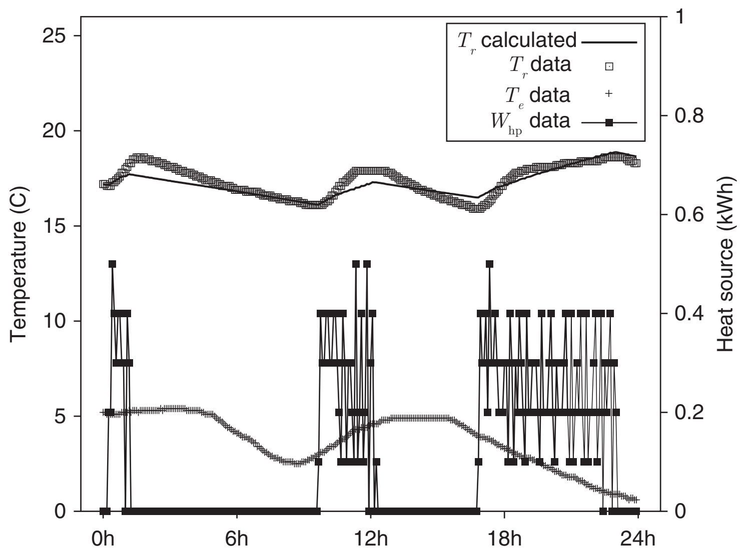

Figure 1. Model simulation and data used to fit the model based on a 4 bedroom dwelling on the 30th of November from midnight to midnight. T_e is the outside temperature, T_r the room temperature and W_{hp} the output of the heat pump system used to provide space heating in the dwelling.

The aim of the work presented in this paper is to explore these assumptions through the use of optimisation. We start by relaxing the first two assumptions and we will return to the third assumption after further analysis.

A nonlinear driving force could, in principle, take any form. For simplicity, and for an initial investigation, we will assume that a polynomial may provide a suitable relaxation of the assumption and that a cubic polynomial will be sufficient to capture nonlinearity in the behaviour. A cubic equation will have four parameters, the coefficients of the 0th through 3rd power terms. By asymptotic analysis, we can make the following conclusions:

1. The constant term should be identically 0 as we expect no driving force when there is no difference in temperature (i.e. when $\Delta T \equiv 0$).
2. The coefficient on the quadratic term should also be 0 as we expect the temperature in the room to be driven up when the temperature difference is negative and the temperature in the room to go down when the difference is positive.

With these choices, the driving force component of Eq. 1 would be

$$L\Delta T(t) \Rightarrow \alpha\Delta T(t) + \gamma\Delta T^3(t) \tag{4}$$

$$= \Delta T(t) \cdot (\alpha + \gamma\, \Delta T^2(t)) \tag{5}$$

We write Eq. 5 in this form to show the link with the original term in Eq. 1: α is equivalent to L when $\gamma = 0$.

The second assumption, that heat transfer takes place instantaneously, is relaxed by adding time delays to the heat source and driving force terms separately: the current (at time t) rate of change in the temperature is due to the heat input from the heat source (i.e. the heat pump) some time before, $t - \lambda_2$, and also to the driving force from the environment due to the difference in temperatures at some earlier time, $t - \lambda_1$.

Taking both relaxations into account, we end up with a more general model:

$$C\frac{\mathrm{d}}{\mathrm{d}t}T_r(t) = W_{\mathrm{hp}}(t - \lambda_2) \tag{6}$$

$$-\Delta T(t - \lambda_1) \cdot \left(\alpha + \gamma\Delta T^2(t - \lambda_1)\right)$$

There are now 5 variables which must be adjusted to fit the data: $\{C, \lambda_1, \lambda_2, \alpha, \gamma\}$. The new model extends the previous model: if λ_1, λ_2 and γ are 0, we end up with the same $L = \alpha$. In other words, Eq. 6 is a generalisation of Eq. 1.

The introduction of time delays converts the ordinary differential equation into a *delay differential equation* which requires special attention to solve. In particular, a simple discretisation is no longer suitable. Further, the addition of extra optimisation variables has made the problem larger and the behaviour more complex. Figure 2 shows the behaviour of the objective function (Eq. 3) using the generalised model (Eq. 6). The objective function is seen to be non-convex (recall, the goal is minimisation) and *noisy;* it may also be multi-modal but this cannot be determined from a single hyper-line. The noise is a result of the numerical solution of the delay differential equation and the sensitivity of this solution to the parameter values.

To cater for the extra complexity, an evolutionary optimisation known as the *Plant Propagation Algorithm* [5] has been used and is briefly described in the next section.

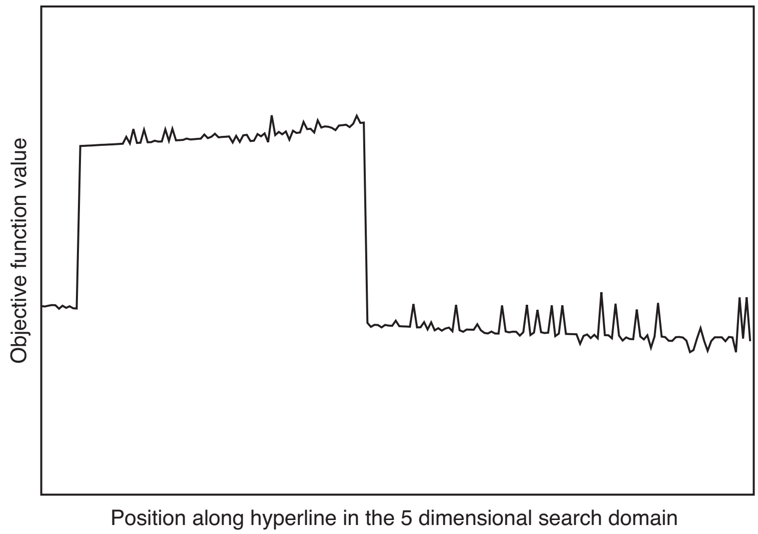

Figure 2. Plot of objective function value along a hyperline connecting two arbitrary points in the 5 dimensional search space.

4. THE PLANT PROPAGATION ALGORITHM

The Plant Propagation Algorithm (PPA) is a Nature-inspired algorithm for optimisation and search. It emulates the way plants, in particular the strawberry plant, propagate. A basic PPA, called *Strawberry,* has been described and tested on single objective as well as multi-objective continuous optimization problems in [5]. Although the problems were of low dimension, the results demonstrated robustness with respect to parameter values, making it simple to apply, and high probability of finding good solutions. This algorithm, shown in Algorithm 1, is an example of a *Variable Neighbourhood Search* (VNS) algorithm.

Algorithm 1 The Pseudo-code of PPA, [5]

Initialization:

- Generate a population $P = \{X_i,\ i = 1,\ \ldots, n_P\}$;
- Set g, the generation counter to 1: $g \leftarrow 1$;

Main Algorithmic Body of the PPA:

1: **for** $g = 1 : g_{\max}$ **do**

2: Compute fitness $N_i = f\ (X_i),\ \forall X_i \in P$;

3: Sort *P* in ascending order of N;

4: Create new population Φ;

5: **for** $i = 1, \ldots,\ n_P$ **do**

6: Select X_i from P;

$\boldsymbol{r}_i \leftarrow$ set of runners where both the size of the set and the distance for each runner (individually) are functions of the fitness N_i. See Eq. 7 and 8;
$\Phi \leftarrow \Phi \cup r_i$ {append to population};
end for
$P \leftarrow$ new population (see text);
end for
Return: ***P***, the population of solutions.

Selection can be done any number of ways, from simply selecting the n_p best solutions in *P* through to purely random selection. For the case study below we are using size 2 tournament selection, selecting from the whole population. This provides some evolutionary pressure towards the most fit solutions while still allowing all solutions to participate in propagation.

The number of runners, $n_{r,i}$ for the solution i is determined by

$$n_{r,i} = \left\lceil N_i \rho_1 n_{r,\max} \right\rceil \tag{7}$$

for some random number $\rho_1 \in [0,1]$ and the distance for each runner is

$$\delta_{i,j} = (1 - N_i)_{\rho_2} - 0.5 \qquad \forall j = 1, \ldots, n_d \tag{8}$$

where $\rho_2 \in [0,1]$ is another random number and n_d is the dimension of the problem (number of optimisation variables). The distance δ is then used to modify the given point X_i towards the boundary of the domain. The number of runners is proportional to the fitness whereas the distance to propagate is inversely proportional, as motivated by the biological analogy.

There are four possible approaches to updating the population from one generation to the next. The new population can be:

1. Solely the new solutions found by propagation;
2. The new solutions together with those selected for propagation;
3. The new solutions with an elite set; or,
4. The new solutions, an elite set and the solutions chosen for propagation.

All approaches are considered in the case study below to see how they compare. Each can be argued as appropriate from a biological perspective.

Besides its simplicity and robustness, another attractive feature of this method is its inherently parallelisable nature. Our optimisation problem falls into the category of optimisation problems with *computationally expensive,* objective functions. Each evaluation of a point in the search space requires the solution of the delay differential equation over the time domain. Depending on the time period selected, a single evaluation can take many seconds through to hours. An effective search therefore will require significant computation and parallel computing is one solution to this requirement.

5. IMPLEMENTATION

The algorithm presented above has been implemented in Octave 3.2[1] with the capability to make use of multiple processing cores if available. The evaluation of the objective function, z, as shown in Eq. 3, is done as follows:

1. Assign the point X to the individual parameters used by the model: λ_1, λ_2, α, γ and C.
2. Simulate the dwelling's behaviour over the design time domain by solving the model (Eq. 6) using the lsode method [4]. This solution is executed in small time steps to generate the data required for providing access to data sufficiently fine for the delay element.
3. From the full simulation data, the values at the time points present in the original data set are extracted.
4. The difference between the simulated data values and the collected data values are calculated using a suitable norm. We have used a 2-norm for the results presented below.

The lsode method is robust and the differential equation is stable under most circumstances. As a result, the objective function is feasible in all cases we have considered.

Using this procedure to evaluate z, the Plant Propagation Algorithm described above is implemented as follows to allow for parallel execution.

1: **for** $g = 1 : g_{\max}$ **do**
2: $N \leftarrow$ fitness (P)
3: $V \leftarrow$ new solutions based on selection and runner generation, as described in Algorithm 1
4: evaluate, in parallel, z for each point in V
5: $P \leftarrow$ new population based on existing population and new points, subject to evolution strategy option chosen
6: **end for**

The parallel evaluation of the new points is done with the parcellfun function in Octave. This evaluates a given expression on a number of cores simultaneously and is suitable for the parallel evaluation of computation that requires no inter-core communication while being evaluated. This is ideal for our case where the main computational effort is the solution of the delay differential equation.

---

[1]http://www.octave.org/

Table 1. Statistical summary for the different population evolution strategies with 10 runs for each strategy.

Strategy	z_{min}	z_{ave}	σ	z_{max}	n_{gen}
1	7.59	9.12	1.44	12.28	57
2	6.21	7.64	1.08	9.15	116
3	6.29	7.28	1.10	9.43	126
4	5.46	6.73	0.94	8.52	157

6. MODEL FIT

The Strawberry algorithm has been run on an 8 core Intel i7 2.4GHz system with 8 GB of RAM and operating under Linux. Using data from the Energy Saving Trust[2], a $63m^2$ four bedroom house was selected. A single day, 30 November, was chosen as a good representative of the range of behaviour we wish to capture in the model, including heat pump operation and changing environment temperature. This example was used above; see Figure 1.

Table 1 shows the statistical analysis of 10 runs for each population evolution strategy. z is the objective function value with z_{min} and z_{max} being the best and worst solution found over the 10 runs for each evolution strategy. The values for z_{ave} and n_{gen} are averages over the 10 runs and σ is the standard deviation of z over the 10 runs. Two stopping criteria were implemented: a maximum number of generations, $g_{\text{max}} = 1000$, and stopping if no improvement in the best found solution is made in 50 generations. The n_{gen} values are the last generation in each run for which the best solution improved.

We see that, statistically and in an absolute sense, the best strategy is the fourth. This strategy uses both an elite set, of size 1, and the members of the population selected for propagation in the creation of the new population. The number of generations for convergence is also the largest for this strategy. This indicates that the strategy is performing a more thorough search of the domain. The fourth strategy is able to avoid getting stuck in a particular part of the search domain for longer.

We can use the solutions obtained to gain some insight into the solution space for this case study. The parallel coordinate plot shown in Figure 3 shows the five optimisation variables along with the objective function value for the 10 individual points found by the fourth strategy. All the values have been normalised based on the domain for the search for the optimisation variables

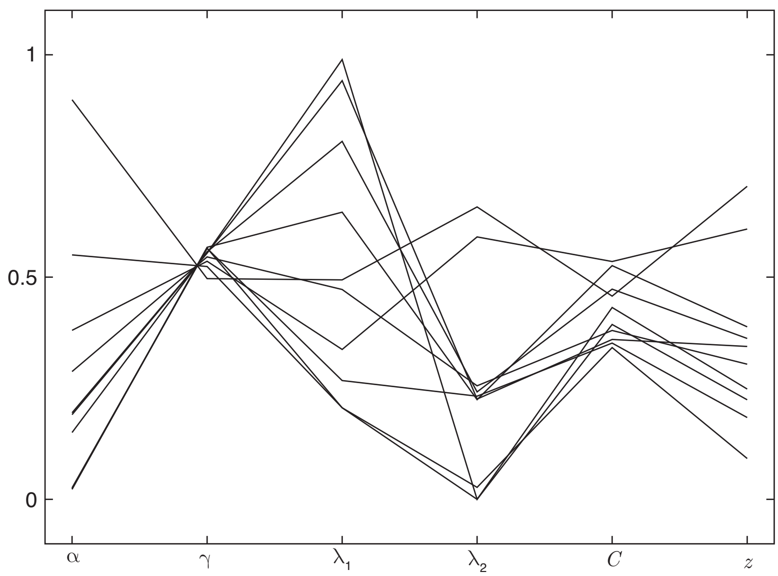

Figure 3. Parallel coordinate system plot of the better solutions obtained in the 40 runs. Each poly-line represents a different solution. The variables and the objective function value have been normalised. For the variables, the full domain is included, from lower bound to upper bound: $\alpha \in [-0.1, 0.1]$, $\gamma \in [-0.1, 0.1]$, $\lambda_1 \in [0, 5]$, $\lambda_2 \in [0,1]$ and $C \in [1,10]$. For the objective function value, the range is [5,10].

and for the range [5,10] for the objective function value. We see some evidence of clustering, most prominently for γ but also to some degree for λ_2, the time delay for the operation of the heat pump to affect the room temperature.

7. MISSING DATA

If the solutions identified above were plotted, the result would be visually similar to Figure 1, with the difference in the behaviour of the fit only marginally discernible by superimposing the two plots. In other words, although the fit is improved, as measured numerically by the objective function, qualitatively the behaviour is no different. The two assumptions that have been relaxed do have an impact but it is minimal. We therefore now consider the third assumption, that the data available is not comprehensive, and use optimisation to explore this assumption.

The model described above, whether with or without time delays, assumes that the only inputs and outputs of heat are the heat pump and the external environment in the form of the driving force. However, many other possible sources and sinks of heat are possible in a dwelling including, for instance, incident solar radiation, cooking and other activities, and human occupancy itself (humans radiate on the order of 100 W or more, depending on the activity [1]). Also, even just opening windows and doors could have a significant effect on the temperature within a house. The models above do not include any terms for these possible sources or sinks of heat.

As the data available also do not include any of these alternative sources or sinks, it is not possible to model these directly. However, it is possible to use the last model above (Eq. 6) as the basis for analysis to help identify when other sources and sinks of heat may be having an impact on the temperatures in the house. To investigate these, a *slack* heat source, $W_{\mathrm{so}}(t)$, and a slack heat sink, $W_{\mathrm{si}}(t)$, are added to the model:

$$C\frac{\mathrm{d}}{\mathrm{d}t}T_r(t) = W_{\mathrm{hp}}(t-\lambda_2) \tag{9}$$

$$-\Delta T(t-\lambda_1)\cdot\left(\alpha+\gamma\Delta T^2(t-\lambda_1)\right)$$

$$+\ W_{\mathrm{so}}(\mathrm{t}) - W_{\mathrm{si}}(t)$$

This model is more complex, in scale, than the previous versions. The optimisation procedure must now determine values for continuous functions $W_{\mathrm{so}}(t)$ and $W_{\mathrm{si}}(t)$ together with the parameters already described. The addition of slack variables, without constraints, means that the data can be fit exactly. However, this is not necessarily desirable as it means that any noise in the data is also taken into account in the fit. To avoid fitting noise, we introduce a penalty into the objective function and the optimisation problem now becomes

$$\min z = ||T_{\mathrm{r}} - \hat{T}_{\mathrm{r}}|| + p\,(||W_{\mathrm{so}}|| + ||W_{\mathrm{si}}||) \qquad p \geq 0 \tag{10}$$

subject to the model in Eq. 9.

Solving this extended model directly is computationally difficult and expensive. Instead, we have reformulated this model as a linear programme (LP) through the explicit discretisation of the time domain using a 5 minute time step to match the granularity availablein the data sets. The problem becomes one of finding the discrete set of values W_{so} and W_{si} at the discrete time values in the data set. The results from the visualisation (cf. Figure 3) were

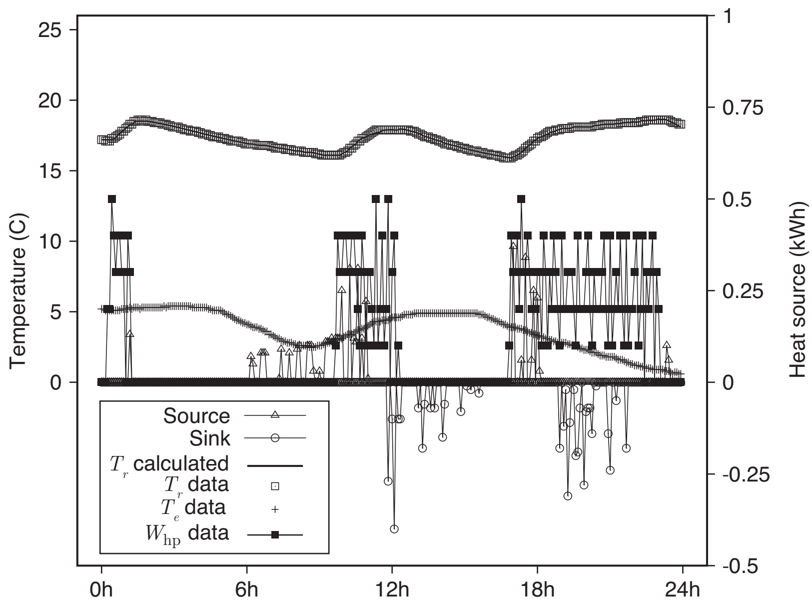

Figure 4. Model fit using expanded model which caters for unknown sources and sinks of heat [6]. Upward spikes with triangles denote possibly unknown sources of heat and downward spikes with circles possible unknown sinks of heat.

used to fix the λ variables for subsequent analysis to remove the difficulties associated with the solution of delay differential equations. Specifically, a delay of $\lambda_1 = 2$ hours for external temperature effects and a delay of 15 minutes $(\lambda_2 = 0.25)$ for the impact of the heat pump were used [6]. These values are reasonable, from an engineering standpoint, and correspond to one of the best solutions obtained. Other values could be used, of course.

Although large, the resulting LP problem is tractable and can be solved in a few seconds. We have implemented this model in GAMS [3]. The resulting model's behaviour is illustrated in Figure 4 with a penalty value $p = 5$. This figure shows the missing sources and sinks of heat identified by the slack variables. Possible heat sources are indicated by spikes upwards from 0 with a triangle and possible heat sinks (or losses) by spikes downwards with circles. Heat sources are identified in the early morning hours (6-10am) and heat losses both in the early afternoon and mid-evening.

We can now conjecture on the possible reasons for the identified missing sources and sinks of heat. For instance, the early morning heat sources could be due to cooking and showering or bath activities by the occupants. The afternoon losses could be due to people entering and leaving the house. Finally, the evening losses could be due to the not atypical process of cooling down the house by opening windows even though the primary heating source is turned on.

8. CONCLUSIONS

This paper has demonstrated the use of optimisation as a means of gaining insight into the behaviour of a complex system. The example used is a dwelling with an air source heat pump, modelled to predict the temperature in a room. A series of steps, using optimisation at each step, has enabled us to evolve a model from its simplest form, commonly used in the built environment literature, through to one that allows us to postulate possible missing sources and sinks of heat.

Due to the complex objective function behaviour, including noise and multimodality, a nature inspired plant propagation algorithm has been used for model fitting when time delays in heat transfer were being investigated. The Strawberry method [5] has been used to solve this problem. Possible alternative population evolution strategies for this method have been explored, including the use of elitism.

The final results show a fit that highlights times during the day where possible missing sources and sinks of heat may be present. We can only speculate as to the providence of these sources and sinks but they do allow us to suggest the collection of further data. For instance: sensors on doors and windows could be placed at little cost; solar data from the meteorological sources could be incorporated; data on appliance use (cookers, television sets, computers) could be collected. With these data, more detailed and accurate predictive models would be possible.

ACKNOWLEDGEMENTS

The authors gratefully acknowledged the financial support provided by the Energy Technologies Institute and the EPSRC (EP/G062129/1) as well as the expert assistance from PassivSystems.

REFERENCES

[1] Binggeli, C. *Building systems for interior designers.* Wiley, Hoboken, NJ, 2010.

[2] Boait, P. J., Fan, D., and Stafford, A. Performance and control of domestic ground-source heat pumps in retrofit installations. *Energy and Buildings 43*, 8 (2011), 1968–1976.

[3] Brooke, A., Kendrick, D., Meeraus, A., and Raman, R. *GAMS: A user's guide.* GAMS Development Corporation, Washington, 1998.

[4] Radhakrishnan, K., and Hindmarsh, A. C. Description and use of LSODE, the Livermore Solver for Ordinary Differential Equations. Tech. Rep. Lawrence Livermore National Laboratory Report UCRL-ID-113855, NASA, 1993.

[5] Salhi, A., and Fraga, E. S. Nature-inspired optimisation approaches and the new plant propagation algorithm. In *Proceedings of ICeMATH 2011, The International Conference on Numerical Analysis and Optimization* (Yogvakarta, 5-8. June 2011), pp. K2:I–8.

[6] Zhang, D., Papageorgiou, L. G., and Fraga, E. S. Optimisation based analysis of a dwelling with an air source heat pump. In *Proceedings of the. 22nd European Symposium on Computer Aided Process Engineering* (2012), I. D. L. Bogle and M. Fairweather, Eds., Elsevier, pp. 312 316.