Proceedings of the 2021 IISE Annual Conference
A. Ghate, K. Krishnaiyer, K. Paynabar, eds.

Season-dependent Parameter Calibration in Building Energy Simulation

Ziang Xu, Cheoljoon Jeong, Eunshin Byon
Industrial and Operations Engineering, University of Michigan
Ann Arbor, MI 48109

Kristen Cetin
Civil and Environmental Engineering, Michigan State University
East Lansing, MI 48824

Abstract

As the energy consumption from residential and commercial buildings makes up approximately three-quarters of the U.S. electricity loads, analyzing building energy consumption behavior becomes essential for effective power grid operation. An accurate physics-based building energy simulator that is built on first principles can predict an individual building's energy response, such as energy consumption and indoor environmental conditions under different weather and operational control scenarios. In the building energy simulator, several parameters that specify building characteristics need to be set *a priori*. Among those parameters, some parameters are season-dependent, whereas other parameters should be globally employed throughout a year. Existing studies in parameter calibration ignore such heterogeneity, which causes suboptimal calibration results. This study presents a new calibration approach that considers the seasonal dependency.

Keywords: Building energy model, gradient descent, optimization

1 Introduction

A physics-based building energy model (BEM), which simulates the building energy operations, has become an essential part in optimizing the building design and operations. In the building energy simulator, several parameters that specify building characteristics need to be set *a priori*. These parameters are referred to as calibration parameters. Physical laws to identify appropriate values of those parameters are often unavailable. Parameter calibration is a process to estimate the parameters using field data.

A predominant approach in the parameter calibration literature is the Bayesian calibration [1]. However, this approach is computationally expensive and inefficient when data size is large. A lightweight Bayesian calibration that uses a linear regression emulator is proposed for dynamic building energy models [2]. In [3] Hamiltonian Monte Carlo sampling is employed to obtain the posterior distribution in Bayesian calibration more efficiently. Chong and Lam [4] reduce the computational cost by using a representative subset of the entire dataset. While these approaches improve the computational efficiency, our prior study suggests that the Bayesian approach provides inadequate posterior when the mean of the prior distribution is not set around the unknown true value [5]. Alternatively, gradient-based algorithms are employed in [6] to choose sensitive parameters and calibrate the model parameters. A detailed review on the BEM parameter calibration is available in [7].

Typically there are a large number of calibration parameters in the BEM. Calibrating all these parameters is not practical. Thus, selecting influential parameters is important for successful model calibration. Massimiliano et al. [8] choose parameters that are related to climatic conditions, location, lighting, control and operation, water loop, air loop, air handling units, and domestic hot water (DHW). Chong et al. [9] select the building wall and material parameters. Building upon these studies and the domain knowledge, this paper focuses on calibrating parameters related to lighting, ventilation, DHW, window material (optical properties), heating and cooling systems, which substantially influence the building energy use.

Among these parameters, lighting, DHW, window material, and ventilation parameters can be considered as global parameters which need to be applied in the year-long simulation. On the contrary, heating and cooling parameters are season-dependent. The aforementioned existing studies do not take the seasonal dependency into consideration in their analysis. In this study, we devise a new optimization algorithm, referred to as block gradient decent (BGD), to account for the seasonal dependency in the optimization procedure. The proposed approach is a gradient-based optimization. The innovation in BGD is that we categorize the parameters into three groups, global parameters, heating season parameters, and cooling season parameters, and optimize them in each group sequentially and iteratively.

When the simulator is a black box computer model, no mathematical closed-form expression is available to quantify the output, given the simulation input. It implies that no gradient information can be directly available or computed. We approximate the gradient using first-order difference. Doing so requires an additional simulation run for every parameter in every gradient approximation. Further, in the gradient descent-based approach, an appropriate step size needs to be employed in deciding how much each parameter can be updated. We employ the backtracking line search method, which is known to make the gradient descent algorithm converge quickly to a critical point. The line search method decides an appropriate step size iteratively, so it requires a new simulation run for each trial of step size. While pursuing efficiency, this additional loop in the line search slows down the calibration process. To reduce computational time, we employ the multi-threads programming technique. Specifically, we approximate all gradients concurrently with multiple threads and also conduct the line search in a parallel way instead of doing sequentially. This multi-thread computing mechanism speeds up the computation by fourteen times faster than the typical computation without multi-thread parallel computing.

The rest of this paper is organized as follows. In Section 2, we present the proposed season-dependent parameter calibration methodology. Section 3 presents the advantage of our algorithm with a case study using real-world data. Finally, Section 4 concludes.

2 Methodology

2.1 Mathematical formulation

Let $\mathbf{x} \subseteq \mathbb{R}^m$ denote the vector of physically observable input variables of dimension m in a system. Let $\mathbf{y}(\mathbf{x})$ denote the physical process response at input $\mathbf{x}$. Let $\mathbf{y}^c(\mathbf{x}, \boldsymbol{\theta})$ denote the response vector from the computer model at input $\mathbf{x}$ with $\boldsymbol{\theta} \in \Theta \subseteq \mathbb{R}^p$, which is a set of calibration parameters. The goal is to identify the parameter value $\boldsymbol{\theta}$ that minimizes the difference between the real physical process $\mathbf{y}(\mathbf{x})$ and the computer model output $\mathbf{y}^c(\mathbf{x}, \boldsymbol{\theta})$ as follows.

$$\boldsymbol{\theta}^* = \underset{\boldsymbol{\theta}\in\Theta}{\arg\min}\, L(\boldsymbol{\theta}), \tag{1}$$

where $L(\boldsymbol{\theta})$ implies a loss function. Among several loss functions, the ℓ_2 norm is widely used due to its mathematical tractability [10]. This type of problem is called *parameter calibration* in the literature [5].

More specifically, in building energy simulation, let y_i denote an observed building energy consumption record at hour index i. Let $y_i^c(\mathbf{x}, \boldsymbol{\theta})$ denote the corresponding output from the simulator at i with the hourly temperature $\mathbf{x}$ and calibration parameters $\boldsymbol{\theta}$. Based on the seasonal dependency of each parameter, we consider the three loss functions as follows.

$$L_g(\boldsymbol{\theta}) = \frac{1}{|\mathcal{I}_g|} \sum_{i\in\mathcal{I}_g} (y_i - y_i^c(\mathbf{x}, \boldsymbol{\theta}))^2 \tag{2}$$

$$L_s(\boldsymbol{\theta}) = \frac{1}{|\mathcal{I}_s|} \sum_{i\in\mathcal{I}_s} (y_i - y_i^c(\mathbf{x}, \boldsymbol{\theta}))^2 \tag{3}$$

$$L_w(\boldsymbol{\theta}) = \frac{1}{|\mathcal{I}_w|}\sum_{i\in\mathcal{I}_w}(y_i - y_i^c(\mathbf{x},\boldsymbol{\theta}))^2 \tag{4}$$

where $\mathcal{I}_g$ is an index set of the entire dataset, $\mathcal{I}_s$ is an index set for the data collected in heating season, and $\mathcal{I}_w$ is an index set associated with observations collected during cooling season. $|\cdot|$ denotes the size of the corresponding dataset.

We divide the parameter vector into three groups $\boldsymbol{\theta} = [\boldsymbol{\theta}_g^T, \boldsymbol{\theta}_s^T, \boldsymbol{\theta}_w^T]^T$. Here, $\boldsymbol{\theta}_s, \boldsymbol{\theta}_w \subset \boldsymbol{\theta}$ are the heating season and cooling season parameters, respectively. Then, we can obtain the parameters by solving the following three optimization problems with loss functions $L_g(\boldsymbol{\theta})$, $L_s(\boldsymbol{\theta})$, and $L_w(\boldsymbol{\theta})$.

$$\theta_g^* = \arg\min_{\boldsymbol{\theta}_g\subset\boldsymbol{\theta}} L_g(\boldsymbol{\theta}), \quad \theta_s^* = \arg\min_{\boldsymbol{\theta}_s\subset\boldsymbol{\theta}} L_s(\boldsymbol{\theta}), \quad \theta_w^* = \arg\min_{\boldsymbol{\theta}_w\subset\boldsymbol{\theta}} L_w(\boldsymbol{\theta}) \tag{5}$$

2.2 Solution Procedure

We design an optimization algorithm to solve the optimization problems in (5). We call the proposed algorithm the block gradient descent (BGD), as we borrow an idea from the block coordinate descent (BCD). The fundamental idea of BGD is to design a block-type gradient descent algorithm so that the parameters in different groups can be adaptively calibrated with their own objective functions over iterations. Before presenting the BGD procedure, we would like to mention the difference between BGD and BCD: each objective (loss) function of the three sub-problems in (5) does not guarantee to be convex, while each sub-problem in BCD is assumed to be convex. Indeed, our simulator is a black-box computer model, so the convexity of the three loss functions cannot be explicitly known.

Since the loss functions have no mathematical closed-form expressions due to the black box nature of the BEM simulator, the gradients cannot be explicitly computed. Thus, we use a first-order (forward) finite-difference to approximate the gradients for each loss function as follows:

$$[\nabla L_g(\boldsymbol{\theta})]_j = \frac{\partial L_g}{\partial \theta_{g,j}} \approx \frac{L_g(\boldsymbol{\theta}:\theta_{g,j}+h) - L_g(\boldsymbol{\theta})}{h}, \quad \forall j = 1,\ldots,|\boldsymbol{\theta}_g|, \tag{6}$$

$$[\nabla L_s(\boldsymbol{\theta})]_j = \frac{\partial L_s}{\partial \theta_{s,j}} \approx \frac{L_s(\boldsymbol{\theta}:\theta_{s,j}+h) - L_s(\boldsymbol{\theta})}{h}, \quad \forall j = 1,\ldots,|\boldsymbol{\theta}_s|, \tag{7}$$

$$[\nabla L_w(\boldsymbol{\theta})]_j = \frac{\partial L_w}{\partial \theta_{w,j}} \approx \frac{L_w(\boldsymbol{\theta}:\theta_{w,j}+h) - L_w(\boldsymbol{\theta})}{h}, \quad \forall j = 1,\ldots,|\boldsymbol{\theta}_w|, \tag{8}$$

where $|\cdot|$ is the length of each parameter vector and $h > 0$ is a small perturbation to the j^{th} parameter of each $\boldsymbol{\theta}_g, \boldsymbol{\theta}_s, \boldsymbol{\theta}_w$. Let $\boldsymbol{\theta}_g^k$ denote the iterate of $\boldsymbol{\theta}_g$ at the k^{th} iteration. With the gradient information, it can be updated by

$$\boldsymbol{\theta}_g^{k+1} = \boldsymbol{\theta}_g^k - \alpha_g \nabla L_g(\boldsymbol{\theta}^k), \tag{9}$$

with α being a step size. Likewise, $\boldsymbol{\theta}_s^k$ and $\boldsymbol{\theta}_w^k$ can be updated in a similar manner.

The BGD algorithm works by cyclically optimizing one block of parameters each time while keeping other blocks fixed. Specifically, it iteratively optimizes $\boldsymbol{\theta}_g$ (first) with other parameters fixed, and it optimizes $\boldsymbol{\theta}_s$ (second) with the previously optimized $\boldsymbol{\theta}_g$, and then it optimizes $\boldsymbol{\theta}_w$ (third) with the previously optimized $\boldsymbol{\theta}_g$ and $\boldsymbol{\theta}_s$ until the stopping criteria are satisfied. Algorithm 1 summarizes the BGD algorithm. The stopping criteria can be set as the relative difference of function values is less than a small tolerance ϵ, e.g.,:

$$\max\{|L_g(\boldsymbol{\theta}^{k+1}) - L_g(\boldsymbol{\theta}^k)|/|L_g(\boldsymbol{\theta}^k)|, |L_s(\boldsymbol{\theta}^{k+1}) - L_s(\boldsymbol{\theta}^k)|/|L_s(\boldsymbol{\theta}^k)|, |L_w(\boldsymbol{\theta}^{k+1}) - L_w(\boldsymbol{\theta}^k)|/|L_w(\boldsymbol{\theta}^k)|\} < \epsilon \tag{10}$$

or the maximum number of iteration is larger than some value.
To adaptively choose the step size, we use the backtracking line search for each inner loop. The procedure is as follows. We first fix a constant parameter $0 < \beta < 1$ and $0 < c \leq 1/2$, then for each iteration, we start with $\alpha = 1$ and find the step size α by updating $\alpha \leftarrow \beta\alpha$ while satisfying the following condition:

$$L(\boldsymbol{\theta} - \alpha\nabla L(\boldsymbol{\theta})) > L(\boldsymbol{\theta}) - c\alpha||\nabla L(\boldsymbol{\theta})||_2^2. \tag{11}$$

Thus, we can identify an appropriate step size α to ensure the sufficient decrease of the objective functions.

Algorithm 1 Block Gradient Descent (BGD)

Input: $\mathbf{y} = \mathbf{y}(\mathbf{x}), \mathbf{y}_s = \mathbf{y}_s(\mathbf{x}), \mathbf{y}_w = \mathbf{y}_w(\mathbf{x})$
Initialization: Randomly choose $\boldsymbol{\theta}^0 = (\boldsymbol{\theta}_g^0, \boldsymbol{\theta}_s^0, \boldsymbol{\theta}_w^0)$. Set $k = 0$.
while convergence criterion not met **do**
  while convergence criterion not met **do**
    Set α_g using backtracking line search and approximate gradient $G_g^k = \nabla L_g(\boldsymbol{\theta}_g^k, \boldsymbol{\theta}_s^k, \boldsymbol{\theta}_w^k)$
    Update $\boldsymbol{\theta}_g^{k+1} \leftarrow \boldsymbol{\theta}_g^k - \alpha_g^k G_g^k$
  end while
  while convergence criterion not met **do**
    Set α_s using backtracking line search and approximate gradient $G_s^k = \nabla L_s(\boldsymbol{\theta}_g^{k+1}, \boldsymbol{\theta}_s^k, \boldsymbol{\theta}_w^k)$
    Update $\boldsymbol{\theta}_s^{k+1} \leftarrow \boldsymbol{\theta}_s^k - \alpha_s^k G_s^k$
  end while
  while convergence criterion not met **do**
    Set α_w using backtracking line search and approximate gradient $G_w^k = \nabla L_w(\boldsymbol{\theta}_g^{k+1}, \boldsymbol{\theta}_s^{k+1}, \boldsymbol{\theta}_w^k)$
    Update $\boldsymbol{\theta}_w^{k+1} \leftarrow \boldsymbol{\theta}_w^k - \alpha_w^k G_w^k$
  end while
  Let $k \leftarrow k + 1$.
end while

2.3 Multi-thread computing

To accelerate the BGD algorithm, we implement the gradient approximation and line search with multi-threads programming. Consider the global parameter updating. In the gradient approximation, the j^{th} thread will run a simulator and approximate the gradient of $\theta_{g,i}$. Similarly, in the backtracking line search, instead of finding α in (11) sequentially, we run a simulator with $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \beta^t \nabla L(\boldsymbol{\theta})$ in the t^{th} thread, compute and save the corresponding value of the objective function. Specifically, since we run the line search concurrently with multi-threads, we find the best step size for $\boldsymbol{\theta}_g$, for example, as follows.

$$\alpha_g = \beta^{t^*}, \tag{12}$$

where

$$t^* = \underset{t \in \{1,2,\ldots,N_T\}}{\arg\min} \{L_g(\boldsymbol{\theta} : \boldsymbol{\theta}_g - \beta^t \nabla L_g(\boldsymbol{\theta}))\} \tag{13}$$

with N_T available threads. By running the simulator with multiple threads in parallel, we approximate the gradients of all parameters at the same time and try different step sizes concurrently.

3 Implementation Results

In this section, we show the performance of our proposed seasonal-dependent calibration method. We use the EnergyPlus 9.3.0 [11], developed by the U.S. Department of Energy's National Renewable Energy Laboratory. We use the year-long energy consumption dataset collected from an actual building located at Mueller, Austin in Texas.

3.1 Parameter Selection

Before performing a parameter calibration, we first determine a subset of the parameters among hundreds of parameters in EnergyPlus. This step is important because we need to run simulation in a manageable time. Thus, we should carefully choose parameters that affects model outputs the most. According to the previous studies in [8, 9], we choose parameters from the following aspects: lighting, ventilation, DHW, window material (optical properties), heating and cooling system. Finally, we divide the parameters into three groups: global parameters $\boldsymbol{\theta}_g$ (i.e., the parameters that are used throughout a year), heating season parameters $\boldsymbol{\theta}_s$ (i.e., the parameters that are used in heating season) and cooling season parameters $\boldsymbol{\theta}_w$ (i.e., the parameters that are used in cooling season). Table 1 summarizes the parameters and their schedule in Texas.

Table 1: Parameter selection

Parameter group	Parameter	Description	Schedule (month)
global parameter $\boldsymbol{\theta}_g$	θ_1	solar transmittance	January through December
	θ_2	solar reflectance	
	θ_3	lighting level	
	θ_4	ceiling fan design level	
	θ_9	cooling supply air flow rate	
	θ_{10}	heating supply air flow rate	
	θ_{11}	maximum supply air temperature	
	θ_{12}	heater thermal efficiency	
	θ_{13}	fan total efficiency	
	θ_{14}	ventilation design flow rate	
heating season parameter $\boldsymbol{\theta}_s$	θ_5	gross rated cooling COP	March through November
	θ_6	furnace heating coil nominal capacity	
cooling season parameter $\boldsymbol{\theta}_w$	θ_7	gross rated total cooling capacity	January through April, November, December
	θ_8	burner efficiency	

3.2 Parameter Setting

In Algorithm 1, we set a tolerance $\epsilon = 10^{-4}$ and a small perturbation $h = 10^{-4}$. We also set the maximum number of iteration to be 300 for each inner loop and 300 for an outer loop. For the multi-thread computing, we use 14 available threads. In the backtracking line search, β is usually set as 0.9 or 0.8. However, we get $0.9^{14} \approx 0.22877$ and $0.8^{14} \approx 0.04398$, which means the search ranges are unduly narrow. Therefore, in our implementation, we set $\beta = 0.6$ to have $0.6^{14} \approx 7.83642 \times 10^{-4}$. We consider this search range is wide enough to cover an appropriate step size.

3.3 Implementation Result

We compare our season-dependent calibration procedure with the general procedure that does not account for the seasonality. For this alternative method, we employ a gradient descent (GD) and calibrate all parameters with year-long data. We evaluate the performance of the proposed BGD with GD in terms of MSE. Figure 1 shows that the parameter calibration that considers seasonal dependency (BGD) achieves a lower MSE and requires a smaller number of iterations than the one that does not (GD).

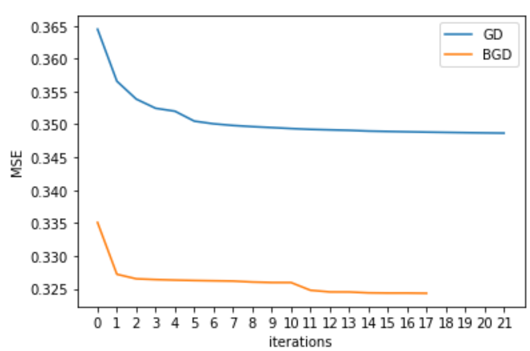

Figure 1: MSE of GD and BGD over iterations until convergence (blue line is GD and orange line is BGD)

Table 2 compares the MSEs from BGD and GD with that from the baseline setting. The baseline parameer setting was obtained from the Building America House Simulation Protocol [12]. Both BGD and GD achieve lower MSEs than the original baseline setting. Further, BGD outperforms GD, because BGD runs simulations with a schedule that considers seasonality whereas GD ignores the seasonal dependency. Specifically, BGD obtains the gradients of $\boldsymbol{\theta}_s$ and $\boldsymbol{\theta}_w$ with the respective seasonal portion of observational data and corresponding simulation outcomes, but GD uses the year-long data, which misguides the calibration direction.

Table 2: Results of BGD, GD and Baseline

	BGD	GD	Baseline
MSE	**0.32431**	0.34868	0.38341

4 Conclusion and future research

We show that the discrepancy between the actual energy consumption and the simulated energy consumption can be reduced, when we take the seasonal dependency of parameters into consideration. To demonstrate the performance of the proposed season-dependent calibration method, we design the BGD algorithm and compare with GD in terms of MSE and the number of iterations until convergence. The results show that BGD outperforms the GD procedure.

Several studies devise new approaches to improve GD-based updates in the optimization process, such as adding a momentum, an exponential moving average of all the past gradients at parameter update [13]. The algorithms with those ideas, e.g., AdaGrad, RMSProp, Adam, etc. have been successfully applied in neural networks. In the future, we plan to incorporate such algorithms into the season-dependent calibration framework. Further, as many researchers and practitioners use Bayesian calibration for the BEM parameter calibration, we will compare our approach with the Bayesian calibration method.

Acknowledgement

This work was supported by the National Science Foundation under Grants CMMI-1662553, CMMI-1662691, and CBET-1804321.

References

[1] M. C. Kennedy and A. O'Hagan, "Bayesian calibration of computer models," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol.63, pp.425-464; 2001.

[2] Q. Li, G. Augenbroe, and J. Brown, "Assessment of linear emulators in lightweight Bayesian calibration of dynamic building energy models for parameter estimation and performance prediction," *Energy and Buildings*, vol.124, pp.194-202; 2016.

[3] K. Menberg, Y. Heo, R. Choudhary, "Efficienct and Reliability of Bayesion Calibration of Energy Supply System Models," in *Proceedings of 2017 IBPSA Building Simulation Conference*, August 7-9, 2017, San Francisco, USA; 2017.

[4] A. Chong and K. P. Lam, "A Comparison of MCMC Algorithm for the Bayesian Calibration of Building energy Models," *IBPSA Building Simulation Conference*, August 7-9, 2017, San Francisco, USA; 2017.

[5] B. Liu, X. Yue, E. Byon, and R. Kontar, "Parameter Calibration in wake effect simulation model with Stochastic Gradient Descent and stratified sampling," *submitted*, 2020.

[6] J. Sun and T. A. Reddy, "Calibration of Building Energy Simulation Programs using the Analytic Optimization Approach (RP-1051)," *HVAC & R Research,* 12(1), pp.177-196; 2016.

[7] D. Coakley, P. Raftery, and M. Keane, "A review of methods of building energy simulation models to measured data," *Renewable & sustainable energy reviews*, vol.37, pp.123-141; 2014.

[8] M. Manfren, N. Aste, and R. Moshksar, "Calibration and uncertainty analysis for computer models - A meta-model based approach for integrated building energy," *Applied Energy*, vol.103, pp.627-641; 2012.

[9] A. Chong, K. P. Lam, M. Pozzi, and J. Yang, "Bayesian calibration of building energy models with large datasets," *Energy and Buildings*, vol.154, pp.343-355; 2017.

[10] R. Tuo and J. C. Wu, "Efficient Calibration for Imperfect Computer Models," *The Annals of Statistics*, vol.43, pp.2331–2352; 2015.

[11] U.S. Department Energy, National Renewable Energy Laboratory, "EneryPlus Essentials," Technical Report; 2019.

[12] E. Wilson, C. E. Metzger, S. Horowitz, and R. Hendron, "2014 Building America House Simulation Protocols," U.S. Department Energy, National Renewable Energy Laboratory, Technical Report; 2014.

[13] Ning Qian, "On the momentum term in gradient descent learning algorithms," *Neural networks: The Official Journal of the International Neural Network Society*, 12(1), pp.145–151; 1999.