Power Capacity Profile Estimation for Building Heating and Cooling in Demand-Side Management

Juan A. Gomez, Miguel F. Anjos

Abstract

This paper presents a new methodology for the estimation of power capacity profiles for smart buildings. The capacity profile can be used within a demand-side management system in order to guide the building temperature operation. It provides a trade-off between the quality of service perceived by the end user and the requirements from the grid in a demand-response context. We use a data-fitting approach and a multiclass classifier to compute the required profile to run a set of electric heating and cooling units via an admission control module. Simulation results validate the performance of the proposed methodology under various conditions, and we compare our approach with neural networks in a real-world-based scenario.

Keywords: Smart buildings, power demand, residential load sector, least squares, parameter estimation, classification.

1. Introduction

In the context of power systems, reducing peaks and the fluctuation of consumption brings stability to the system and benefits to the players in the power supply network. In this respect, demand-response (DR) programs and demand-side management (DSM) systems encourage and facilitate the participation of the end users in the grid decisions. This participation is increasing with the development and implementation of smart buildings. DR programs have mostly been oriented to large consumers, but smart buildings can exploit the DR potential in residential and commercial buildings as well. These represent around 70% of the total energy demand in the United States [1]. In Canada, space heating is responsible for more than 60% of the total residential energy consumption, due to the cold climate [2]. Across the country, electric baseboards account for 27% of heating equipment, reaching 66% in the province of Quebec. On the other hand, the province of Ontario is typically a summer-peaking region due to the high temperatures during that season and the high penetration of air-conditioning systems [3, 4].

Several authors have published DSM-related results. Normally their research motivation is oriented to load management, user behavior, cost performance, and curve shaping. Imposing a capacity constraint is a common idea among these approaches. Costanzo et al. [5] propose a multilayer architecture that provides a scheme for online operation and load control given a maximum consumption level. In the stochastic DSM program in [6], a DR aggregator imposes a capacity constraint. Bidding curves and price analyses are reported in order to guide end-users about increasing capacity. Rahim et al. [7] evaluate the performance of several heuristic-based controllers. They define the load management as a knapsack problem with preset power capacities for each time slot. In a similar way, [8] assumes a consumption limit that allows the activation of only one load at a time. Li et al. [9] look for an optimal allocation of capacities based on a queuing strategy. The service provider determines the capacity to assign to each user from a set of renewable resources.

The idea of capacity subscription is explored in [10], where the individual consumer’s demand is limited in a competitive market. On the other hand, the heuristic algorithm proposed in [11] aims to minimize the error between the actual power curve and the objective load curve by moving the shiftable loads. In this case the objective load curve can be seen as a soft constraint capacity profile. A variation of the capacity limit is presented in [12], where each individual user has a predefined budget to maximize his/her satisfaction.

All the approaches mentioned represent the capacity as a given parameter, and some of them recognize the importance of using a forecasting tool to determine its value. Estimating the user consumption is a key step in the decision-making process for users and for higher levels in the power system. Relevant publications can be found in the load-forecasting literature. Suganthi and Samuel [13] give a comprehensive review of forecasting methods from classical time series to more sophisticated machine learning tools.

Load estimation methods are classified depending on the level of aggregation of the input data: they can be
bottom-up or top-down [14]. Bottom-up models extrapolate the behavior of a larger system based on its inner elements. Top-down models make decisions from a global perspective and share them among all the subsystems.

This paper proposes an approach for the estimation of a power capacity profile that works in combination with the admission controller (AC) module presented in [5]. This profile is used to ensure enough power to meet the demand for the next planning horizon (e.g., the next day in a day-ahead DR market). This novel approach takes advantage of the structure derived from the estimation problem to compute capacity profiles efficiently and reliably. Estimating the capacity that will be necessary allows us to define a relationship between the total expected demand and the level of service the user desires while providing DR. In this scenario the user will book a variable maximum power capacity per time frame over the planning horizon, ensuring a pre-established level of service. This approach could also include external factors such as peak control and pricing policies. The motivation is that a defined power budget limits the consumption and encourages load shifting. It also facilitates the integration of differential pricing for both energy and power.

This paper is structured as follows. We describe the proposed methodology in Section 2. We give simulation results for the real-world-based scenario in Section 3, and Section 4 presents our conclusions.

2. Power Capacity Profile

Figure 1 shows the application of the AC module presented in [5]. The online algorithm in the AC has four stages. First, the space heaters and the air conditioners create requests \( r_{i,t} \) when the room temperature is out (or going out) of the thermal comfort zone. Second, the algorithm sorts all the requests from the highest to the lowest priority value; the priority value is the normalized difference between the temperature in the room and the external temperature. Third, the AC accepts the highest priority requests until the given capacity \( C_h \) is consumed; the other requests are rejected. Finally, it sends the signal \( x_{i,t} \) back to each smart load \( i \) either to run (if accepted) or to stand by for the next time step (if rejected).

Figure 2 presents a basic example of the AC operation. A smart house with two rooms, R1 and R2, is simulated over a horizon of 5 time frames. Each time frame has 10 time steps where the smart loads can send requests. Typically, a time frame would be equivalent to an hour in a realistic scenario. There is a 1.5 kW space heater in each room, and the external temperature is 5°C (Figure 2(a)).

We can see the peak reduction obtained by the AC in Figure 2(b); the end-user agrees to have a preset power capacity (dashed red line), which constrains the consumption to at most 1.5 kW. The peak of consumption, for this example 3 kW, would be attained when the two space heaters are being used at the same time step. Figures 2(c) and 2(d) show the internal temperature in each room within a certain comfort zone. In a similar way, we can see the
under time-of-use pricing conditions because the customer can profit from the cheaper time frames by reshaping the load curve while ensuring the desired QoS.

In a smart building it is possible to compute the QoS from the information provided by thermostats and smart loads connected to the AC. In the spirit of [23], we define the QoS for each time frame $h$ as follows:

$$ QoS_h = \frac{\sum_{i=1}^{S} \sum_{t=1}^{N_h} x_{i,t} \times 100\%}{100\%} \quad N_h > 0 $$

where $N_h = \sum_{i=1}^{I} \sum_{t=1}^{S} r_{i,t}$.

The accepted requests have to satisfy

$$ \sum_{i=1}^{I} x_{i,t} P_i \leq C_h \quad \forall t \in [1, 2, \ldots, S] $$

Equation (2) indicates that the AC accepts requests until the capacity limit is reached. In the framework of this article we assume that both air-conditioning units and electric baseboard heaters have a constant level of consumption [24]. Let $C_h \in \Omega$, where $\Omega$ is a set of capacities that can work in combination with the AC and the set of loads. In other words, we do not want a capacity to operate a fractional number of loads in the time step $t$. Given that $\Omega$ is a discrete set we can define the classification problem

$$ \Phi(T_e, QoS_h) = C_h $$

that determines $C_h \in \Omega$ for a given external temperature $T_e$ and the QoS $h$ defined by the user. We solve this classification problem using a three-step approach: selection of the training set from historical data, function fitting, and final classification. We illustrate the steps in this section with a group of space heaters; Section 3 includes experimental results for both types of loads.

### 2.1. Sampling From Historical Data

The real data is obtained from the smart energy management system, which records the accepted requests, the rejections, and the evolution of the QoS over time. We simulate this historical data to understand the system dynamics and to implement a prediction model. The simulation conditions are:

- The set of heaters is composed of four identical units of 1.5 kW of consumption.
- The heat transfer is computed using the specific heat and Fourier’s law formulations implemented in [5] (see Section 3 for more details).
- The external temperature corresponds to the complete year 2013 (8760 hours) in the Montreal area [25].
• The comfort intervals for the internal temperatures are taken from the ISO 7730 standard analyzed in [26]. For an office category B the intervals are [20 – 24°C] and [23 – 26°C] for heating and cooling respectively.

• $C_h$ is randomly chosen from $\Omega = [1.5, 3.0, 4.5, 6.0]$ based on the interval of temperature; the highest capacities are not necessary during the warmer days (for example, with $T_{eh}^h = 19°C$ every value in $\Omega$ will return a $QoS_h$ near 100%, affecting the quality of the data training set and the estimation).

Figure 3: Histogram of hourly external temperatures in Montreal, Canada for 2013.

Figure 4: Graph of $QoS$ vs. temperature for the sampled historical data.

where $\hat{QoS}_h$ is the quality of service from the prediction model at time frame $h$.

Additionally, we will compare two different optimality criteria: the least squares value (LSV) and the least absolute value (LAV). Typically, the LSV gives more weight to distant points while the LAV is resistant to outliers [28].

The optimization problems are:

\[
\begin{align*}
\min_{\beta_1, \beta_2, \beta_3, \beta_4} & \sum_{h=1}^{H} (QoS_h - \hat{QoS}_h)^2 \\
\min_{\beta_1, \beta_2, \beta_3, \beta_4} & \sum_{h=1}^{H} |QoS_h - \hat{QoS}_h|
\end{align*}
\]

Figure 5 shows the results for a least-squares fitting of a sigmoid function.

2.2. Data Fitting

Once we have identified these features in the data set we can solve an optimization problem for the capacity estimation. We fit the sigmoid function

\[
\hat{QoS}_h = \frac{\beta_1}{1 + e^{\beta_2 T_{eh}^h + \beta_3 C_h + \beta_4}},
\]

Once we have solved the optimization problem [5] or [6] we can use (4) to compute the expected required capacity for the desired $QoS$. 

Figure 5: Fitted sigmoid function.
2.3. Motivation for Using a Sigmoid Function

The selection of a sigmoid function has both a graphical justifi cation and an interesting background. We provide intuition into why it works for the heating case; the cooling case is similar. This analysis applies to any external temperature regardless of the time frame where it occurs; therefore we omit the subscript \( h \) and use \( N \) in the place of \( N_h \) to increase readability.

We make the following assumptions:

- If \( T^{e'} < T^e \) then \( N(T^{e'}) > N(T^e) \) for any temperatures \( T^{e'} \) and \( T^e \).
- \( C \in [C_{\text{min}}, \infty) \) where \( C_{\text{min}} = \max(P_i) \).
- Each load generates at most one request per time step, and therefore the maximum number of requests per step equals \( I \).
- There exists a temperature \( T^e \) at which all the heaters generate requests at every time step, and therefore \( N(T^e) = I \times S \).

Considering the worst-case scenario for any time frame in Equations (1) and (2), we have:

\[
\text{QoS}(\hat{T}^e, C_{\text{min}}) = \frac{\sum_{i=1}^{I} \sum_{t=1}^{S} x_{i,t}}{I \times S} \quad \text{for all } \omega \in \Omega
\]

Equation (3) allows us to accept at least one request at every time step. Therefore, the total number of accepted requests satisfies:

\[
\sum_{t=1}^{S} \sum_{i=1}^{I} x_{i,t} \geq S.
\]

After substituting (9) into (7) we can obtain a minimum QoS:

\[
\text{QoS}(T^e, C_{\text{min}}) \geq \frac{1}{I} \quad \text{(10)}
\]

We can see similar behavior for scenarios with temperature \( T^e > T^{e'} \) and \( N(T^{e'}) < N(T^e) \). Let \( F \) be the minimum number of time steps where requests are received. Since each load \( i \) will request at most once per time step, we have:

\[
F = \left\lceil \frac{N(T^e)}{I} \right\rceil - \frac{N(T^e)}{I} + \alpha, \quad 1 > \alpha \geq 0 \quad \text{(11)}
\]

The variable \( F \) also becomes the minimum number of accepted requests due to the \( C_{\text{min}} \) in Equation (8). By substituting (11) into (8) we obtain:

\[
\text{QoS}(T^e, C_{\text{min}}) = \frac{\sum_{i=1}^{I} \sum_{t=1}^{S} x_{i,t}}{N(T^e)} \geq \frac{F}{(F - \alpha)I}. \quad \text{(12)}
\]

When \( \alpha = 0 \) we get the same condition as in Equation (10). A sigmoid function helps to represent the asymptotic extremes and monotonic behavior of the QoS. In the first case, we see how the QoS is bounded below in Equations (10) and (12), and it is bounded above by definition (QoS \( \leq 100 \)). In the second case, the temperature and requests are inversely proportional (if \( T^{e'} < T^e \) then \( N(T^{e'}) > N(T^e) \)), so QoS \((T^{e'})\) is monotonically increasing. Using a linear function would capture the monotonic condition but not the asymptotic extremes.

For cooling systems we would change the first assumption to \( T^{e'} > T^e \), giving \( N(T^{e'}) > N(T^e) \). This leads to a similar monotonically decreasing sigmoid function over the interval of external temperature where cooling is required.

2.4. Classification

As stated previously, we have a discrete set of capacities that are suitable for the performance of the system. We solve for \( C_h \) in (4) in order to compute the continuous signal \( \hat{C}_h \). Finally, we use the multiclass classifier

\[
C_h = \arg \min_{\omega \in \Omega} | \hat{C}_h - \omega | \quad \text{(13)}
\]

to find the required capacity.

Figure 6 shows the effect of the classifier; it assigns areas to each of the capacities based on the midpoints for each pair of sigmoid curves from Figure 5.

![Figure 6: Classification areas.](image-url)

3. Experimental Results

In the previous section we introduced the methodology with an example for a given set of homogeneous space heaters. In this section we carry out several experiments to assess and validate the performance of the proposed methodology under different conditions.

It is important to ensure coherence in the thermal system when defining the set of loads. The loads must keep the temperature in the comfort range during the warmest and coldest time frames in the data sets. This design step must include the specific features of the building such as
size, surfaces in contact with external temperatures, wall insulation materials, and thermal load inside the room. A poorly balanced thermal system could lead to a QoS of 100% with temperatures far from the comfort zone.

At the end of each time step, we compute the temperature in the rooms using the same thermal equations as in [5]:

\[
\frac{dQ_{\text{tot}}}{dT_{\text{room}}} = m_{\text{room}}C_{\text{room}},
\]

(14)

\[
\frac{dQ_{\text{exch}}}{dt} = -K_{\text{wall}} \frac{A}{\chi} (T^e - T^{\text{room}}),
\]

(15)

\[
Q_{\text{tot}} = Q_{\text{exch}} + \eta P_i,
\]

(16)

where \(K_{\text{wall}} = 4.8 \times 10^{-4} \text{ kW/m} \cdot \text{°C}\) is the average thermal conductivity of the wall, and \(\eta = 100\%\) is the efficiency of the loads. We choose a room size of 60 m\(^3\), which corresponds to an air mass of \(m_{\text{room}} = 72 \text{ kg}\) with a specific heat capacity \(C_{\text{room}} = 1.0 \text{ kJ/kg} \cdot \text{°C}\). The surface area in contact with the external temperature is \(A = 12 \text{ m}^2\) with a thickness of \(\chi = 0.2 \text{ m}\). This remains constant for all the experiments.

For a more realistic scenario both types of loads are managed by the AC; the space heaters and the air conditioners will create requests when the temperature in each room is moving out of the comfort zone.

The experiments include:

- Sets \(\mathcal{P}\) with homogeneous and heterogeneous power \(P_i\) values.
- Three different types of \(\Omega\) sets: computed from all possible combinations of values in \(\mathcal{P}\); computed from some of the combinations in \(\mathcal{P}\); and given by an external entity.
- Two fitted functions.
- Two optimality criteria: LAV and LSV.
- Comparison with two neural networks (NNs) with different topologies.

The experiments are carried out in two stages. In the training stage we reproduce the approach presented in Section 2 in order to determine the classification areas. Then in the test stage we use the classification areas to estimate the capacity profiles for given levels of the QoS. When the profiles have been computed, we run a simulation to verify the actual QoS performance.

In Sections 3.1 and 3.2 we illustrate the methodology on a three-load instance: an apartment with three rooms. In Section 3.3 we report results for an instance with 50 loads to demonstrate the scalability of our methodology.

### 3.1. Training for Three-Load Instance

We required two training sets: one for heaters and one for air conditioners. Each training set is defined over the corresponding interval of temperature \((T_{\text{h}}^c \leq 20 \text{ °C}\) for heaters and \(T_{\text{h}}^c \geq 26 \text{ °C}\) for air conditioners) and randomly chosen as in Subsection 2.1. The historical sets are simulated using the hourly temperature in Montreal for the year 2013 (8760 data points).

As mentioned before, we will compare this methodology with two other approaches. In the first case, we use the polynomial function

\[
\widehat{Q}_{\text{oS}} = \beta_1 + \beta_2 T_{\text{h}}^c + \beta_3 T_{\text{h}}^s + \beta_4 T_{\text{h}}^c C_{\text{h}}
\]

(17)

in the fitting step. A priori the sigmoid function gives a better representation of the historical set due to its monotonically increasing behavior and the asymptotic extremes. The function in Equation (17) captures only the monotonous condition. To fit each function we solve a nonlinear optimization problem using the BFGS method; it finds a solution in a few seconds.

We use NNs, which are widely used in many different types of problems, as a second benchmark. For classification problems the NN typically has the same number of neurons in the output layer as the number of classes. The NN computes the probability that each input belongs to each class, and it chooses the class with maximum probability. We implemented two NNs with \(A = 1\) and \(B = 2\) hidden layers (5 neurons each), cross entropy as a performance measure in the learning process, and a validation subset of 30\% of the points. The training time of the NNs varies between 10 and 20 seconds using scaled conjugate gradient backpropagation.

Finally, the total confusion or misclassification index measures the performance of each approach. It indicates the percentage of the total set of data that was incorrectly classified.

Tables 1 and 2 show the training results for the different scenarios and approaches. Scenarios 1–7 and 8–14 correspond to heating and cooling respectively. In scenarios 1–3 and 8–10 the loads are homogeneous and the \(\Omega\) set corresponds to all possible combinations of the loads. In scenarios 4–6 and 11–13, both homogeneous and heterogeneous loads are tested with a \(\Omega\) set that was defined separately from the loads. Finally, scenarios 7 and 14 contain a heterogeneous set of loads and all possible combinations in \(\Omega\).

In general, we observe a better performance in the sigmoid fitting (SLAV and SLSV) than in the polynomial cases (PLAV and PLSV). There is no clear difference in terms of the fitting criterion. The sigmoid function seems to be competitive with both NNs in the first six scenarios of each table.

As stated before, the sigmoid function provides a better representation of the structure of the problem. Figure 7 shows the classification areas obtained by fitting the sigmoid and polynomial functions for scenario 2. For a
Table 1: Confusion (%) in training stage for the heating scenarios

<table>
<thead>
<tr>
<th>Scenario</th>
<th>$P$</th>
<th>$\Omega$</th>
<th>PLAV</th>
<th>PLSV</th>
<th>SLAV</th>
<th>SLSQ</th>
<th>$NN_A$</th>
<th>$NN_B$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>[1.5,1.5,1.5]</td>
<td>[1.5,3.0,4.5]</td>
<td>28.31</td>
<td>33.92</td>
<td>12.54</td>
<td>13.12</td>
<td>11.25</td>
<td>10.15</td>
</tr>
<tr>
<td>2</td>
<td>[2.0,2.0,2.0]</td>
<td>[2.0,4.0,6.0]</td>
<td>18.94</td>
<td>20.38</td>
<td>13.48</td>
<td>11.70</td>
<td>15.76</td>
<td>10.10</td>
</tr>
<tr>
<td>3</td>
<td>[2.5,2.5,2.5]</td>
<td>[2.5,5.0,7.5]</td>
<td>20.58</td>
<td>25.00</td>
<td>20.92</td>
<td>17.95</td>
<td>15.70</td>
<td>10.88</td>
</tr>
<tr>
<td>4</td>
<td>[1.5,1.5,1.5]</td>
<td>[2.5,4.0,6.0]</td>
<td>32.04</td>
<td>28.50</td>
<td>10.01</td>
<td>14.47</td>
<td>16.2</td>
<td>14.25</td>
</tr>
<tr>
<td>5</td>
<td>[2.0,2.0,2.0]</td>
<td>[2.5,4.0,6.0]</td>
<td>25.67</td>
<td>22.01</td>
<td>12.54</td>
<td>14.47</td>
<td>17.56</td>
<td>14.25</td>
</tr>
<tr>
<td>6</td>
<td>[2.5,2.0,1.5]</td>
<td>[2.5,4.0,6.0]</td>
<td>21.46</td>
<td>20.21</td>
<td>7.01</td>
<td>10.51</td>
<td>6.75</td>
<td>7.25</td>
</tr>
<tr>
<td>7</td>
<td>[2.5,2.0,1.5]</td>
<td>[2.5,3.5,4.0,4.5,6.0]</td>
<td>45.63</td>
<td>49.21</td>
<td>34.96</td>
<td>45.38</td>
<td>27.69</td>
<td>25.01</td>
</tr>
</tbody>
</table>

Table 2: Confusion (%) in training stage for the cooling scenarios

<table>
<thead>
<tr>
<th>Scenario</th>
<th>$P$</th>
<th>$\Omega$</th>
<th>PLAV</th>
<th>PLSV</th>
<th>SLAV</th>
<th>SLSV</th>
<th>$NN_A$</th>
<th>$NN_B$</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>[0.5,0.5,0.5]</td>
<td>[0.5,1.0,1.5]</td>
<td>30.15</td>
<td>33.23</td>
<td>12.26</td>
<td>12.73</td>
<td>16.11</td>
<td>17.16</td>
</tr>
<tr>
<td>9</td>
<td>[1.0,1.0,1.0]</td>
<td>[1.0,2.0,3.0]</td>
<td>18.28</td>
<td>19.33</td>
<td>10.63</td>
<td>9.33</td>
<td>19.42</td>
<td>8.35</td>
</tr>
<tr>
<td>10</td>
<td>[1.5,1.5,1.5]</td>
<td>[1.5,3.0,4.5]</td>
<td>23.40</td>
<td>25.75</td>
<td>21.61</td>
<td>18.00</td>
<td>13.54</td>
<td>13.13</td>
</tr>
<tr>
<td>11</td>
<td>[0.5,0.5,0.5]</td>
<td>[1.5,2.0,3.0]</td>
<td>31.32</td>
<td>36.23</td>
<td>12.69</td>
<td>19.42</td>
<td>9.57</td>
<td>9.14</td>
</tr>
<tr>
<td>12</td>
<td>[1.0,1.0,1.0]</td>
<td>[1.5,2.0,3.0]</td>
<td>22.08</td>
<td>22.44</td>
<td>13.96</td>
<td>17.34</td>
<td>5.36</td>
<td>12.40</td>
</tr>
<tr>
<td>13</td>
<td>[1.5,1.0,0.5]</td>
<td>[1.5,2.0,3.0]</td>
<td>22.18</td>
<td>24.24</td>
<td>8.03</td>
<td>11.17</td>
<td>22.71</td>
<td>13.19</td>
</tr>
<tr>
<td>14</td>
<td>[1.5,1.0,0.5]</td>
<td>[1.5,2.0,2.5,3.0]</td>
<td>46.53</td>
<td>46.84</td>
<td>39.19</td>
<td>44.61</td>
<td>26.84</td>
<td>23.38</td>
</tr>
</tbody>
</table>

Figure 7: Comparison of sigmoid and polynomial areas for scenario 2.

QoS of 90%, we see that the polynomial function gives a transition between areas either before or after the sigmoid function. If it is before, $T \in (-18, -8)\degree C$, we will obtain a worse QoS and lower temperatures in the rooms. If it is after, $T \in (2, 8)\degree C$, we will have extra capacity that is not required. This lower utilization of the capacity becomes more important if the user is paying in advance for a renewable source that will not be used.

On the other hand, scenarios 7 and 14 are significantly different: the NNs have considerably better performance than any other approach. Looking deeper into the characteristics of these scenarios we see a special condition: several values in $\Omega$ can generate the same QoS at the same temperature. We may have the same performance in scenario 7 for $\omega = 4$ and $\omega = 4.5$ if the three heaters send requests at the same time. In the first case, the AC will accept $P_1$ and $P_3$ and leave $P_2$ for the next time step. In the second case the order of acceptance changes but the QoS is the same. Figure 8 shows the training set for this scenario; we can see how $C = 4$ is distributed over its adjacent classes.

Although the NNs have a better training performance, they might minimize the confusion value by eliminating one of the classes. Let $W_\omega$ be the set of points that belong to class $\omega$, and let $W_1^\omega$ and $W_0^\omega$ be the subsets of points correctly and incorrectly classified respectively. Let $\Gamma$ be the total number of misclassified points. The approach presented in this article separates any two contiguous sets following the fitted function, and therefore $W_1^\omega + W_0^\omega = |W_1^\omega| + |W_2^\omega| = W_2^\omega$, and $W_1^\omega + W_2^\omega = \Gamma$.
If we assume that the NN eliminates class 2 we have \( W_1^1 = | W_1 |, \ W_2^0 = | W_2 |, \) and \( W_1^2 + W_2^0 = | W_2 | = \Gamma \). We can conclude that eliminating one class improves the confusion (i.e., \( \Gamma < \Gamma \)) if \( W_1^2 < W_2^0 \).

At this point we can see the advantage of exploiting the features of the problem. In the approach presented in this paper the fitted function acts as a constraint that represents the structure of the data sets. On the other hand, the flexibility of the NNs allows a lower misclassification, but we see in Subsection 3.3 that this has an unexpected impact on the QoS.

### 3.2. Results for Three-Load Instance

The experiments use data for a period of two years (2014 and 2015) for the Montreal area (17,520 data points). The user sets a QoS of 90%. Figures 9–14 show the results for scenarios 2 and 7 (heating) and scenario 14 (cooling). These box plots contain the minimum value, maximum value, and interquartile range for the hourly QoS and the hourly average temperature in the three rooms for each of the methods compared.

For scenario 2 (Figures 9 and 10), we see that the sigmoid and NN cases perform slightly better than the polynomial function. Although the QoS and the temperature do not vary significantly, the use of the resource differs: the polynomial function reports around 60% of utilization of capacity while the other four methods achieve a utilization between 70% and 75%. This effect was previously observed in Figure 7. Scenarios 1, 3 to 6, and 8 to 13 have similar results.

In the case of scenarios 7 and 14 we observe a special situation: although the training results for the NNs are better we have a worse QoS (Figures 11 and 12) and temperature management (Figures 13 and 14). We previously saw in Figure 8 that the areas for classes 3.5, 4, and 4.5 are not clearly defined. We also saw that different capacities can result in a similar QoS at the same temperature due to the load shifting. Nevertheless, eliminating one of the classes can have negative effects on the final output;
in this case the NNs tend to eliminate class 4.5 in order to minimize the confusion value. Although $C = 4.5$ and $C = 4.0$ can accept two out of the three loads if all of them arrive at the same time, the situation changes when the loads arrive at different times. For example, $C = 4.5$ will satisfy any of the combinations of two loads arriving simultaneously: $[2.0, 2.5], [1.5, 2.5]$, and $[1.5, 2.0]$, whereas $C = 4.0$ will not accept $[2.0, 2.5]$. It is therefore preferable not to eliminate a class because of the dynamics in the system.

3.3. Results for Fifty-Load Instance

To demonstrate the scalability of the proposed methodology, we present results for an instance with 50 space heaters. This instance represents an apartment building with three different types of heaters $P = [1.5, 2.0, 2.5]$ with respectively 20, 15, and 15 loads of each type. We consider the scenario in which the building operator chooses $\Omega = [25.0, 45.0, 70.0, 90.5]$. Figures 15 and 17 give a summary of the results.

Figure 15 shows that the classification areas have the expected sigmoid shape. Figure 16 shows that as the external temperature increases, the capacity required decreases. Finally, Figure 17 shows that the average $QoS$ and the average room temperature remain in the comfort zone.

An important feature of this novel approach is that the $QoS$ aggregates all the requests from the loads. Therefore,
4. Conclusions

Understanding the requirements of residential consumption is key to facilitating increased participation in DR programs. The methodology proposed in this paper computes a power capacity profile that meets the user’s expectations and at the same time provides information to residential power management systems. The use of the AC and the implementation of the QoS index allow us to aggregate a set of loads, simplifying the decision-making process.

The approach we have presented takes advantage of the inner structure of the defined problem, ensuring a good representation of the historical data and a reliable tool for future estimation. The shaving effect can be achieved, controlling the peak consumption, respecting the QoS, and ensuring a better utilization of the power capacity available. The proposed method computes capacity profiles for a specific comfort zone with a defined set of loads. For different configurations of the building and/or different boundary conditions, the user can easily compute the new classification areas for different scenarios and intervals of comfort. The quality of the historical data and coherence in the thermal system when defining the set of loads are key to the applicability of this method.

Future work will explore the applicability of the proposed methodology to more complex systems with different types of buildings and loads and also take into account the user behavior.

Finally, the approach presented is computationally efficient, it utilizes data that is normally available in the smart building context, and it performs well for heating and cooling, offering better performance than NNs in a real-world-based scenario.

Acknowledgments

We thank the editor and the two anonymous referees for their detailed and helpful comments on earlier versions of this paper.

This research was supported by the Canada Research Chair on Discrete Nonlinear Optimization in Engineering.

References

