# Smart Room-by-Room HVAC Scheduling for Residential Savings and Comfort

Daniel Petrov, Rakan Alseghayer, Panos K. Chrysanthis, Daniel Mossé
Department of Computer Science, University of Pittsburgh
{dpetrov, ralseghayer, panos, mosse}@cs.pitt.edu

***Abstract*—HVAC systems account for a significant amount of the energy spent in residences with forced-air systems. Moreover, HVAC is a major contributor to greenhouse gas and fine particulate matter emissions. Recently, the use of sensors, wireless, microprocessors, and the Internet of Things (IoT) have been proposed to increase energy efficiency in residential buildings by controlling the duty cycle of furnaces and air-conditioners. In this paper, we propose *ILPSS (Integer Linear Programming for Smart Scheduling)*, a novel solution for optimizing the duty cycle of the HVAC equipment and improving users' comfort by allowing users to specify comfort levels at specific times in each room of a residence. Our proposal builds on multiple-variable, linear regression model and integer linear programming and can be run on a home IoT hub. Our experimental evaluation on a real dataset shows that our proposed solution saves energy (up to 45%) and meets users' comfort needs, compared to commodity and current smart HVAC systems.**

***Index Terms*—IoT; HVAC; smart home; energy savings;**

## I. Introduction

HVAC (Heat, Ventilation, Air Conditioning) systems account for about 50% of energy spent in the over 100 million residences in the USA. In the past, many measures have been taken to address the high energy usage for space conditioning, but only recently have the use of sensors, wireless, microprocessors, and the Internet of Things (IoT) been proposed to increase energy efficiency in residential buildings.

We have also proposed an IoT solution, called *D-DUAL* [1], to optimize the duty cycle of HVAC equipment. *D-DUAL* utilizes regression techniques to predict the time needed to reach the desired temperature on time for each request. *D-DUAL* combines three scheduling principles, namely *shortest job first, elevator algorithm*, and *latest deadline*, to ensure that the target temperatures, specified in users' requests, will be reached between the time the requests were submitted and the specified deadlines. Even though it saves significant amounts of energy, compared to commodity HVAC systems, *D-DUAL* cannot guarantee *users comfort*. We define the "comfort zone" of a user through functions that define a boundary for the maximum deviations in temperature and time, from the target time and temperature in the user's request, that the user tolerates (Figure 1).

In this paper, we propose a two-dimensional *comfort zone* model, namely a temperature-temporal model, that is combined with *D-DUAL*'s time-temperature prediction model to optimize HVAC duty cycles using integer linear programming (ILP). Given the comfort zones of users, our proposed *ILPSS* (Integer Linear Programming for Smart Scheduling) IoT-based solution schedules the duty cycles of HVAC systems intelligently for energy reduction while meeting users comfort requirements for target temperature within an interval of time, on a per-room basis, in residential buildings. The time-temperature predictive model is based on multiple linear regression (MLR) that predicts the time needed to reach the desired temperature for each request. This is achieved via sensor readings in each room.

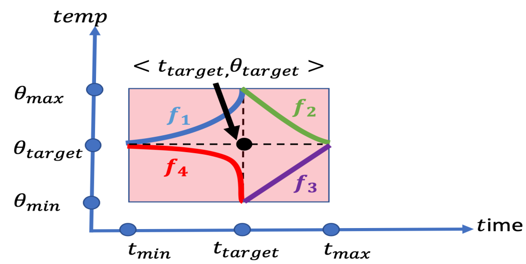


Fig. 1. Comfort Zone for a room

*ILPSS* is a lightweight computational solution that can run on a "smart" gateway in a real-life IoT "hub" and keep the computations local in the hub, avoiding exposure of users' data to privacy and security concerns.

All requests and data sensor readings from all rooms are delivered to the smart gateway, which takes all requests and sensor data for different rooms and prepares a schedule that controls the duty cycle of the HVAC system to minimize energy while maintaining users comfort.

### Contributions

- We model a user-defined comfort zone, which reflects user preference with respect to time and temperature (§III).
- We propose *ILPSS*, which combines ILP-modeled scheduling with a regression prediction model to reduce the loss of energy through scheduling the HVAC system (§III).
- We conduct an experimental evaluation on a Raspberry Pi and show that *ILPSS* (1) reduces the energy consumption for space conditioning by up to 45% compared to commodity HVAC systems, (2) maintains the temperature for each room within the comfort zone, and (3) produces results on low-cost, low-power hardware in real time within 1 second (§IV).

978-1-7281-5416-9/19/$31.00 ©2019 IEEE

## II. Models

Our solution shares the predictor with *D-DUAL* [1] and, thus, in this section we review the system model and the prediction model of *D-DUAL*. We summarize the symbols used throughout the paper in Table I.

### A. System Model

A room has direct space conditioning capabilities if there is a vent installed in the room and that vent is connected to the controller that is in charge of space conditioning (in part of) the building. Each such room is equipped with a self contained sensing unit whose measurements are denoted $\{x_{ij}\}$, whereby $i$ denotes the sample number and $j$ denotes the sensor that generated the measurement.

*Definition 1:* (*Window of measurements*) A window $w$ is a vector of $n$ consecutive measurements of the sensors, ordered in time. The oldest measurement in the window is at time $t-w$, and the most recent one is at time $t$. Each measurement contains the values from all available sensors that are fed into the thermal energy exchange function (discussed in Section III) used in the multiple linear / polynomial regression.

*Definition 2:* (*Request*) *request* $u_i$ is a tuple

$$u_i(i, t_c, \theta_{i,target}, t_{i,target}) \tag{1}$$

whereby $t_c$ is the timestamp that marks when the request was generated in room $i$, $\theta_{i,target}$ is the desired temperature to be achieved for this room, $t_{i,target}$ is a moment in the future by which the desired temperatures should be reached (the *"deadline"*).

The requests should be received long enough in advance to achieve the feasibility of satisfying them. For example, if a user submits a request for a change of the temperature in a room by 20 degrees and $t_{i,target}$ is 5 seconds from now, the request is infeasible.

*Definition 3:* (*Objective*) Given the set $Q_c$ of current sensor readings for all rooms $i$, $1 \leq i \leq m$, the previous $w$ sets of sensor readings $Q_{c-1}, Q_{c-2}, ..., Q_{c-w}$ ($p$ sensors per room) and the set $R$ of requests $u_i$, generate a schedule $S$ that optimizes the duty cycle of the furnace/AC and achieves the requested target temperatures $\theta_{i,target}$ by the users' deadlines $t_{i,target}$ for each room $i$ and maintains the users' comfort.

The schedule should be produced within 1 sec. Given the objective criterion, our solution is indifferent to the underlying infrastructure that feed data into it. It is equally suitable for IoT deployments that have wired, Wi-Fi, or ad-hoc communication networks.

### B. Prediction Model

The intuition is to predict what amount of thermally conditioned air is needed for a room to reach the temperature desired by its users. Once this information is available, it is translated into a period of time in which the vent in the room should be open and air should be blown through it. These predicted times are used to make a schedule that optimizes the duty cycle of the HVAC to reach the target temperatures in all rooms and maintains the temperatures within the comfort zones of users. The change of temperature for a room is influenced by the exchange of energy between the air in the room and the environment. We use the sensor readings to formally express as "energy exchange function $f_{ee}$", defined in [1]:

TABLE I
Energy Saving Approaches Using HVAC Scheduling

| | |
|---|---|
| $x_{i,j}$ | sensor reading |
| $p$ | number of sensors |
| $i, j$ | running counter |
| $w$ | window length |
| $u$ | user request |
| $R$ | set of requests |
| $t_c$ | current time |
| $\theta$ | temperature |
| $Q_c$ | a set of sensor readings at time $c$ |
| $m$ | number of rooms |
| $f_{ee}$ | thermal energy exchange function |
| $g$ | temperature change function |
| $\delta_t$ | period of time |
| $f$ | user-defined function |
| $cz$ | comfort zone |
| $b, c$ | vector of values |
| $A$ | matrix |
| $y$ | binary variable |
| $k$ | coefficient |
| $C$ | constant |
| $st$ | decision variable |
| $pen,\ penPlus,\ penMinus$ | "penalty" variables |

*Definition 4:* (*Thermal energy exchange*) Given $p$ sensors installed in a room to measure the factors that affect the change of the temperature in that room, the linear function:

$$f_{ee}(x_1, x_2, ..., x_p) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_p x_p \tag{2}$$

measures the *thermal energy exchange* at a given unit of time $t$. The $x_i$ values are the measurements at time $t$, read from the sensors $i$, and $\beta_i$ values are the coefficients of the function $f_{ee}$, $1 \leq i \leq p$.

Our temperature change function is defined as follows:

*Definition 5:* (*Temperature change*) Given function $f_{ee}(x_1, x_2, ..., x_p)$, the function

$$g(x_1, x_2, ..., x_p, \delta_t) = f_{ee}(x_1, x_2, ..., x_p) \times \delta_t \tag{3}$$

calculates the temperature change (in degrees) in the room, if the sensor readings $x_1, x_2, ..., x_p$ do not change for a period of time $\delta_t$.
The variables $x_i$, and $\delta_t$ are called the *explanatory variables* and $g$ is called *response variable* in the MLR–based predictor.

## III. ILPSS Solution

Our *ILPSS* solution is depicted in Figure 2. An MLR–based predictor and an ILP–based scheduling mechanism $S$, are interwoven in *ILPSS* to generate a HVAC schedule that optimizes energy consumption and keeps the temperature in all rooms within the comfort zone, as defined by user requests for each room.

In this section, the modeling of comfort zone is discussed first, followed by the principle of adversarial change of temperature in each room, caused by the environment, namely Newton's Law of Cooling. The section is concluded with an elaborate description of the scheduler.

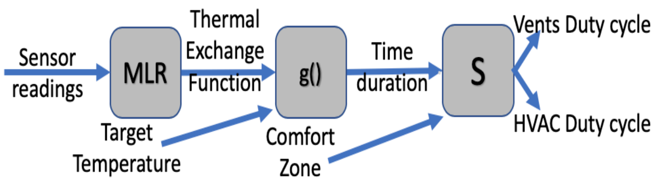


Fig. 2. ILPSS Solution with scheduler S

### A. Modeling Comfort Zone

Users' requests for a room $i$ are in the format $u_i(i, t_c, \theta_{i,target}, t_{i,target})$. Note that users will be satisfied if the appropriate temperature is reached several minutes/seconds before or after the deadline.

This tolerance in temperature and time is what defines "comfort zone", and is depicted in Figure 1. $t_{min}$ and $t_{max}$ show the earliest and latest point in time that the user tolerates reaching a temperature between $\theta_{min}$ and $\theta_{max}$. The functions $f_1$, $f_2$, $f_3$, and $f_4$ are user-defined functions of time that yield the temperature at that moment. The simplest way to model the comfort zone is to set $f_1$, $f_2$, $f_3$, and $f_4$ to be constants: comfort zone within the rectangle in Figure 1. Furthermore, users with smaller tolerance to deviations will specify a smaller comfort zone. We constrain the comfort zone to be continuous (i.e., there is no "bubble" in it). Taking a practical perspective, it does not make sense for a user to tolerate two different temperatures at the same point in time, but to have zero tolerance to a temperature in between them (i.e., the bubble). Formally:

*Definition 6:* (*Comfort Zone*) The comfort zone $cz_u$ of user $u$ is defined by:

$$cz_u =< t_{min,u}, t_{max,u}, \theta_{min,u}, \theta_{max,u}, f_{1,u}, f_{2,u}, f_{3,u}, f_{4,u} > \tag{4}$$

where $t_{min,u} \leq t_{i,target} \leq t_{max,u}$, $\theta_{min,u} \leq \theta_{i,target} \leq \theta_{max,u}$ for all requests $u_i$ and $f_{1,u}, f_{2,u}, f_{3,u}$ and $f_{4,u}$ are defined by the user $u$. We assume each user has one comfort zone for all rooms they visit. The comfort zone is modeled with Equation 5:

$$\begin{aligned} cz \;=\; & \int_{t_{min}}^{t_{i,target}} f_1(\partial t) - \int_{t_{min}}^{t_{i,target}} f_4(\partial t) \\ & + \int_{t_{i,target}}^{t_{max}} f_2(\partial t) - \int_{t_{i,target}}^{t_{max}} f_3(\partial t) \end{aligned} \tag{5}$$

### B. Newton's Law of Cooling

The surrounding environment impacts the temperature in all rooms of a building. Often the change of the temperature in rooms caused by the environment is in a direction that is opposite to the buildings' inhabitants' requests. Temperature difference between rooms that have common surface (wall, floor / ceiling) also changes the temperatures in both rooms, despite the insulation. Our prediction model captures the impact of both the environment and neighboring rooms. Note that when no thermally conditioned air is supplied for a sufficiently long amount of time, the exchange of heat between rooms will usually stop way ahead of the exchange of heat between the building and the environment. We adopted a conservative approach, whereby we consider the exchange of heat between each room and the environment when calculating divergence from the desired temperature once the delivery of thermally conditioned air for that room is over. This is calculated with Newton's Law of cooling:

*Definition 7:* (*Newton Law of Cooling*) The rate of change of temperature, $\partial\theta$, with respect to time, $\partial t$, should be proportional to the difference between the temperature of the room, $\theta$, and the ambient temperature $\theta_a$ (i.e., the temperature outside):

$$\frac{\partial\theta}{\partial t} = k(\theta - \theta_a) \tag{6}$$

$$\theta = Ce^{-kt} + \theta_a, \; when \; \theta \geq \theta_a \tag{7}$$

$$\theta = \theta_a - Ce^{-kt}, \; when \; \theta < \theta_a \tag{8}$$

By integrating both sides of Equation 6, we receive Equation 7 and Equation 8. Having the two latest temperature measurements in a room, the ambient/outside temperature and the temperature $\theta_{min,u}$, we can calculate the maximum length of the time interval, $t_{Newton,i}$ between the end of supply of thermally conditioned air for that room and $t_{min,u}$ (see Figure 1) [2]. The scheduler is discussed next.

### C. ILPSS Scheduler

The *ILPSS* scheduling algorithm (see Lines 7-11, Algorithm 1) takes two types of input, namely new sensor readings $x_{ij}$ and user requests.

When a new request is received, it is parsed in the *parseRequest()* primitive. In case several requests are received simultaneously, they are parsed sequentially. The *useCoefficientsToDeriveTime()* primitive is executed to derive the expected amount of time needed to reach the temperature for each request, given the last known sensor measurements for that room, assuming all air from HVAC goes to that room. When all predictions are in place, the ILP scheduler is run to generate a schedule, adhering to the objective of the solution, as defined in Section 3. The algorithm is depicted in Algorithm 1.

The *recalculateSchedule()* primitive is based on the *Integer Linear Programming Model* (ILP). ILP is a mathematical

**Algorithm 1** ILPSS Scheduling Algorithm

```
1:  procedure ILPSSSCHEDULER( )
2:      while 1 do
3:          if newData x_ij is available then
4:              slideWindow();
5:              recalculateRegressionCoefficients(x_ij);
6:          end if
7:          if newRequests are available then
8:              parseRequest();
9:              useCoefficientsToDeriveTime();
10:             recalculateSchedule();
11:         end if
12:     end while
13: end procedure
```

optimization problem, whereby the decision variables $st$ are restricted to be integers, and the objective function and the constraints are linear. The canonical form of ILP is as follows:

$$\begin{aligned} & maximize\ c^T st \\ & subject\ to\ Ast \leq b, \\ & st \geq 0 \\ & and\ st \in Z \end{aligned} \tag{9}$$

whereby $c$ and $b$ are vectors, $c^T$ is the transpose of $c$ and $A$ is a matrix [3], [4]. There are two algorithms used for solving ILP problems, namely the *Branch and Bound* algorithm, and the *Cutting-Plane* algorithm. The former is computationally cheaper and thus is wildly adopted into ILP libraries.

We model the scheduling problem as a classical scheduling problem. The decision variables $st_i$ define when the thermal conditioning of room $i$ starts. The predictor provides the duration $\delta_i$ of thermal conditioning needed to reach the target temperature $\theta_{i,target}$ by the deadline $t_{i,target}$. We calculate the coefficients $c_i$ as follows:

$$c_i = \frac{abs(\theta_{i,c} - \theta_{i,target})}{\delta_i} \tag{10}$$

Our objective is to minimize the function $c^T st$. The coefficients $c_i$ are the "penalty" we have to pay if we miss the deadline $t_{i,target}$. Furthermore, we define the constraints that (1) $st_i$ should not start before the length of the duration $\delta_i$ and (2) the length of $t_{Newton,i}$ should be subtracted from the deadline. This will ensure the temperature in room $i$ will be within the comfort zone. Formally:

$$st_i \geq t_{i,target} - (t_{Newton,i} + \delta_i) \tag{11}$$

Usually, residential buildings have single ducting that is shared for both heating and cooling. Thus, for our ILP-modeled scheduling, we have to ensure that cooling and heating jobs are not scheduled at the same time. We have to ensure that for each pair of rooms $i$ and $j$, whereby one needs heating and the other one needs cooling, having durations $\delta_i$ and $\delta_j$ and deadlines $t_{i,target}$ and $t_{j,target}$, respectively, the difference between the starting times $st_i$ and $st_j$ is at least $\delta_i$. The same is valid if $st_j$ precedes $st_i$. This decision structure is modeled with the introduction of a new binary variable $y_{i,j}$. It has value 1 when $st_i$ is smaller than $st_j$ and 0 otherwise. We add two additional constraints for each pair of rooms that require change of temperature in different directions:

$$\begin{aligned} M * y_{i,j} + (st_i - st_j) \geq \delta_j\ and \\ M * (1 - y_{i,j}) + (st_j - st_i) \geq \delta_i \end{aligned} \tag{12}$$

where $M$ is a very big constant, in particular $M > \delta_i \forall i$. We do not need additional pairs of constraints for rooms that either both need heating or both need cooling. The last step is to incorporate the "penalty". Let $pen_i$ be an unrestricted variable and $pen_i \geq 0$. Then

$$st_i + \delta_i + t_{Newton,i} + pen_i = t_{i,target} \tag{13}$$

When $pen_i$ is positive, the deadline is met. When it is negative, the deadline is not met. We substitute

$$pen_i = penMinus_i - penPlus_i \tag{14}$$

$$\begin{aligned} st_i + penMinus_i - penPlus_i = t_{i,target} \\ -(\delta_i + t_{Newton,i}) \end{aligned} \tag{15}$$

All variables $st_i, penMinus_i, penPlus_i$ are non-negative. The objective function becomes:

$$minimize \sum_{\forall rooms} c_i * penPlus_i \tag{16}$$

Once the ILP problem is solved, the variable $st_i$ for each room $i$ will provide the starting time for supplying thermally conditioned air to the respective room $i$.

## IV. Experiments and Analysis

In this section, we present the datasets, our testbed, the experimental evaluation results, and their analysis for our *ILPSS* solution.

### A. Experimental Framework

**Testbed** We implemented our solution and its algorithm in Java. We used the GEKKO Optimization Suite for the ILP problem [5]. We ran the experiments on a Raspberry Pi Zero W. The operating system used was Raspbian Stretch Lite.

**Metrics** We evaluated the performance of the systems in terms of *wall-clock time*, *energy decrease*, and *monetary cost*.

*Wall-Clock Time*: We measure the scalability of our solution when deployed on low cost hardware. It is measured as the amount of time it takes for the computer to run our algorithm with the number of sensor readings in our experiments.

*Energy Decrease*: This is our optimization criterion. The length of the duty cycle of HVAC can be translated to the energy spent; that is, the longer the HVAC works, the higher the amount of energy spent on space conditioning. This metric reflects how capable the solution is in thermally conditioning rooms without violating the comfort requirements of the users. The metric shows the amount of energy spent as a percentage of the amount of energy spent by a commodity system to achieve the same goal.

*Monetary Cost*: We used a real dataset that shows the fluctuation in the price of electricity in Pennsylvania in the United States, and we calculated the cost in US Dollars for each schedule produced by each of the scheduling mechanisms we tested. The price changes every five minutes, and is often higher at peak times in order to discourage usage and mitigate shortages in production.

**Datasets** *HiberSense Historical Data* [6]: The dataset we used in our experiments is the proprietary dataset we used in [1]. It consists of thousands of measurements from the HVAC related data within one family house for three days, collected between 2018-02-01 and 2018-02-03. The house has two rooms on the first floor and two rooms on the second floor. The data, available for each room, contains the measurements for motion, voltage of the sensors' battery, two different temperature measurements, humidity level, air pressure, and light level. Information about the state of the vent in each room is available as well. Vents can be either open or closed. The vents have the same sensors, except motion. Moreover,

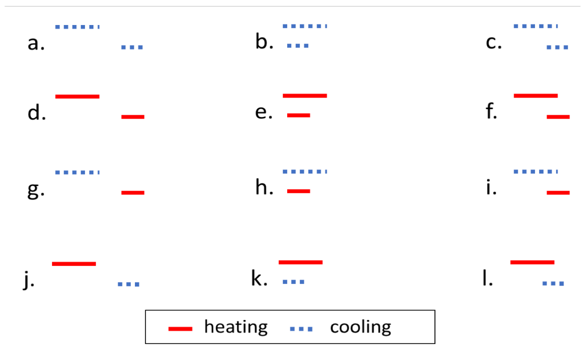


Fig. 3. Canonical cases for scheduling

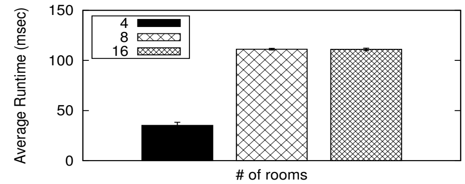


Fig. 4. Avg time to run MLR for different # of rooms for window length 8

the dataset contains the following reading for each thermostat: fan state of the HVAC (on/off), state of the HVAC (off, heat, cool), temperature set on the thermostat, override state of the thermostat, hold state, and the method the data was collected (push / pull). Outside temperature was collected once per hour. User preference levels were collected as well: the minimum and the maximum temperature the user tolerates, as well as the safety boundary temperatures, beyond which the health of the user is jeopardized. All the data is timestamped with precision within a second. The sensors in each room reported new measurements whenever there was a difference in the value of at least one reading, compared to the last values sent, or if a 15 minute time span passed. The thermostats reported every 3 seconds. For all experiments, we fed the predictor with eight consecutive measurements, called a "window". We know from our previous work that windows of length less than eight produce inaccurate results, and windows of sixteen or more do not produce accurate results either [1], [7]–[9].

We used a real dataset of fluctuating electricity prices [10]. The dataset is for 11 days (9 – 20 of June 2015), and each day has around 280 records (one every 5 minutes). The price goes negative when there is not enough demand and the providers are trying to stimulate consumption. The price is per kWh. The dataset was downloaded from the website of ComEd [10].

**Occupancy** plays role in temperature change on a per room basis. Each person emits $50W$ of power when still. This number can go to as high as $260W$ when actively exercising [11]. We incorporated the presence of humans in our energy exchange function as another sensor reading. For our experimental evaluation, we used the dataset produced by scholars at University of Texas San Antonio [12]. The dataset contains data for June 2014, for three rooms in one house, namely a kitchen, living room and bedroom. We duplicated the bedroom data from [12] to accommodate the two bedrooms in our dataset. The data granularity is one reading for each room every fifteen minutes, or 96 per day.

**Canonical Scheduling Cases** For our experimental evaluation, we adopted the canonical scheduling cases, as defined in [1]. The case when the temperature does not have to be changed implies that the target temperature is achieved already, and such rooms can be ignored. The canonical cases are summarized in Figure 3. There are 12 cases on the diagram, labeled $a$ to $l$. Each case has 2 lines, one for each room. The cases that require cooling are depicted in blue (dotted line) and the cases that require heating are depicted in red (solid line). A longer line depicted longer amount of time during which thermally conditioned air should be blown into the room. Often the rooms require different amounts of conditioned air to be provided in order to reach their desired temperatures. We depicted the case in which the request that has arrived earlier also requires more air (and thus time) to reach the goal. The opposite case, when the shorter request has an earlier arrival time, is symmetric. We did not depict it due to space limitations.

Given the intervals for two rooms there are three cases to be considered: (i) the intervals for the two rooms do not overlap, (ii) the intervals overlap completely (i.e., one contains the other one), and (iii) partial overlap, depicted in Figures 3(a), 3(b), and 3(c), respectively. In these three cases, the temperature in both rooms should be decreased. The three cases when both rooms should be heated follows in Figures 3(d), 3(e), and 3(f). The mixture of heating and cooling is depicted in Figures 3(g) to 3(l). The cases when the cooling predeceases the heating arrival is depicted in Figures 3(g), 3(h), and 3(i), and the opposite case in Figures 3(j), 3(k) and 3(l). Moreover, the temperatures in all rooms may need to be adjusted in the same direction (i.e., all need to be cooled down, or all need to be warmed up). If the temperature in all rooms is expected to be adjusted in the same direction, there is no difference from a scheduling point of view if all rooms require heating or cooling. This effectively implies that Case d is identical to Case a, Case e to Case b and Case f to Case c. The other possibility is to have a mixture of heating and cooling. Similarly, the order of arrivals for cooling and heating, when one room requires heating and the other cooling, do not make a difference from scheduling point of view. It is to be noted that Cases j, k, and l are identical to Cases g, h, and i, respectively. We derive the conclusion that there are six base cases for scheduling and we refer to them as *canonical cases*– they are depicted in Figures 3(a), 3(b), 3(c), 3(g), 3(h), and 3(i).

## *B. Experiments & Experimental Results*

We ran three experiments. The first experiment ran 5 times and reported the average and standard deviation of the metrics we collected during the experimental evaluation. The others ran only once as their results did not fluctuate between runs.

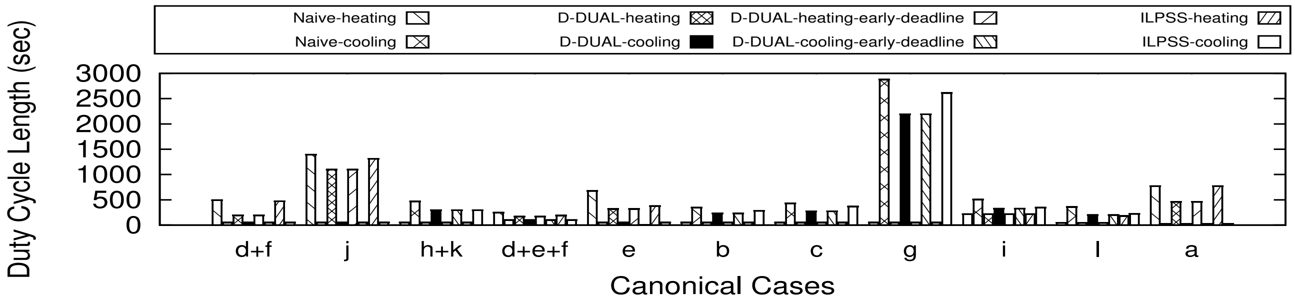


Fig. 5. Total durations of HVAC operation for *ILPSS*, *state-of-the-art* and *Naive* HVAC for all canonical cases

*Experiment 1: Scalability* (Figure 4) In our first experiment we studied how our solution scales up with an increase in the number of rooms. We measured the wall-clock time needed by our experimental testbed to run the scheduler. For a window length of 8, we set the number of rooms to 4, 8, and 16. It took less than 40 milliseconds for *ILPSS* to run scheduler for 4 rooms. When we experimented with 8 and 12 rooms, we got results that were comparable and close to 110 milliseconds. The standard deviation values are 3.08, 0.83, and 1.22 for 4, 8 and 12 rooms, respectively. The results show that we reached a plateau, whereby the increase in the number of rooms does not increase the time to run the scheduler. Moreover, the increase in time for running it between 4 and 8, rooms is expected, because the increase in the number of constraints in the ILP model is quadratic to the number of rooms, given that the number of rooms that need heating is similar to those that need cooling.

*Experiment 2: Energy Savings* (Figure 5) In this experiment we measured the energy savings caused by our *ILPSS* solution against three other approaches. Specifically, we compare it against the commodity solutions available today (we call it *Naive*), D-DUAL, and D-DUAL when run for deadlines that match the beginning of the comfort zone in time. We pick two times a day from the three days of data we have, and we derived scenarios from the dataset. Those scenarios resemble the canonical scheduling cases discussed earlier. This gave us a total of six scenarios that are diverse in the nature of their requests. In other words, each one of those scenarios consists of four requests (one for each room), that vary in the time of arrival and the cooling/heating request. We compared the performance of our *ILPSS* solution against the *Naive* approach for the predicted duration it takes the HVAC to fulfill the requests. Figure 5 shows the total time needed to regulate (i.e., cool and/or heat) the temperature in the four rooms. The results show that our *ILPSS* scheduling reduces the time needed to regulate the temperature in the four rooms by up to 45% (23% on average).

The six cases, namely (a), (d+f), (j), (h+k), (d+e+f), and (e), cover four of the canonical cases discussed earlier (see §III). For the other canonical cases, we synthesized data in order to evaluate our *ILPSS* solution. Furthermore, we did not want to combine many different canonical cases into one experiment, when possible. Thus, in most cases, whereby heating and cooling were needed, three rooms required temperature adjustment in the same direction and one room in the opposite. The only exceptions to this rule are cases (g) and (i), where it was more natural to warm up two rooms and cool down the other two. This explains why our solution mostly wins for only one of the temperature changes–there is no difference if a single room is scheduled with *Naive* or *ILPSS*.

It is to be noted that our *ILPSS* provides either less energy savings or is on-par with D-DUAL. However, the latter does not provide any guarantees that the temperature in each room will be within the comfort zone of its users. Furthermore, even when D-DUAL is given shorter deadlines, namely at the start time of the comfort zone, it provides comparable results with the same solution when it is used with the original deadlines. This shows that the produced schedule for thermal conditioning concludes early enough to not be affected by the change of deadlines. Moreover, this shows that D-DUAL reaches the desired temperatures in each room, but it does not assure that they will be within the comfort zones of users, with respect to time and temperature. There is only one case, namely (i), whereby D-DUAL and *ILPSS* show comparable results for heating - i.e. this is the case when D-DUAL will meet the comfort zone requirements.

*Experiment 3: Cost Comparison* (Figure 6) We took the schedules from the previous experiment and used the prices for three consecutive days in our energy price dataset. We calculated the monetary cost of each schedule in USD. For simplicity, we assumed that the furnace, as well as the AC unit, of the building were electric. It is realistic to assume that each of these consumes 10kW per hour. Consistently, the schedules produced by D-DUAL and our *ILPSS* solution induce lower cost, compared to *Naive*. Furthermore, it is shown again that the schedules of D-DUAL for our experiments are concluded before the beginning of the comfort zones of users. Thus, their cost are equal for all cases. In case (j) our *ILPSS* solution hit a low demand time when the price of electricity was negative. *ILPSS* induces costs that are between 7% and 138% cheaper (for case (j)) (61% on average) compared to *Naive*.

**Take Away:** In our first experiment, we found that the overhead induced by using ILP is negligible and it is the predictor that must be taken into consideration when dimensioning a system that uses *ILPSS*. In the second experiment, we found that our solution saves energy over the commodity approach

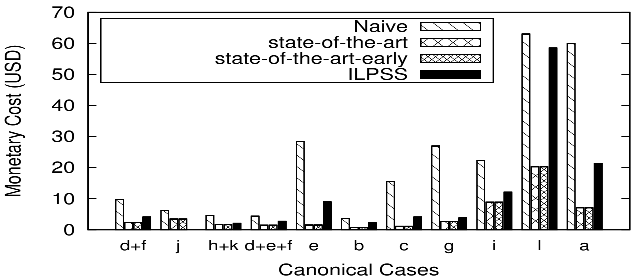


Fig. 6. Monetary cost comparison, for window length 8 for 4 rooms

and provides guarantees that the temperature will be within the comfort zone of the users. In the third experiment, we showed that our solution saves not only energy, but also money when compared to the commodity solution.

## V. Related Work

Many buildings in the US are equipped with thermostats to control the temperature and sensors to detect the presence/absence of humans. Such "smart" buildings have systems in place to control the lighting as well. Recently, buildings are built with sensors and actuators that allow fine-grain control of the temperature at the room level and the duty cycles of the lighting. Moreover, the emergence of the Internet of Things (IoT) enabled new technologies that facilitate increased autonomy in space conditioning and lightning – smartphone-based geo-fencing, as well as connected thermostats, power plugs, and light bulbs aim to improve quality of life [13]–[16].

Room-level zoning of HVAC systems is tackled in [17], [18]. The three pillars of the work are HVAC dimensioning, occupancy prediction, and thermal leakage. The first study presented in the paper argues that the HVACs installed into houses are under-dimensioned, and thus their efficiency is decreased. Smaller HVAC installations and retrofitting the buildings to room-level zoning is one of the ways to optimize energy consumption for space conditioning. Another way is improved insulation and the last one is "smart" resource allocation, whereby rooms that have no occupants are not conditioned. The occupancy detection is tackled further in [18]. Our work also tackles room-level zoning of HVAC systems, and to a certain extent we base our work on the studies in [17]. We assume that we can accurately detect room occupation, and rather than sensing the presence of occupants in rooms and using that information to control vents, we focus on HVAC duty cycle scheduling that aims to decrease energy consumption without affecting the comfort of the users.

The authors of [19], [20] used a model-predictive control mechanism that detects occupants in a room and uses that information to instruct the HVAC to compensate for the presence of people in the thermally conditioned room. The solution relies on semiparametric regression to calculate the temperature change in the room, given the presence of occupants. The energy gain they achieve is driven by the fact that the presence of people in a room increases the temperature in the room. Our work builds on top of that assumption and achieves energy savings from both an accurate *thermal energy exchange function* that can make use of occupants if such information is available, but also our novel ILP-based scheduling mechanism that optimizes the work cycles of the HVAC.

## VI. Conclusions

In this paper, we introduced *ILPSS* , an IoT solution that schedules the duty cycles of HVAC systems, that considers both energy saving and user comfort. Our *ILPSS* solution combines ILP-based scheduling and MLR prediction techniques. It is a lightweight computational solution that can be deployed on a cheap IoT gateway, and carries the computations locally to avoid data shipping, that would raise security and privacy concerns. Our experimental evaluation with real data showed that our approach achieves energy savings up to 45%, compared to the baseline commodity HVAC systems (naive), and it uses 21% more energy, compared to *D-DUAL.*

## References

[1] D. Petrov, R. Alseghayer, D. Mosse, and P. K. Chrysanthis, "Data-driven user-aware hvac scheduling," in *IGSC*, 2018.
[2] A. Galant, R. Kutner, and A. Majerowski, "Heat transfer, newton's law of cooling and the law of entropy increase simulated by the real-time computer experiment in java," ser. ICCS, 2003.
[3] H. A. Taha, *Operations Research: An Introduction (8th Edition)*, 2006.
[4] A. Schrijver, *Theory of Linear and Integer Programming*, 1986.
[5] L. Beal, D. Hill, R. Martin, and J. Hedengren, "Gekko optimization suite," *Processes*, vol. 6, no. 8, p. 106, 2018.
[6] HiberSense Inc., "Dataset," hibersense.com/, 2018.
[7] D. Petrov, R. Alseghayer, M. Sharaf, P. K. Chrysanthis, and A. Labrinidis, "Interactive exploration of correlated time series," ser. ExploreDB, 2017.
[8] R. Alseghayer, D. Petrov, P. K. Chrysanthis, M. Sharaf, and A. Labrinidis, "Detection of highly correlated live data streams," ser. BIRTE, 2017.
[9] R. Alseghayer, D. Petrov, and P. K. Chrysanthis, "Strategies for detection of correlated data streams," ser. ExploreDB, 2018.
[10] ComEd Inc., "Dataset2," https://hourlypricing.comed.com/live-prices/, 2019.
[11] T. Barker, D. C. Poole, M. L. Noble, and T. J. Barstow, "Human critical poweroxygen uptake relationship at different pedalling frequencies," *Experimental Physiology*, vol. 91, no. 3, pp. 621–632, 2006.
[12] B. Dong and K. P. Lam, "A real-time model predictive control for building heating and cooling systems based on the occupancy behavior pattern detection and local weather forecasting," in *Building Simulation*, vol. 7, no. 1, 2014, pp. 89–106.
[13] Nest Inc., "Nest Thermostats," nest.com/, 2016.
[14] K. T. e. Nguyen, "Smartphone-based geofencing to ascertain hospitalizations," *Circulation: Cardiovascular Quality and Outcomes*, vol. 10, no. 3, 2017.
[15] Philips Inc., "Philips Hue Personal Wireless Lighting," www2.meethue.com/en-us/, 2018.
[16] D-Link Inc., "D-Link Wi-Fi Smart Plug DSP-W215," us.dlink.com/products/connected-home/wi-fi-smart-plug/, 2018.
[17] T. I. Sookoor, B. Holben, and K. Whitehouse, "Feasibility of retrofitting centralized HVAC systems for room-level zoning," in *IGSC*, 2012.
[18] E. Soltanaghaei and K. Whitehouse, "Practical occupancy detection for programmable and smart thermostats," *Applied Energy*, 2017.
[19] A. Aswani, N. Master, J. Taneja, D. Culler, and C. Tomlin, "Reducing transient and steady state electricity consumption in HVAC using learning-based model-predictive control," *Proceedings of the IEEE*, vol. 100, 2012.
[20] A. Aswani, N. Master, J. Taneja, V. Smith, A. Krioukov, D. Culler, and C. Tomlin, "Identifying models of HVAC systems using semiparametric regression," in *ACC 2012*, 2012.