IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received November 24, 2020, accepted November 29, 2020, date of publication December 4, 2020,
date of current version December 16, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3042534

Optimal Control Based on Scheduling for Comfortable Smart Home Environment

SEHRISH MALIK[1], KYUTAE LEE[1], AND DOHYEUN KIM[2]
[1]Division of Information and Communication Engineering, Kongju National University, Cheonan 331717, South Korea
[2]Computer Engineering Department, Jeju National University, Jeju 63243, South Korea

Corresponding authors: KyuTae Lee (ktlee@kongju.ac.kr) and DoHyeun Kim (kimdh@jejunu.ac.kr)

This work was supported in part by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education under Grant 2020R1I1A3070744, and in part by Institute for Information and Communications Technology Planning and Evaluation (IITP) Grant funded by the Korean Government (MSIT), AutoMaTa: Autonomous Management Framework Based on Artificial Intelligent Technology for Adaptive and Disposable IoT, under Grant 2018-0-01456.

ABSTRACT Smart home environments account to a major portion of the total energy consumption in today's world. The residents of smart home environments wish to find solutions that reduce the energy costs along with providing an optimal indoor environment for the residents. Another significant aspect in smart home systems is efficiency of tasks management and control commands' execution for smart home actuators. In this paper, we propose an optimal control solution for smart home environment based on smart home energy optimization and control tasks' load dispatching and scheduling. Optimal control is achieved by first defining an objective function for minimizing energy cost which is implemented using VB-PSO (velocity boost particle swarm optimization) algorithm. Next, the control tasks are generated using rule set implemented in fuzzy logic; defined based on optimal values achieved from VB-PSO. A Markov model based mechanism dispatches control tasks at scheduler, for efficient scheduling and optimal control. The results show that the proposed optimization scheme saves up to 29.73% energy costs on average, in comparison to baseline scheme. The proposed tasks' load dispatching scheme of admission control, makes the job of load balancing among the processors efficient while giving priority to the urgent tasks. The results for scheduler evidently show the low dropping probabilities for urgent tasks along with showing 34.9% reduction in tasks' starvation rate and 36.82% reduction in average tasks' instances missing rates.

INDEX TERMS Optimal control, task scheduling, and smart home.

I. INTRODUCTION

An average household consumes 90 million BTUs of energy yearly and a major chunk of this energy is wasted based on survey conducted by U.S department of energy [1]. One of the major reasons behind energy waste are lifestyle of the residents of these households e.g. leaving the lights on while exiting rooms, forgetting to turn off appliances after use, keeping entertainment units on while no one is in room etc., all these practices end in the wastage of energy. The residential sector accounts to a major portion of the total energy consumption. In the fig. 1, we can observe the monthly energy consumption trends of the four major sectors such as industrial, transportation, residential and commercial [2]. In residential sector, energy consumption distribution can be seen at peak during the winter season. Hence, managing and optimizing the energy consumption based on peak and off-peak strategies while also considering the variation in energy costs based on time period is vital. Optimizing the indoor smart home parameters in order to reduce energy consumption; and maintaining the user comfort index inside the smart home is one of the essentials in smart homes. A typical smart home environment consists of sensor networks and actuator networks. Sensing data from sensors is passed onto the smart home system, where system performs optimization mechanisms and makes actuator control decisions based on the optimal values. The actuator control commands are then sent to the actuator network where respective actuators are controlled. The system involves two main types of tasks as sensing tasks and control tasks. The efficient scheduling of these tasks is also very significant for enhancing the smart home's system performance.

The main challenges for the energy management systems include collection of relevant data, forecasting energy

The associate editor coordinating the review of this manuscript and approving it for publication was Ton Duc Do.

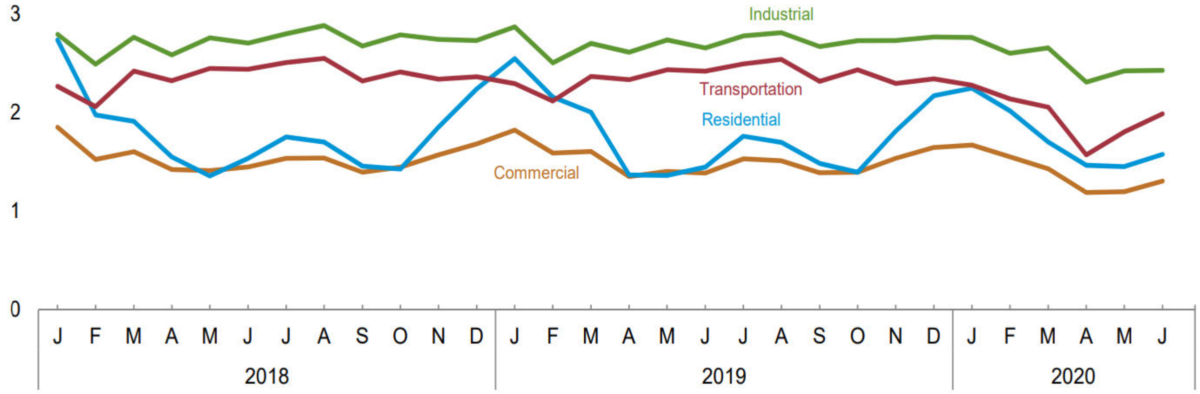


FIGURE 1. Total energy consumption by end-use sector, monthly (unit: quadrillion Btu) [2].

consumption, reducing energy costs, continuous energy consumption monitoring, maintaining user comfort index, environmental conditions based rule design, smart control of home appliances, and efficient scheduling [3], [4].

Recent years' surveys [3]–[6] on energy consumption in smart homes and smart buildings show that a lot of work has been done in the field of energy consumption forecasting, energy consumption optimization and energy consumption scheduling. Based on the recent survey results, some of the major limitations in the existing solutions are consideration of efficient load scheduling algorithms with load balancing and admission control measures, and optimal decision making in order to prioritize existing appliances load.

In this paper, we propose a scheduling mechanism based optimal control solution for smart homes. The proposed solution has two main phases. In the first phase, we design an objective function for computing optimal energy consumption for environment parameters of temperature and humidity. The optimization algorithm used to implement our designed objective function is VB-PSO. In the second phase, we focus on presenting a solution for the above-mentioned limitations of optimal decision making for optimal control, load prioritization, load balancing and admission control for efficient load scheduling. A fuzzy logic-based solution is implemented for optimal control decision making and generation of control tasks while a load dispatching algorithm using Markov model is presented for load balancing, load prioritizing and distributed load scheduling.

The rest of the paper is divided as followings. Section 2 presents the literature review and comparisons; section 3 presents the proposed scheduling based mechanism for optimal control. In section 4, we provide the input tasks modeling for simulations of the proposed system. Implementation environment is presented in section 5, results analysis is presented in section 6 and section 7 concludes the paper.

II. RELATED WORKS

Han *et al.* propose a home energy consumption and generation management system [7]. The proposed architecture monitors the smart home energy consumption of appliances and lights; and monitors the energy generation of the renewable energies. The considered renewable energies are solar and wind power. The system creates and profiles for energy consumption and energy generation and decisions are made based on the energy consumption to energy generation ratios. A multi-objective energy-saving system for residential buildings based on simulated scenarios, aims to achieve a balance between the energy savings and user comfort based on defined user constraints [8]. A multi-agent system consists of multiple agents which interact with each other to make decisions in a distributed way. In a smart home scenario, the appliances are considered as agents which work together in order to minimize the overall energy consumption and maintain a balance between user comfort, energy cost and energy savings [9].

An overview and survey of smart home energy management systems (HEMS) is presented in [10]. The review focuses on architecture and functionalities of HEMS, as well as smart homes' infrastructures, energy renewable resources and energy utilization. Strategies on energy scheduling for home appliances are also surveyed aiming at energy cost reduction and efficiency.

An energy aware smart home control system aims to reduce the packet loss due to the interventions triggered from co-existence of wireless local area network and sensor network. The system optimizes the energy consumption of smart home appliances; and consideration of natural light source while controlling the energy demand [11].

The work in [12] presents an approach based on dynamic programming for optimal water heater control in smart home by optimizing the heating schedules. The system aims at reducing energy and maintaining heat levels to the users' comfort. The work in [13], proposes a heuristic approach for efficient home energy management controller to reduce energy consumption and maximize the user comfort. A bat algorithm and fuzzy logic-based energy consumption optimization technique is proposed in [14], for user comfort management in smart buildings. The work in [15] presents an approach to optimize the energy consumption of household appliances based on retail electricity price. The work

TABLE 1. Summary of approaches proposed in related works.

Reference	Scheduling	Optimal Energy Consumption	Load Balancing	Admission Control	Minimize Tasks Starvation Rate
[7]	✓	✓	-	-	-
[8]	✓	✓	-	-	-
[9]	✓	✓	-	-	-
[11]	✓	✓	-	✓*	-
[12]	✓	✓	-	-	-
[13]	✓	✓	✓	-	-
[14]	-	✓	-	-	-
[15]	✓	✓	✓	-	-
[16]	✓	✓	✓	-	-
[17]	-	✓	-	-	-
[19]	✓	✓	✓	-	-
[20]	✓	✓	✓	-	-
[21]	✓	✓	✓	-	-
[22]	✓	✓	✓	-	-
[23]	✓	✓	✓	-	-
[24]	✓	✓	✓	-	-

presented in [16], struggles to find a trade-off between energy consumption cost and user comfort by implementing a hybrid meta-heuristic optimization technique. In [17], authors present a fusion of computational intelligence and IoT resulting into intelligent IoT based system for optimizing the smart homes' residents' comfort. The system studies the users' behavior and adjusts accordingly to improve user comfort. The work in [18] presents a survey on narratives of simple life, smart homes, smart homes comfort level expectations and definitions and contradictions. The survey aims to contribute towards better understanding of the current market picture and the areas required to be focused in order to optimize comfort level and save energy.

A comprehensive comparative analysis of optimization techniques used in smart HEMS is presented in [19]. A comparison between mathematical optimization techniques and heuristic optimization techniques is drawn with the conclusion that both techniques have own advantages and limitations. The suitable technique among the two can only be decided based on the given scenarios and assumptions.

An integrated solution based on multi-restricted scheduling scheme and grey wolf optimization algorithm for scheduling is presented in [20]. It aims at providing a cost-effective energy management solution for smart home appliances. An energy optimization solution for varying power cost scenarios is presented in [21]. An evolutionary algorithm for demand response implementation is presented in [22]. The solution balances load scheduling and optimizes energy consumption. The work in [23] presents an integrated approach smart HEMS based on the automated switching off system for load balancing and a scheduling scheme for appliances. The appliance scheduling scheme is based on least-slack time while prioritizing user comfort. Another appliance scheduling for optimal load balancing is proposed in [24], aiming to reduce energy costs and managing peak load hours. The work in [25] provides a literature survey for scheduling approaches in the smart HEMS based on power demand, pricing strategies for peak-periods and off-peak periods, electricity cost reduction, peak-to-average ratio and maximizing user comfort index.

In previous studies, we have observed that many proposed systems lack the understanding of real world smart home settings; also most of related works do not consider the combination of modules involved in the smart home functionalities and the flow of workings from one module to next, keeping in target the efficient output from the overall system when integrating the modules. Table 1 shows the summary of related works; highlighting the solutions involved in the proposed approaches. The literature review summary exposes that most of solutions focus on combination of optimal energy consumption and scheduling, while some also include load balancing. Whereas, not much attention is paid, on designing approaches with strategies for admission control and minimizing tasks starvation rates. While, reference [11] in Table 1 includes a limited tasks dropping rate reduction strategy; still it does not consider varying loads among different priority tasks. Ignoring the properties of load balancing, admission control and efficient scheduling with minimizing tasks starvation rates can result in poor execution of HEMS. Carefully designed HEMS, with all aforementioned properties can adjust to any given loads with varying priorities.

Hence, in this work we present a system that aims to schedule control tasks for optimal energy consumption of actuators in smart home environments. The system targets to bring an integrated solution which optimizes the energy consumption, generates the optimal control tasks, performs the load balancing and admission control on processors, and also makes attempt for further efficient scheduling by allocating any possible free resources to starving tasks.

III. PROPOSED MECHANISM FOR OPTIMAL CONTROL

In this section, we present the proposed scheduling based mechanism for optimal control in smart home.

A. OPTIMIZATION MECHANISM

The optimization module takes current temperature, current humidity, energy cost based on temperature and humidity values, user desired temperature ranges and user desired humidity range. The optimization module aims to find optimal temperature and humidity values based on user desired ranges and with an aim to minimize the energy cost (Fig. 2).

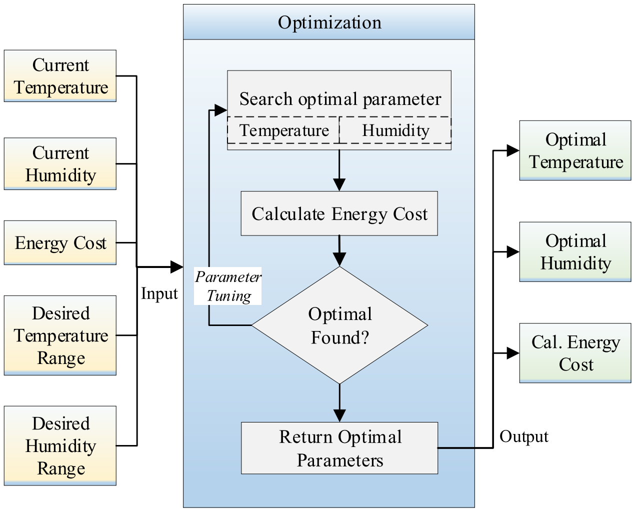


FIGURE 2. System flow for optimization process.

The aim of minimizing the energy cost with best possible (optimal) temperature and humidity values is achieved by first defining an objective function and then implementing the objective function using an optimization algorithm. In subsection 1, we present the development of defined objective function. In subsection 2 we present the optimization algorithm used for deploying objective function.

1) OBJECTIVE FUNCTION

Now, we design the objective function for the optimization process. The objective function is designed to minimize the energy consumption with optimal temperature and humidity values settings. For simplicity, we consider a simple smart home model with 2 sensors and 4 actuators. The considered sensors are temperature and humidity. The considered actuators are heater, chiller, humidifier, and dehumidifier.

TABLE 2. Parameter notations.

Notation	Parameter Description
T_C	Current temperature
H_C	Current humidity
E_{CT}	Current energy consumption units for temperature
E_{HT}	Current energy consumption units for humidity
UPT_{MD}	User preferred temperature most desired
UPT_{LD}	User Preferred Temperature least desired
UPH_{MD}	User Preferred Humidity most desired
UPH_{LD}	User Preferred Humidity least desired
DLD_{CT}	Calculated desire level difference for current temperature
DLD_{CH}	Calculated desire level difference for current humidity
T_O	Optimal temperature
H_O	Optimal humidity
ECS_O	Optimal energy consumption savings
E_O	Optimal energy consumption

The objective function will take three types of parameters as user desired parameters, current environment parameters and optimal parameters. The table 2 shows the involved parameters and used notations.

User set temperature is given in a range which specifies user's most desired temperature UPT_{MD} to user's least desired temperature UPT_{LD}. User set humidity is given in a range which specifies user's most desired humidity UPH_{MD} to user's least desired humidity UPH_{LD}. In the case of multiple users, the system will calculate the average of all the users' desired ranges and set the values of UPT_{MD}, UPT_{LD}, UPH_{MD} and UPH_{LD} based on the calculated averages. DLD_{CT} and DLD_{CH} are the calculated desire level differences between the current and user most desired temperature and humidity values in the given user parameters value ranges of [$\mathrm{UPT_{MD}}$, $\mathrm{UPT_{LD}}$] and [$\mathrm{UPH_{MD}}$, $\mathrm{UPH_{LD}}$] respectively ((1) and (2)). DLD_{COST} calculates the unit cost of energy based on the temperature and humidity values at the current iteration, during the optimization process (3). The energy consumption savings at each iteration are calculated by finding the difference between the energy cost of current set parameter values and the energy cost of desired level difference set parameters (4). Once the optimization algorithm has found the optimal parameters of temperature and humidity then the final optimal energy consumption savings are given (5). Similarly, the final optimal energy consumption can be calculated using the found optimal parameters (6).

$$\mathrm{DL}D_{CT} = |\mathrm{UPT}_{MD} - T_C| \quad T_C \in [UPT_{MD}, UPT_{LD}] \tag{1}$$

$$DLD_{CH} = |UPH_{MD} - H_C| \quad H_C \in [UPT_{MD}, UPT_{LD}] \tag{2}$$

$$DLD_{COST} = \left(DLD_{CT} * E_{cost\_T}\right) + \left(DLD_{HT} * E_{cost\_H}\right) \tag{3}$$

$$ECS_O = [E_{CT} + E_{CH}] - DLD_{COST} \tag{4}$$

$$\begin{aligned} ECS_O = {} & [E_{CT} + E_{HT}] - [(|UPT_{MD} - T_O| * E_{costT}) \\ & + (|\mathrm{UPH}_{\mathrm{MD}} - \mathrm{H}_{\mathrm{O}}| * \mathrm{E}_{\mathrm{costH}})] \end{aligned} \tag{5}$$

$$E_O = \left(T_O * E_{costT}\right) + \left(H_O * E_{costH}\right) \tag{6}$$

2) OPTIMIZATION ALGORITHM

The optimal values of temperature and humidity are achieved by deploying the above proposed objective function using optimization algorithm of VB-PSO (Velocity Boost Particle Swarm Optimization), which a variation of PSO algorithm.

In PSO, a particles' population (typically between 12 to 20), with each particle having its current position (*present*) and velocity (v), iterate the search space for optimal solution. The aim of PSO particles is to find positions (temperature and humidity values) that result in minimum calculated energy cost. Within the given range. In PSO each particle maintains local best (pbest) and global best (gbest), where pbest is particles' best values till current iteration and gbest is the population's best values till current iteration.

$$\begin{aligned} v = {} & v + c\mathbf{1} \times rand \times (pbest - present) + c\mathbf{2} \\ & \times rand \times (gbest - present) \end{aligned} \tag{7}$$

$$present = present + v \tag{8}$$

Equation (7) and (8) are used to update the velocity and position of particles in each iteration. Where, rand is a random number generated between 0 and 1, and c1, c2 are the learning factors; usually both c1 and c2 are kept 2.

In the VB-PSO, each particle's performance is maintained via a predefined threshold, and if no improvement in pbest of the particle is observed then velocity of the particle is boosted using a new inertia weight (equation (9)).

$$New\ Inertia\ Weight = c1 + \frac{Rand()}{3} \tag{9}$$

Once the global best values are found, the optimization algorithm returns the optimal values for temperature and humidity along with the calculated energy consumption using objective function.

B. FUZZY CONTROL LOGIC

Once the optimal parameters of temperature and humidity are found, then these optimal parameters are passed onto the fuzzy control module as input. At fuzzy control module, the fuzzy rules for chiller and heater are defined based on the optimal temperature and fuzzy rules for humidifier and dehumidifier are defined based on optimal humidity.

The constraints associated with the fuzzy rules set are given in the table 3. Where, T_c is current temperature, T_{min} is minimum temperature, T_{max} is maximum temperature and T_o is optimal temperature; and H_c is current humidity, H_{min} is minimum humidity, H_{max} is maximum humidity and H_o is optimal humidity.

TABLE 3. Constraints for actuators actions.

Actuator Action	Constraint
Heating	$T_c < T_{min} \leq T_o \leq T_{max}$
Chilling	$T_{min} \leq T_o \leq T_{max} < T_c$
Humidification	$H_c < H_{min} \leq H_o \leq H_{max}$
Dehumidification	$H_{min} \leq H_o \leq H_{max} < H_c$

Fuzzy control module takes the optimal values as input from optimization module and it takes the current sensing values as input from smart home environment. It then defines the antecedents and consequents for temperature and humidity values. Next, it populates the membership functions. It then defines the fuzzy rule sets of heating and cooling for temperature values and it defines the fuzzy rule sets of humidification and dehumidification for humidity values. The fuzzy rules sets are used to evaluate the fuzzy output for input values (Fig. 3). There are four actuator actions to be performed in the considered smart home environment scenario as chilling, heating, humidification and dehumidification. The output of fuzzy control module is control tasks, which represent the triggered control commands for actuators.

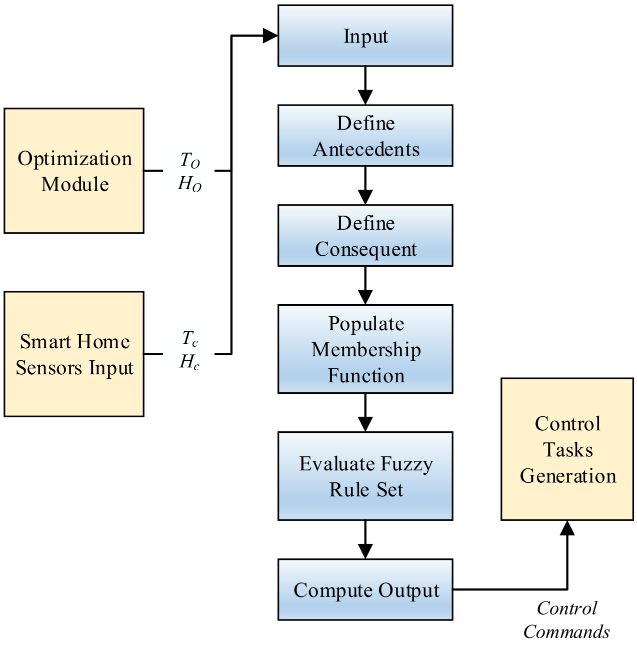


FIGURE 3. Fuzzy control flow diagram.

A task is represented as {T-ID, AT, ET, D, P, PB/U, Val} where T-ID is the unique identifier of task, AT is arrival time of task, ET is execution time of task, D is deadline of the task, P is period of the task, PB/U is priority bit or urgency factor of the task, and Val defines the starvation threshold count for the task.

C. CONTROL TASKS SCHEDULING MECHANISM

In this subsection, we present the control tasks scheduling mechanism. In fig. 4, we present the scheduling flow from the control tasks generation module. The scheduling scheme used is a combination of load balancing at processors, tasks admission control at processors and UM (urgency measure) function-based scheduling.

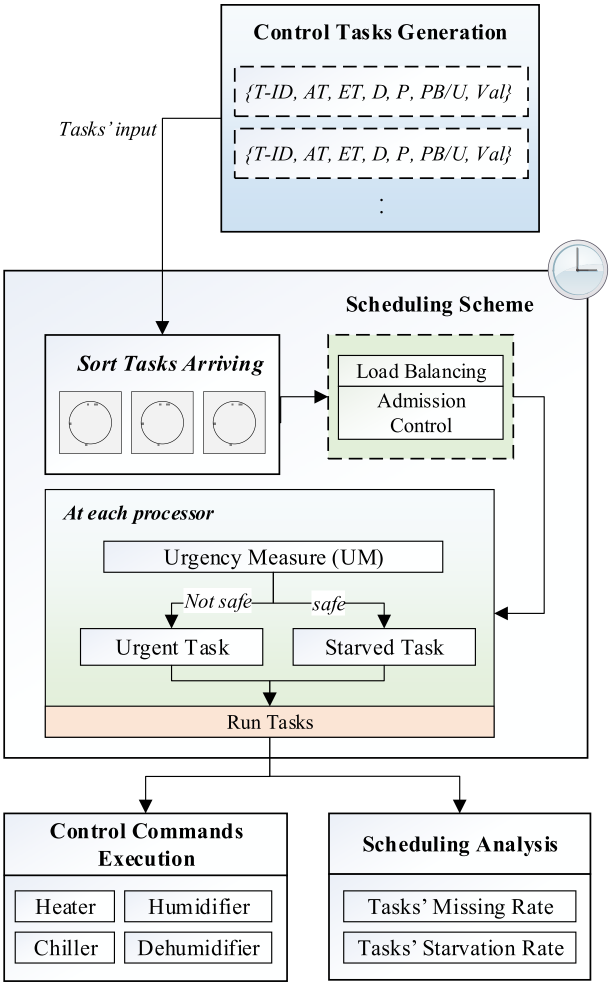


FIGURE 4. Scheduling scheme for tasks execution.

Tasks admission control is developed using one-dimensional Markov chain model [26]. The tasks admission control mechanism is presented in detail in next subsubsection. The UM function used is introduced in FEF (Fair Emergency First) scheduling algorithm [27]. The FEF scheduling algorithm is designed to maximize the machine resources and minimize the tasks starvation rate. The primary focus of FEF algorithm is to meet the tasks' deadlines based on their priorities; in parallel, saving the starving tasks by rightly utilizing any free resources. The starving tasks can be defined as any tasks which are in waiting state for a long period, due to system priorities, load or unexpected events. We have modified the FEF scheduling to current number of tasks types and their priorities.

The control tasks are categorized into two types; urgent control tasks and normal control tasks. The urgent control tasks have high priority and should be executed as they arrive; while normal control tasks have low priority as compared to urgent control tasks. The scheduler selects the processor for task dispatching based on the current tasks load among processors and admission control thresholds. Then at each processor, the tasks scheduling is performed using tasks priority while maintaining an effort to minimize tasks starving rate by implementing UM function among urgent control tasks and normal control tasks. UM function returns safe, if the execution of starving task does not risk the safe execution of urgent task, else it returns not safe.

1) LOAD BALANCING AND ADMISSION CONTROL MECHANISM

In this subsubsection, we present the admission control mechanism aiming for load balancing and admission control of tasks in the scheduling processor.

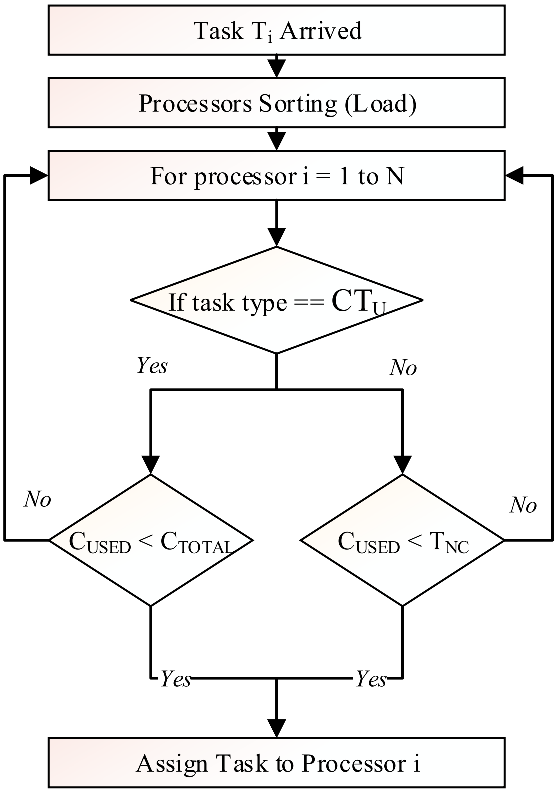


FIGURE 5. Load dispatching mechanism for admission control.

In fig. 5, we present the proposed load dispatching mechanism for admission control, with aim of load balancing based on system priorities. At first the tasks arrive at the system module of task load dispatcher. The arriving tasks are modeled and the task priority tags of normal sensing task (ST_N), urgent control tasks (CT_U) or normal control tasks (CT_N) are labeled. Next, all the existing processors are sorted based on current tasks load at each processor. The dispatching unit maintains four parameter states for each processor as shown

TABLE 4. Processor parameters at dispatching unit.

Parameter	Description
C_{TOTAL}	Processor total capacity
C_{USED}	Processor used capacity
T_{UC}	Threshold for urgent control tasks
T_{NC}	Threshold for normal control tasks

in Table 4. The threshold for urgent control tasks is considered as the total available capacity at the processor (C_{TOTAL}).

The processing capacity reserved for a type of tasks can be determined by calculating the difference between C_{TOTAL} of a processor and threshold value set for the said type of task. We have two types of tasks at the system; normal tasks and urgent tasks. Normal tasks include normal sensing task (ST_N) and normal control tasks (CT_N). Urgent tasks include urgent control tasks (CT_U).

A new arriving task is iterated through the available processors list. For each processor iteration,

- If the task arrived is CT_U and C_{USED} is less than the T_{UC}, then the task is accepted to be processed and hence dispatched to the current processor. If C_{USED} is greater or equal to T_{UC} then the task is forwarded to the next processor. We set T_{UC}equal to C_{TOTAL}.
- If the task arrived is ST_N or CT_N and the C_{USED} is less than the set value of $T_{NC,}$then the task is accepted and dispatched to the current processor; else it is blocked at this processor and passed onto next processor.

The action of load dispatching, results in two responses; normal tasks' blocking and urgent tasks' dropping. The system will have two probabilities; probability for urgent tasks' dropping rate (P_{UTD}) and probability for normal tasks' blocking rate (P_{NTB}). Where, P_{UTD} shows the dropping (missing) rate of urgent tasks at the given system state and P_{NTB} shows the blocking rate of normal tasks at the given state.

At first, we consider a scenario where no admission control for the tasks is implemented. The transition rates for the no task admission control scheme are as follows:

$$q(i, i+1) = (CT_U + CT_N + ST_N) \times (T_{NC} \le i < C_{TOTAL}) \tag{10}$$

$$q(i+1, i) = i+1 \times (T_{NC} \le i < C_{TOTAL}) \tag{11}$$

$$p_x = \frac{\frac{(CT_U+CT_N+ST_N)^x}{x!}}{\sum_{n=0}^{C_{TOTAL}} \frac{(CT_U+CT_N+ST_N)^n}{n!}} \times (0 \le x < C_{TOTAL}) \tag{12}$$

For the no task admission control scheme, the stable state probabilities of P_{UTD} and P_{NTB} are thus obtained as:

$$P_{UTD} = P_{NTB} = \frac{\frac{(CT_U+CT_N+ST_N)^{C_{TOTAL}}}{C_{TOTAL}!}}{\sum_{n=0}^{C_{TOTAL}} \frac{(CT_U+CT_{N+ST_N})^n}{n!}} \tag{13}$$

The transition rates for task admission control scheme for load balancing while prioritizing the tasks is given as follows:

$$q(i, i+1) = (CT_U + CT_N + ST_N) \quad (0 \le i < T_{NC}) \tag{14}$$

$$q(i+1, i) = i+1 \quad (0 \le i < T_{NC}) \tag{15}$$

$$q(i, i+1) = (CT_U) \quad (T_{NC} \le i < C_{TOTAL}) \tag{16}$$

$$q(i+1, i) = i+1 \quad (T_{NC} \le i < C_{TOTAL}) \tag{17}$$

Now, the steady state probabilities of P_{UTD} and P_{NTB} for the admission control scheme can be expressed as follows, (18)–(21), as shown at bottom of the page.

IV. INPUT TASK MODELING

In this section we present our smart home's input tasks modeling for simulated tasks dataset and the simulation setup.

The input tasks for simulated tasks dataset are generated based on sensor values inputs and system thresholds to generate control tasks set to be simulated. The tasks generated have initial parameters as tasks ID, arrival time, execution time, and deadline and period. Next the system computes the tasks parameters as start time and finish time based on initially generated parameters. Table 5 shows the list of tasks parameters generated for the tasks.

$$p_x = \frac{\frac{\left(CT_U^{x-T_{NC}}\right).(CT_U+CT_N+ST_N)^x}{x!}}{\sum_{n=0}^{T_{NC}} \left(\frac{(CT_U+CT_N+ST_N)^n}{n!}\right) + \sum_{n=T_{NC}+1}^{C_{TOTAL}} \left(\frac{\left(CT_U^{n-T_{NC}}\right).(CT_U+CT_N+ST_N)^{T_{NC}}}{n!}\right)} \quad (T_{NC} \le i < C_{TOTAL}) \tag{18}$$

$$p_x = \frac{\frac{(CT_U+CT_N+ST_N)^x}{x!}}{\sum_{n=0}^{T_{NC}} \left(\frac{(CT_U+CT_N+ST_N)^n}{n!}\right) + \sum_{n=T_{NC}+1}^{C_{TOTAL}} \left(\frac{\left(CT_U^{n-T_{NC}}\right).(CT_U+CT_N+ST_N)^{T_{NC}}}{n!}\right)} \quad (0 \le i < T_{NC}) \tag{19}$$

$$P_{UTD} = p^{C_{TOTAL}} \tag{20}$$

$$P_{NTD} = \sum_{x=T_{NC}}^{C_{TOTAL}} p^{C_{TOTAL}} \tag{21}$$

TABLE 5. Generated task parameters list.

Task	CPU Time
Task ID	Tasks' unique identifier
Arrival Time	Task arrival time at system
Execution Time	Time required to complete task
Deadline	Time before which task should complete
Start Times	Tasks' start time at machine
Finish Time	Tasks' finish time at machine

The execution time for sensing tasks is set to be 20 milliseconds (ms) and the priority is set to be normal periodic tasks. The execution time for all the control tasks is set to be 520 milliseconds (ms) and the priorities for environmental conditions control actuator (heater, chiller, humidifier, and dehumidifier) are set to be either urgent control task or normal control task (Table 6).

TABLE 6. List of tasks execution times and priority type for tasks.

Task	CPU Time	Task Priority	Task Type
Temperature Sensing Task	20ms	Normal Task	Periodic
Humidity Sensing Task	20ms	Normal Task	Periodic
Heater Control Task	520ms	Urgent/Normal Task	Event-Driven
Chiller Control Task	520ms	Urgent/Normal Task	Event-Driven
Humidifier Control Task	520ms	Urgent/Normal Task	Event-Driven
Dehumidifier Control Task	520ms	Urgent/Normal Task	Event-Driven

V. IMPLEMENTATION ENVIRONMENT

We have used python for implementing the core programming logic of the optimization, admission control and task scheduling algorithms. Python is a very popular general-purpose programming language; widely used for developing desktop based and web-based applications. The development environment for the system is shown in table 7.

TABLE 7. List development environment.

System Component	Value
Operating System	Windows
CPU	Intel ® Core ™ i5-4570 CPU at 3.20 GHz
Primary Memory	8 GB
Programming Language	Python 3

In order to simulate our testing scenarios, we have used online available data of energy consumption, temperature and humidity [28].

VI. PERFORMANCE ANALYSIS

In this section, we present the performance analysis for energy costs optimization process; fuzzy logic membership functions classification and control for temperature and humidity; tasks' admission control approach comparisons with no admission control approach; and tasks starvation rate and missing rates comparisons at the scheduler.

Fig. 6 shows the optimal energy cost for a scenario of tuning humidity and temperature values using optimization algorithm and objective function to strike a balance between user desired range and total energy cost. Two actuators are controlled to optimize the user desired settings; dehumidifier and chiller. Fig. 6 shows the clear difference between the energy costs for actuator control using fuzzy logic with no optimization scheme in comparison to proposed optimization scheme. We can observe 22.8371% energy savings in case of dehumidifier control and 36.06648% energy savings in case of chiller control.

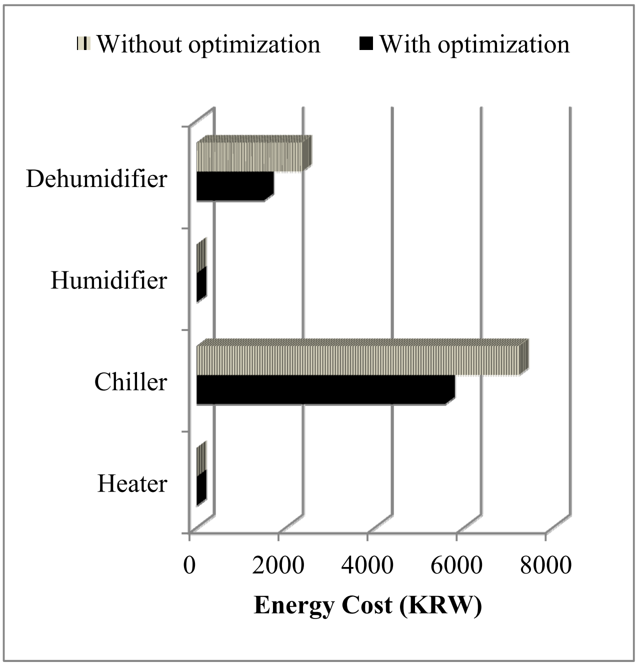


FIGURE 6. Energy cost comparisons for with and without optimization scheme.

Fig. 7 shows the energy optimization results for test run of 200 iterations. The test run iterations are performed based on varying sensing values of input parameters. The optimization with different sensing values' scenarios shows an average of 29.73% savings in energy costs.

We compare our proposed objective function's VB-PSO based implementation with GA (Genetic Algorithm) based implementation, for energy optimization. We have selected GA, as it is one of the most popular algorithms for energy

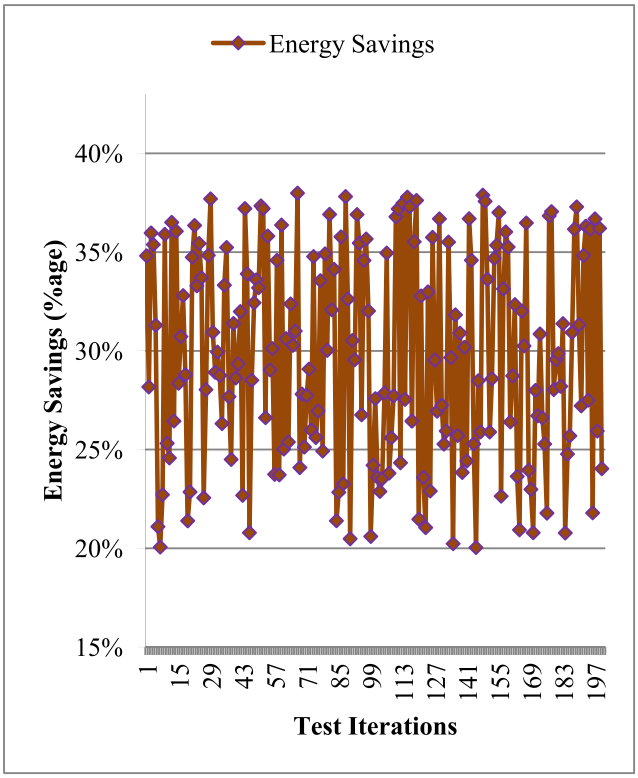


FIGURE 7. Percentage of savings in energy costs from optimization.

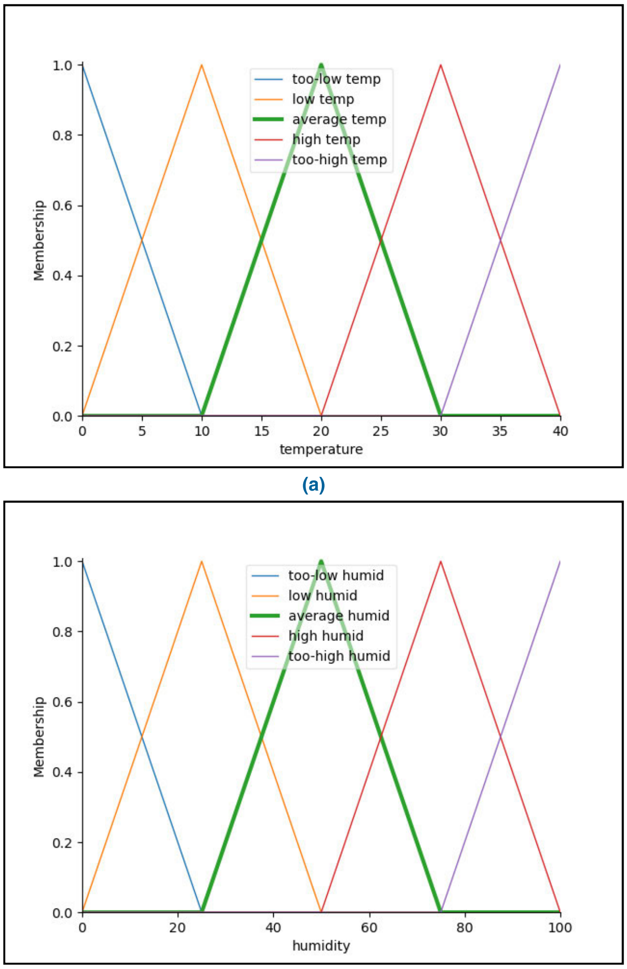
optimization problems. The number of epochs taken to reach optimal solution in PSO scheme are 94 whereas GA takes 131 epochs to reach optimal solution. The average energy savings using PSO is 29.73% while using GA is 28.59%. The average energy saving with PSO is 1.14% higher with respect to GA. We can witness a major difference in number of epochs required to get optimal solution, where PSO takes 37 less epochs in comparison to GA (Table 8).

TABLE 8. Optimization scheme comparisons.

System Component	Optimal Search Epochs	Avg. Energy Savings
VB-PSO Scheme	94	29.73%
GA Scheme	131	28.59%

Fig. 8 shows the custom membership function for temperature and humidity with fuzzy logic. Fig. 8 (a) shows the temperature membership function for the temperature classification using fuzzy logic. The temperature is divided into five classes as too-low, low, average, high and too-high. Fig. 8 (b) shows the humidity membership function for the humidity classification using fuzzy logic. The humidity is divided into five classes as too-low, low, average, high and too-high.

Fig. 9 shows the custom membership function for temperature control and humidity control with fuzzy logic. In fig. 9 (a), it can be observed that the temperature controls are divided as cool temperature, no-change in temperature and heat temperature. The black bar shows the final set temperature with fuzzy logic after getting optimal parameter values. In fig. 9 (b), it can be observed that the humidity controls are divided as decrease humidity, no-change in humidity and increase humidity. The black bar shows the final set humidity with fuzzy logic after getting optimal parameter values.



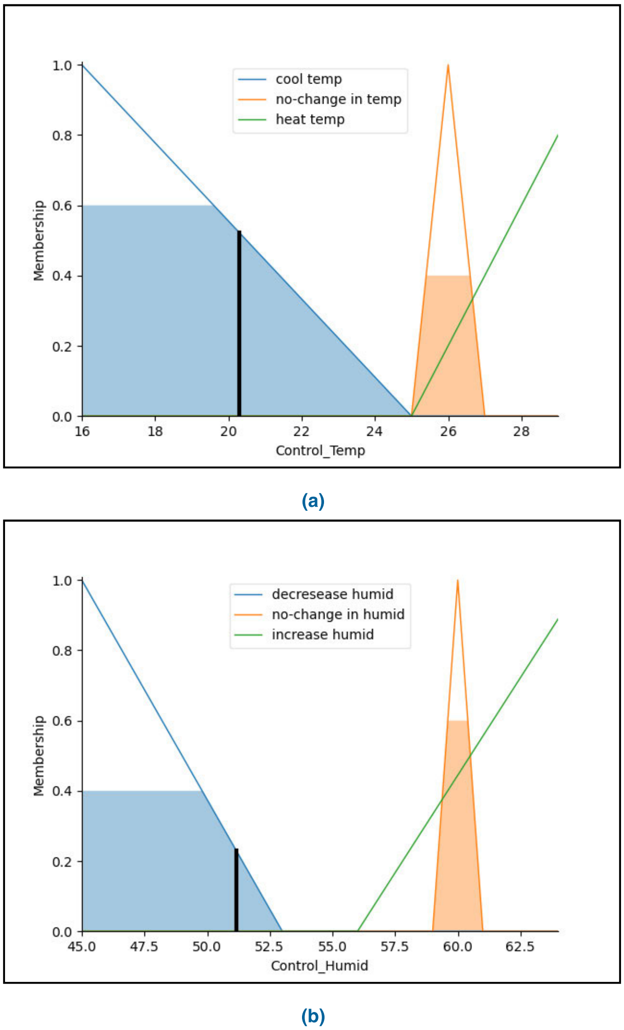
(a)

(b)

FIGURE 8. Degree of membership for parameter classification (a) for temperature; (b) for humidity.

Fig. 10 and fig. 11 show the comparison between scheduling approach with no admission control for tasks' dispatching to processor in comparison to scheduling scheme with admission control. The system states with no admission control are presented from equation (10) to (12) while the system states for proposed admission control are given from equation (13) to (21) in section 3. The threshold for acceptance of normal control tasks (T_{NC}) is set to be 40 with total capacity (C_{TOTAL}) of processor set as 50.

In fig. 10, the dropping probability of urgent tasks is given with respect to varying rate of urgent tasks load at the scheduler. We can observe that as the load of urgent tasks at the scheduler increases, the dropping probability of urgent tasks increases in both the approaches; as it is the worst case



(a)

(b)

FIGURE 9. Degree of membership for control (a) for temperature; (b) for humidity.

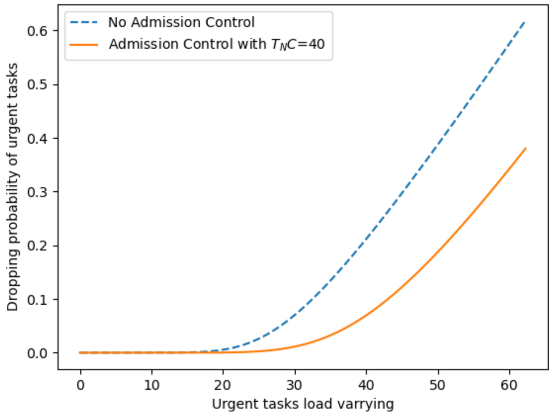


FIGURE 10. Dropping probability of urgent tasks w.r.t varying load of urgent tasks.

scenario for the scheduler to handle. Still, we can also clearly observe that the admission control approach gives almost 50% less dropping probability of the urgent tasks as compared to the no admission control scheduling scheme.

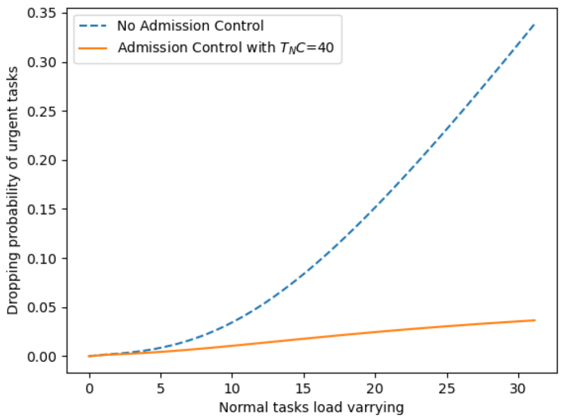


FIGURE 11. Dropping probability of urgent tasks w.r.t varying load of normal tasks.

In fig. 11, the dropping probability of urgent tasks is given with respect to varying rate of normal tasks load at the scheduler. In this scenario, as the load of normal tasks is varying and the scheduler's tasks dispatching unit has set a threshold to spare enough slots for urgent tasks to be processed for admission control, hence the dropping rate of urgent tasks is very low for admission control scheme in comparison to the no admission control.

Once the tasks are dispatched to the processors from scheduler's tasks dispatching unit, next is aiming to minimize any unnecessary starvation of a low priority task or missing of a high priority task. At each processor FEF scheduling algorithm calculates the urgency measure for the available tasks and sends the alert if a slot to accommodate a starving task or about to miss task can be managed; without missing the deadline for any currently queued tasks. Table 9 shows the average percentages for tasks starvation rates and the tasks average missing rate at the scheduler for admission control with UM (FEF) scheduling scheme in comparison to admission control with baseline priority scheduling scheme. In baseline priority scheduling scheme, the tasks are executed based on priority set from dispatching unit; and urgency measure from FEF scheduling algorithm is not calculated. Task starvation rate is the percentage of tasks that are deprived of processor time for long durations and tasks missing rate is the percentage of tasks missing their deadline.

TABLE 9. Average tasks starvation rate and tasks missing rate.

	Average Instances Missing Rate	Task Starvation Rate
Admission Control + UM (FEF Scheduling)	11%	6.51%
Admission Control + Bassline priority Scheduling	17.41%	10%

VII. CONCLUSION

In this work, we have proposed an integrated solution for smart home's efficient control and task management based on optimization and scheduling techniques. The main contributions of this work include the designed objective function for optimal energy consumption and development of efficient scheduling scheme for optimal control of smart home environments. The efficient scheduling scheme is based on admission control, load balancing and reducing tasks' starvation.

The proposed objective function aims for saving energy costs by setting optimal parameters of temperature and humidity. The optimal values for temperature and humidity are then sent to fuzzy logic module where the fuzzy control rules are generated based on the calculated optimal values and the control tasks commands are generated at fuzzy logic module. At scheduler, first the tasks are dispatched following the load balancing and admission control strategy and then at each processor UM based scheduling is performed for minimizing tasks' starvation.

It is evident from the results that the proposed optimization scheme saves energy costs drastically as compared to baseline scheme where no optimization is performed. We have run 200 iterations of optimization with different sensing values' scenarios resulting in various combinations of actuators' control and found an average of 29.73% savings in energy costs. The proposed tasks dispatching scheme of admission control using Markov model makes the task of load balancing among the processors efficient while also giving priority to the urgent tasks. It drastically reduces the dropping rate of urgent tasks with better management of task load distribution. The results for scheduler evidently show almost 50% less dropping probabilities for urgent tasks along with the 34.9% improvement in the tasks starvation rates and 36.82% in the tasks missing rates.

The limitations of proposed admission control scheme for scheduling include the probability of blocking more normal tasks when the arriving load of urgent tasks is high. As, system is set to prioritize the urgent tasks over normal tasks. Also, in case of multi-user scenario, in this work our scope is limited to consider the average ranges of user set requirements. In the future works, one important direction can be learning of multi-user activity patterns and optimize based on the user history.

REFERENCES

[1] *Energy Data Facts*. Accessed: Oct. 22, 2020. [Online]. Available: https://rpsc.energy.gov/energy-data-facts

[2] *Energy Consumption By Sector*. Accessed: Oct. 22, 2020. [Online]. Available: https://www.eia.gov/totalenergy/data/monthly/pdf/sec2.pdf

[3] A. Shah, H. Nasir, M. Fayaz, A. Lajis, and A. Shah, "A review on energy consumption optimization techniques in IoT based smart building environments," *Information*, vol. 10, no. 3, p. 108, Mar. 2019.

[4] J. Leitao, P. Gil, B. Ribeiro, and A. Cardoso, "A survey on home energy management," *IEEE Access*, vol. 8, pp. 5699–5722, 2020.

[5] L. Adua, "Reviewing the complexity of energy behavior: Technologies, analytical traditions, and household energy consumption data in the united states," *Energy Res. Social Sci.*, vol. 59, Jan. 2020, Art. no. 101289.

[6] R. Waheed, S. Sarwar, and C. Wei, "The survey of economic growth, energy consumption and carbon emission," *Energy Rep.*, vol. 5, pp. 1103–1115, Nov. 2019.

[7] J. Han, C.-S. Choi, W.-K. Park, I. Lee, and S.-H. Kim, "Smart home energy management system including renewable energy based on ZigBee and PLC," *IEEE Trans. Consum. Electron.*, vol. 60, no. 2, pp. 198–202, May 2014.

[8] A. Anvari-Moghaddam, H. Monsef, and A. Rahimi-Kian, "Optimal smart home energy management considering energy saving and a comfortable lifestyle," *IEEE Trans. Smart Grid*, vol. 6, no. 1, pp. 324–332, Jan. 2015.

[9] W. Li, T. Logenthiran, and W. L. Woo, "Intelligent multi-agent system for smart home energy management," in *Proc. IEEE Innov. Smart Grid Technol.-Asia (ISGT ASIA)*, Nov. 2015, pp. 1–6.

[10] B. Zhou, W. Li, K. W. Chan, Y. Cao, Y. Kuang, X. Liu, and X. Wang, "Smart home energy management systems: Concept, configurations, and scheduling strategies," *Renew. Sustain. Energy Rev.*, vol. 61, pp. 30–40, Aug. 2016.

[11] M. Khan, B. N. Silva, and K. Han, "Internet of Things based energy aware smart home control system," *IEEE Access*, vol. 4, pp. 7556–7566, 2016.

[12] C. Passenberg, D. Meyer, J. Feldmaier, and H. Shen, "Optimal water heater control in smart home environments," in *Proc. IEEE Int. Energy Conf. (ENERGYCON)*, Apr. 2016, pp. 1–6.

[13] H. Hussain, N. Javaid, S. Iqbal, Q. Hasan, K. Aurangzeb, and M. Alhussein, "An efficient demand side management system with a new optimized home energy management controller in smart grid," *Energies*, vol. 11, no. 1, p. 190, Jan. 2018.

[14] M. Fayaz and D. Kim, "Energy consumption optimization and user comfort management in residential buildings using a bat algorithm and fuzzy logic," *Energies*, vol. 11, no. 1, p. 161, Jan. 2018.

[15] H. Golmohamadi, R. Keypour, B. Bak-Jensen, and J. Radhakrishna Pillai, "Optimization of household energy consumption towards day-ahead retail electricity price in home energy management systems," *Sustain. Cities Soc.*, vol. 47, May 2019, Art. no. 101468.

[16] Z. A. Khan, A. Zafar, S. Javaid, S. Aslam, M. H. Rahim, and N. Javaid, "Hybrid meta-heuristic optimization based home energy management system in smart grid," *J. Ambient Intell. Humanized Comput.*, vol. 10, no. 12, pp. 4837–4853, Dec. 2019.

[17] A. Zielonka, A. Sikora, M. Wozniak, W. Wei, Q. Ke, and Z. Bai, "Intelligent Internet-of-Things system for smart home optimal convection," *IEEE Trans. Ind. Informat.*, early access, Jul. 14, 2020, doi: 10.1109/TII.2020.3009094.

[18] Y. Strengers and L. Nicholls, "Convenience and energy consumption in the smart home of the future: Industry visions from Australia and beyond," *Energy Res. Social Sci.*, vol. 32, pp. 86–93, Oct. 2017.

[19] T. Molla, B. Khan, and P. Singh, "A comprehensive analysis of smart home energy management system optimization techniques," *J. Auton. Intell.*, vol. 1, no. 1, pp. 15–21, 2018.

[20] T. Molla, B. Khan, B. Moges, H. H. Alhelou, R. Zamani, and P. Siano, "Integrated optimization of smart home appliances with cost-effective energy management system," *CSEE J. Power Energy Syst.*, vol. 5, no. 2, pp. 249–258, 2019.

[21] I. Gonçalves, Á. Gomes, and C. Henggeler Antunes, "Optimizing the management of smart home energy resources under different power cost scenarios," *Appl. Energy*, vol. 242, pp. 351–363, May 2019.

[22] I. O. Essiet, Y. Sun, and Z. Wang, "Optimized energy consumption model for smart home using improved differential evolution algorithm," *Energy*, vol. 172, pp. 354–365, Apr. 2019.

[23] B. Silva, M. Khan, and K. Han, "Load balancing integrated least slack time-based appliance scheduling for smart home energy management," *Sensors*, vol. 18, no. 3, p. 685, Feb. 2018.

[24] H. Swalehe, P. V. Chombo, and M. Boonruang, "Appliance scheduling for optimal load management in smart home integrated with renewable energy by using whale optimization algorithm," *GMSARN Int. J.*, vol. 12, no. 2, pp. 65–75, Jun. 2018.

[25] S. N. Makhadmeh, A. T. Khader, M. A. Al-Betar, S. Naim, A. K. Abasi, and Z. A. A. Alyasseri, "Optimization methods for power scheduling problems in smart home: Survey," *Renew. Sustain. Energy Rev.*, vol. 115, Nov. 2019, Art. no. 109362.

[26] Y. Fang and Y. Zhang, "Call admission control schemes and performance analysis in wireless mobile networks," *IEEE Trans. Veh. Technol.*, vol. 51, no. 2, pp. 371–382, Mar. 2002.

[27] S. Malik, S. Ahmad, I. Ullah, D. H. Park, and D. Kim, "An adaptive emergency first intelligent scheduling algorithm for efficient task management and scheduling in hybrid of hard real-time and soft real-time embedded IoT systems," *Sustainability*, vol. 11, no. 8, p. 2192, Apr. 2019.

[28] *Applainces Energy Prediction*. Accessed: Nov. 25, 2020. [Online]. Available: https://www.kaggle.com/loveall/appliances-energy-prediction#__sid=js0

SEHRISH MALIK received the B.S. degree in computer sciences from FAST-NUCES, Islamabad. She is currently pursuing the M.S. degree with Sangmyung University at Cheonan Campus. She is also as a member of the Protocol Engineering Lab (PEL) supervised by Prof. J.-H. Lee. After completing her M.S., she moved to Jeju National University for receiving Ph.D. degree under the supervision of Prof. D.-H. Kim with the Mobile Computing Laboratory (MCL). She successfully defended her Ph.D. thesis, in December 2019. She joined Kongju National University, where she is currently a Research Member of the DSP Laboratory. Her research interest includes machine learning applications in smart environments.

KYUTAE LEE received the B.S. and M.S. degrees in electronics engineering from Korea University, South Korea, in 1984 and 1986, respectively, and the Ph.D. degree in computer and electronics engineering from Korea University, in 1991. He joined the faculty of Kongju National University, South Korea, where he is currently a Professor with the Division of Information and Communication Engineering. His research interests include embedded DSP programming, reconfigurable open architecture, machine learning architecture, and circuit and systems. He is a member of Institute of Electronics and Information Engineers (IEIE).

DOHYEUN KIM received the B.S. degree in electronics engineering and the M.S. and Ph.D. degrees in information telecommunication from Kyungpook National University, South Korea, in 1988, 1990, and 2000, respectively. He was with the Agency of Defense Development (ADD), from March 1990 to April 1995. Since 2004, he has been with Jeju National University, South Korea, where he is currently a Professor with the Department of Computer Engineering. From 2008 to 2009, he was with the Queensland University of Technology, Brisbane, QLD, Australia, as a Visiting Researcher. His research interests include sensor networks, M2M/IOT, energy optimization and prediction, intelligent service, and mobile computing.