SINERGI Vol. 30, No. 3, October 2026: 863-878
http://publikasi.mercubuana.ac.id/index.php/sinergi
http://doi.org/10.22441/sinergi.2026.3.014

Real-time multiobjective optimization for home load scheduling on an embedded device using an integrated SCADA–AI system

Akbar Nursya'banni Ardiyanto, Aryuanto Soetedjo*, Irmalia Suryani Faradisa
Department of Electrical Engineering, National Institute of Technology (ITN) Malang, Indonesia

Abstract
Home energy management systems (HEMS) are designed to monitor and optimize residential energy consumption. This paper proposes an approach that integrates artificial intelligence (AI)-based optimization with a supervisory control and data acquisition (SCADA) system within an HEMS implemented on a Raspberry Pi embedded device using Node-RED and Python. The novelty of this work lies in enabling real-time SCADA monitoring and AI-based optimization on a single embedded platform. The theoretical implication of this study is the conceptualization of an integrated edge-based optimization design for HEMS, where multiobjective scheduling is embedded within the operational layer of the system architecture, contributing to the development of decentralized and edge-enabled energy management systems. The contributions include: (1) the development of an integrated SCADA–AI framework for real-time load scheduling on an embedded device; (2) the formulation of a multiobjective optimization model that minimizes grid-imported energy, maximizes photovoltaic (PV) utilization, and reduces the peak-to-average ratio (PAR); and (3) the comparative evaluation of three meta-heuristic algorithms—genetic algorithm (GA), particle swarm optimization (PSO), and ant colony optimization (ACO)—using real IoT-based SCADA monitoring data. The system schedules household loads, including the water heater, rice cooker, washing machine, and microwave oven. Experimental results show that GA achieved the best performance, reducing grid energy by 6.91%, increasing PV utilization by 14.91%, and lowering PAR by 24.44% compared to random scheduling. The execution times of GA, PSO, and ACO on Raspberry Pi 5 were 0.70 s, 0.38 s, and 1.42 s, respectively, demonstrating suitability for real-time embedded applications.

This is an open-access article under the CC BY-NC license.

Keywords:
AI;
Embedded device;
Load scheduling;
Multiobjective;
SCADA;

Article History:
Received: September 26, 2025
Revised: January 6, 2026
Accepted: March 3, 2026
Published: October 2, 2026

Corresponding Author:
Aryuanto Soetedjo
Department of Electrical Engineering, National Institute of Technology (ITN) Malang, Indonesia
Email:
aryuanto@lecturer.itn.ac.id

INTRODUCTION

Electricity consumption contributes approximately 40% of carbon emissions [1]. Intelligent energy management systems are required to control and monitor energy consumption effectively. Because of the increasing number of electrical appliances in residential homes, home energy consumption is expected to increase by 40% over the next 20 years [2]. This has created both opportunities and challenges in developing a home energy management system (HEMS) [3].

HEMS has attracted the attention of researchers in conducting and implementing related research. The existing related studies can be grouped into three main categories: AI-based optimization approaches, IoT-based HEMS implementations, and supervisory control and data acquisition (SCADA)-based energy management systems (EMS), as described below.

The application of artificial intelligence (AI) techniques to solve optimization problems in HEMS was studied in [4]. Energy management based on fuzzy logic has been proposed for the power allocation of household loads and photovoltaic (PV) [5]. The optimal scheduling of a home energy system using multiobjective integer nonlinear optimization was proposed in [6]. This method reduced the daily operational costs from 34.1% to 81.6%. In [7], a genetic algorithm (GA), binary particle swarm optimization (BPSO), and wind-driven optimization (WDO) were proposed to solve optimization problems for managing supply and demand in the integration of smart grids and home energy management. Fusion-based machine learning techniques have also been proposed to predict home energy consumption more accurately [8]. The algorithm fuses artificial neural network (ANN) and support vector machine (SVM) techniques using the fuzzy approach. A load scheduling method that utilizes mixed-integer linear programming (MILP) was proposed in [9]. A combination of integer programming and reinforcement learning (RL) techniques was proposed in [10] to achieve energy efficiency in HEMS. In this approach, an MILP was used for load scheduling, and an RL controller was used for battery charging and discharging. Although these approaches demonstrate strong optimization performance, most are formulated as offline optimization models and often neglect real-time implementation, including real-time monitoring features.

In [11], a predictive and feedback-based algorithm was proposed to reduce home energy consumption while maintaining standard user comfort. The system was implemented in real-time using a fog-based Internet of Things (IoT) architecture. An actual implementation of an HEMS utilizing data from IoT-based appliances was developed in [12] to minimize energy costs using mixed-integer programming (MIP). In [13], the long short-term memory (LSTM) algorithm was employed to classify home appliances using a dataset obtained from smart meters. The backtracking search optimization algorithm (BSA) and BPSO have also been used for optimal home load scheduling [14]. In addition, an IoT-based multiagent system has been proposed for real-time implementation. These approaches enabled real-time HEMS implementation for energy optimization; however, they are supported by standard or limited Graphical User Interfaces (GUIs), such as provided by the MATLAB GUI [12], a specific mobile application [13], and the ThingSpeak platform [14].

SCADA is an established system with a sophisticated GUI for monitoring, control, and supervision, which is a standard approach in critical infrastructures such as electricity generation and networks, water supply, oil pipelines, transportation systems [15], renewable resources [16], and building applications [17], [18]. In [19], a SCADA system was used to monitor daily power consumption and production from several energy resources (PV, battery system, and main grid), enabling real-time energy management projects. The SCADA system in [20] was used to improve the efficiency of the energy prediction algorithm in the open-pit mining industry. The security issues of SCADA systems in an energy management system (EMS) were studied in [21][22]. IoT-based SCADA was used for real-time monitoring, fault detection, and maintenance prediction of a large PV system [23]. An IoT-based supervisory control and data acquisition (SCADA) system was developed in [24] to monitor the power consumption of home appliances in real-time. In [25], an experimental testbed using an IoT-based SCADA was developed to monitor and control loads in a lecture room. SCADA applications were used to implement the human–machine interface (HMI) for energy management and to simulate the loads in the lecture rooms. In [26], a fuzzy-based embedded system and SCADA were integrated for load scheduling in an HEMS with a grid-connected PV. In these approaches, the adoption of SCADA in the EMS can be categorized into: performing rule-based energy management [19][25], monitoring and control with the AI-based EMS developed in a separate layer [20, 21, 22], or device [26], and monitoring and detection [23][24].

As described previously, although several studies have implemented SCADA and AI-based optimization techniques for the EMS, these components are typically developed and operated separately. In conventional or proprietary SCADA systems, the direct implementation of AI algorithms is often constrained by limited computational resources and vendor-specific software environments designed primarily for monitoring and control. However, recent IoT-based and open SCADA platforms increasingly support edge computing, enabling closer integration with AI-based optimization techniques, though such techniques are still rarely developed, especially for HEMS applications. Therefore, this study aims to fill this research gap by developing a unified SCADA–AI framework implemented on a single device, namely a Raspberry Pi, for home load scheduling. It leverages Node-RED programming [27] to develop the SCADA system, with a Python program to enable easy AI implementation on the embedded device.

To further position this study within recent developments in optimization techniques for HEMS, a comparative-analytical perspective is presented below. Recent studies have adopted advanced multiobjective optimization frameworks, such as Pareto front-based evolutionary algorithms, such as non-dominated sorting GA (NSGA-II [28] and NSGA-III [29]) and multiobjective evolutionary algorithm based on decomposition (MOEA/D) [30], which generate a set of non-dominated solutions that represent optimal trade-offs among multiple objectives. Additionally, Deep RL [31] was also adopted in the HEMS, enabling the system to learn optimal policies based on environmental conditions and user behavior.

Despite their strong theoretical capabilities, those approaches are often computationally expensive and are typically implemented in simulation environments or high-performance computing platforms. In practical HEMS deployments, especially on embedded or edge computing devices, computational efficiency and real-time operation are preferred considerations. Therefore, in this study, since the proposed approach is based on an edge-oriented optimization framework that balances multiobjective optimization with computational feasibility, meta-heuristic optimization techniques, such as GA, particle swarm optimization (PSO), and ant colony optimization (ACO), were adopted. These algorithms can solve nonlinear and multiobjective problems while remaining feasible for embedded implementations. Instead of explicitly constructing a full Pareto front, the multiobjective problem is formulated using a weighted-sum method and evaluated using GA, PSO, and ACO. This approach enables efficient real-time scheduling on an embedded device while still addressing multiple objectives in HEMS applications.

The main contributions of the proposed work are:

- Integration of SCADA and AI-based optimization techniques for real-time home load scheduling on a single embedded device.
- The optimal daily scheduling of household loads in a grid-connected PV system, which considers three main performance criteria: (i) minimizing grid-imported energy, (ii) maximizing the PV utilization, and (iii) minimizing the PAR of the total load profile.
- The evaluation of three meta-heuristic optimization techniques —GA, PSO, and ACO—for solving multiobjective optimization problems in home load scheduling.
- Open-source software and hardware platforms, namely Node-RED and Raspberry Pi, for low-cost HEMS implementation.
- An IoT-based SCADA system for real-time home appliance power monitoring and data collection.

METHOD

System architecture

The configuration of the integrated AI- and SCADA-based HEMS is shown in Figure 1. It consists of three main parts: sensors, a controller hardware platform, and a software platform. Sensors are used to measure PV, grid, and home appliance power. The power data are then sent to the SCADA for further processing. The controller hardware platform consists of the main controller of the HEMS, that is, the Raspberry Pi-embedded device. This device is used to implement the two main tasks: SCADA and AI-based load scheduling. The software platform consists of Node-RED and Python. Node-RED is primarily used to develop SCADA applications. Python is used to implement the optimization technique. The Python program can be embedded in Node-RED to integrate with the SCADA system and the AI-based load scheduling algorithm.

The primary key to the proposed system is the integration of SCADA and AI-based load scheduling, implemented on a Raspberry Pi device. SCADA provides a graphical user interface for monitoring and controlling the power of home appliances. The AI-based load-scheduling module then uses power data to optimize energy consumption by scheduling loads based on PV availability. The detailed algorithm for load scheduling is described in the next section.

Home appliance power monitoring and data acquisition system

The system configuration for home appliance power monitoring and data collection is shown in Figure 2. The system is an IoT-based SCADA monitoring system. Notably, this system has been installed in a home to monitor and collect power data for evaluating the proposed algorithm. The proposed system is an extension and improvement of the previous system; that is, it integrates SCADA and AI into an embedded device, rather than using separate HMI and embedded devices.

The sensors consisted of a solar irradiation sensor, a room temperature and humidity sensor, an outdoor temperature and humidity sensor, and a power-monitoring sensor. A power-monitoring sensor was used to monitor the power from the PV and grid, as well as the power consumption of home appliances, such as a refrigerator, microwave oven, water pump, freezer, water heater, washing machine, and rice cooker.

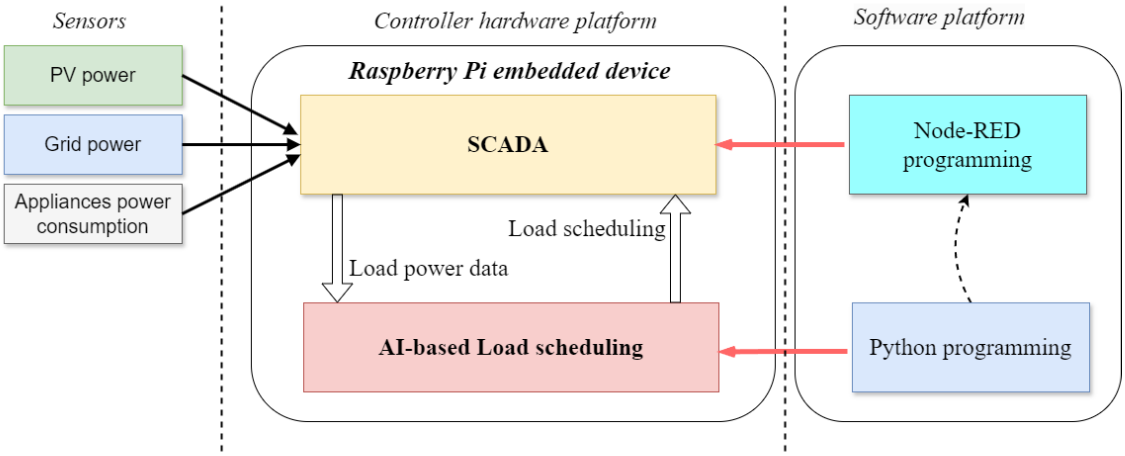

Figure 1. Configuration of the integrated AI and SCADA-based HEMS

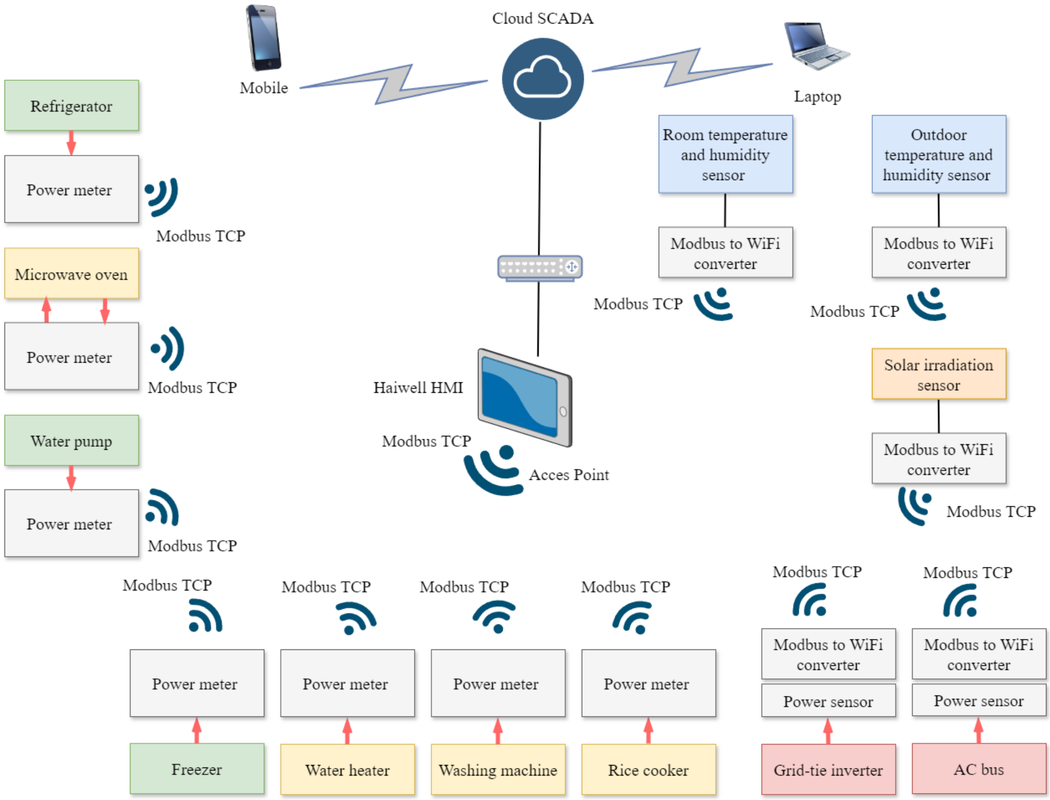

Figure 2. Configuration of the power monitoring and data collection system for home appliances

A Haiwell HMI was employed as the SCADA device. It communicates with the sensor units via Wi-Fi, with the HMI in access-point mode and the sensor units in station mode. The Modbus TCP protocol was adopted to facilitate interfacing with the sensor units. This configuration offers simple and low-cost installation at home. Because the Haiwell HMI is an IoT-based SCADA system, it is connected to the cloud server via the Internet. Thus, the device can be accessed remotely using a web browser on a personal computer or a mobile application on a smartphone.

Multiobjective optimization for home appliance scheduling

Home appliances can be divided into non-schedulable and schedulable loads. A non-schedulable load is one whose usage or operating time cannot be altered, such as lighting, refrigerators, water pumps, and freezers. Meanwhile, the usage time of the schedulable load can be altered by the user, such as in a microwave oven, water heater, washing machine, or rice cooker. This study focuses on the scheduling of schedulable loads for energy management in a home with a grid-connected PV system.

The proposed energy management algorithm aims to solve multiobjective optimization problems: minimizing grid-imported energy, maximizing PV utilization, and minimizing the PAR of power consumption. The first objective aims to reduce electricity costs and reliance on external grid supply, thereby increasing energy autonomy. The second objective aims to increase self-consumption, reduce reverse power flow, and improve the system's overall sustainability. The third objective aims to reduce stress on the grid infrastructure, avoid overloading, and improve voltage regulation.

Mathematically, multiobjective optimization problems can be expressed as follows. The grid-imported energy during a day is expressed as

$$E_{imp} = \sum_{t=1}^{144} \max\left(E_{tot}(t) - E_{PV}(t), 0\right) \quad (1)$$

where $E_{tot}(t)$ is the total load energy consumption at time slot t, and $E_{PV}(t)$ is the PV energy generation at time slot t. In this experiment, the time interval is 10 minutes; thus, the total time for a day is 144.

The PV utilization (PV_{util}) represents the portion of PV power that is directly consumed by the household loads, and is expressed as

$$PV_{util} = \frac{\sum_{t=1}^{144} \min\left(E_{PV}(t), E_{tot}(t)\right)}{\sum_{t=1}^{144} E_{PV}(t)}. \quad (2)$$

The PAR reflects the uniformity of the total load profile, and is expressed as

$$PAR = \frac{\max_t P_{tot}(t)}{\frac{1}{144}\sum_{t=1}^{144} P_{tot}(t)}, \quad (3)$$

where $P_{tot}(t)$ is the total load power at time slot t.

The three objectives are combined into a single scalar function using the weighted-sum method, which converts the multiobjective problem into a single cost that can be effectively solved by standard optimization algorithms such as GA, PSO, or ACO. The resulting optimization problem is expressed as

$$Min\, J = w_1 \frac{E_{imp}}{E_{base}} + w_2\left(1 - PV_{util}\right) + w_3 \frac{PAR}{PAR_{base}}, \quad (4)$$

where w_1, w_2, and w_3 are the weighting coefficients representing the energy cost reduction, PV utilization improvement, and load balancing performance, respectively; E_{base} and PAR_{base} are the base values for normalizing the imported energy and PAR. It is noted that since PV_{util} is to be maximized, the term $(1 – PV_{util})$ is introduced in (4). In the experiments, w_1=0.5, w_2=0.3, and w_3=0.2. The relative priority of each objective in the energy management context determines the weights. Energy cost reduction received the highest weight (w_1 = 0.5), reflecting its status as the primary objective. PV utilization was assigned a moderate weight (w_2 = 0.3) to promote effective use of renewable energy and reduce reliance on the grid. Load balancing was given a lower weight (w_3 = 0.2) to support operational stability and mitigate peak loads without excessively constraining the energy cost and PV utilization objectives.

The total load power is composed of non-schedulable loads and schedulable loads, which are expressed as

$$P_{tot}(t) = RF(t) + WP(t) + FZ(t) + LG(t) + \left(WH \times SWH(t)\right) + \left(WM \times SWM(t)\right) + \left(RC \times SRC(t)\right) + \left(MO \times SMO(t)\right), \quad (5)$$

where $RF(t)$, $WP(t)$, $FZ(t)$, and $LG(t)$ are the power consumptions of the refrigerator, water pump, freezer, and lighting at time slot t, respectively; WH, WM, RC, and MO are the power consumptions of the water heater, washing machine, rice cooker, and microwave oven, respectively.

The $SWH(t)$, $SWM(t)$, $SRC(t)$, and $SMO(t)$ represent the on/off status of the water heater, washing machine, rice cooker, and microwave oven, respectively, which are defined in the following equations:

$$SWH_t : \begin{cases} 1\, if\ t = x_1\ or\ t = x_2, x_1 \in [31,72], \\ \qquad\qquad\qquad\qquad x_2 \in [91,126] \\ 0\ otherwise \end{cases} \tag{6}$$

$$SWM_t : \begin{cases} 1\, if\ x_3 \le 7 \le x_3 + 3,\ \ x_3 \in [33,102], \\ 0\ otherwise \end{cases} \tag{7}$$

$$SRC_t : \begin{cases} 1\, if\ t \in [x_4, x_4 + 2] \cup [x_5, x_5 + 2] \\ \qquad\qquad\qquad \cup [x_6, x_6 + 2] \\ 0\ otherwise \end{cases} \tag{8}$$

$$x_4 \in [31,54], x_5 \in [67,84], x_6 \in [103,120]$$

$$SMO_t : \begin{cases} 1\, if\ t \in [x_7, x_7 + 2] \cup [x_8, x_8 + 2] \\ \qquad\qquad\qquad \cup [x_9, x_9 + 2] \\ 0\ otherwise \end{cases} \tag{9}$$

$$x_7 \in [31,54], x_8 \in [67,84], x_9 \in [103,120]$$

The allowable time slots for operating schedulable loads are expressed in (6)–(9). For instance, from (6), the water heater is operated twice a day with a duration of one time slot (10 min); the first usage can be from time slot 31 (05:00 h) to time slot 72 (11:50 h), and the second usage is from time slot 91 (15:00 h) to time slot 126 (20:50 h).

The three meta-heuristic optimization techniques, namely GA, PSO, and ACO, are evaluated for solving the above optimization problem. The flowcharts of the GA, PSO, and ACO algorithms are depicted in Figures 3, 4, and 5, respectively. The GA algorithm begins by initializing a population of randomly generated candidate solutions. Each individual represents a possible solution, in this case, the schedule of household appliances. After initialization, the fitness of each individual is evaluated using the multiobjective function defined in this study. Based on the evaluated fitness values, a selection process is conducted to identify individuals with better performance to serve as parents for the next generation.

Subsequently, crossover combines the two parent solutions, producing new offspring that explore different regions of the search space. The mutation is then applied to introduce small, random changes to the offspring solutions, helping maintain population diversity and preventing premature convergence.

The new population is formed and evaluated again iteratively. The algorithm terminates when the stopping criterion is satisfied, such as reaching the maximum number of generations or achieving convergence. Finally, the best individual in the population is selected as the optimal scheduling solution for the HEMS.

The PSO algorithm begins with particle initialization, in which the positions and velocities are randomly generated. Each particle represents a candidate scheduling solution. The fitness of each particle is then evaluated using the defined multiobjective function. After evaluation, each particle records its personal best solution (*pBest*), while the best solution among all particles is identified as the global best (*gBest*).

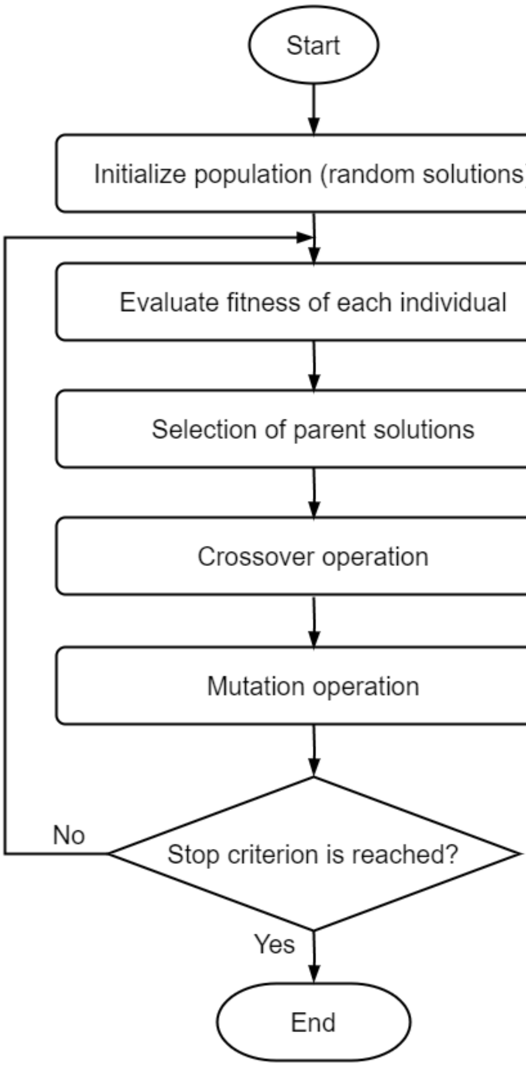

Figure 3. Flowchart of GA

During each iteration, the velocity of each particle is updated based on its current velocity, the distance from its personal best position, and the distance from the global best position, which is expressed as [32]

$$v_i^{t+1} = wv_i^t + c_1 r_1 \left(pBest_i - x_i^t\right) + c_2 r_2 \left(gBest - x_i^t\right) \tag{10}$$

where v_i^{t+1} is the velocity of particle i at iteration t; w is the inertia weight; c_1 is the cognitive coefficient; c_2 is the social coefficient; r_1 and r_2 are random numbers; $pBest_i$ is the best position previously visited by particle i; $gBest$ is the global best position; and x_i^t is the current position of particle i. The position of each particle is updated using the following formula [32]

$$x_i^{t+1} = x_i^t + v_i^{t+1} \tag{11}$$

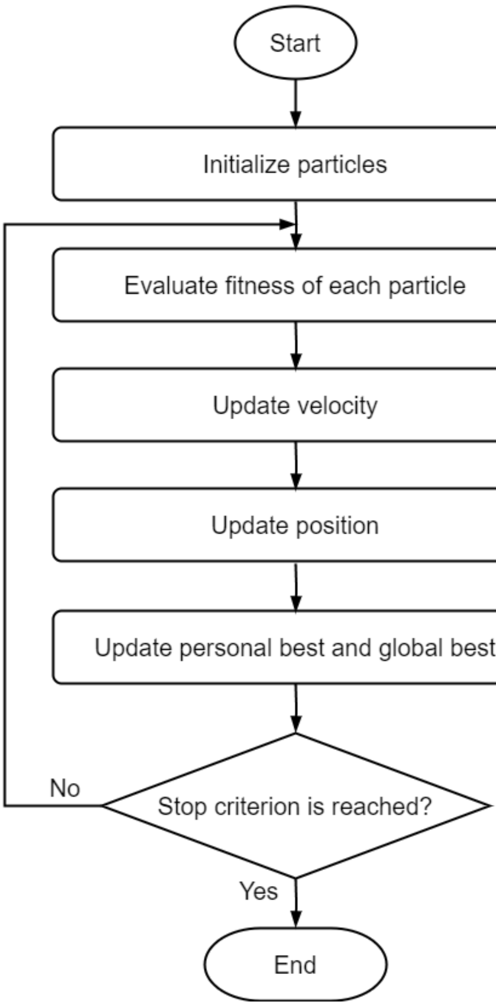

Figure 4. Flowchart of PSO

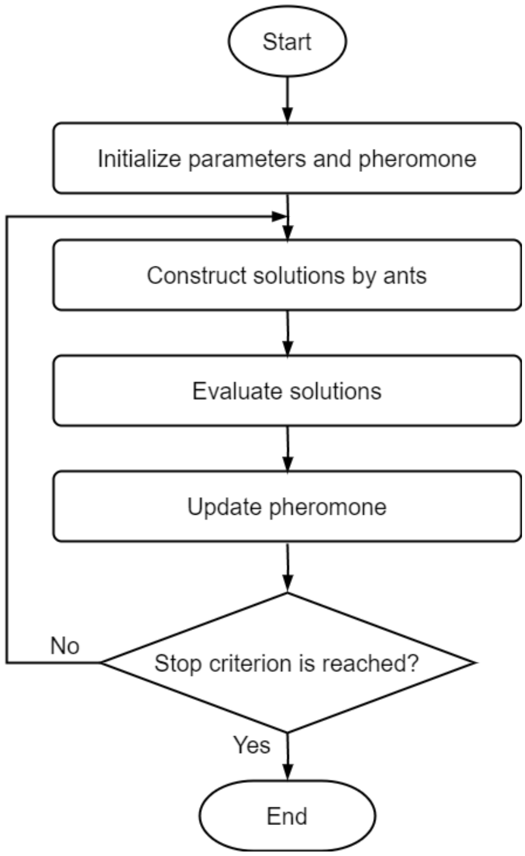

Figure 5. Flowchart of ACO

The updated solutions are evaluated again, and the *pBest* and *gBest* values are revised if better solutions are found. This iterative process continues until the stopping criterion is met. The final global best solution represents the optimal load scheduling configuration for the HEMS.

The ACO algorithm begins with the initialization of parameters and pheromone, such as the number of ants, pheromone trails, heuristic information, and evaporation rate. Next, a set of artificial ants is generated, where each ant k constructs a candidate solution based on a probability function expressed as [33]

$$P_{ij}^k = \frac{\left(\tau_{ij}\right)^\alpha \left(\eta_{ij}\right)^\beta}{\sum \left(\tau_{ij}\right)^\alpha \left(\eta_{ij}\right)^\beta} \tag{12}$$

where τ_{ij} represents the pheromone intensity from state i to j; η_{ij} represents the heuristic information from state i to j; α and β control the influence of pheromone and heuristic information, respectively.

After constructing the solutions, the fitness of each solution is evaluated using the multiobjective function. The pheromone trails are then updated by reinforcing paths associated with better solutions, which increases the likelihood that these paths will be selected in subsequent iterations. First, pheromone evaporation is applied to reduce the existing pheromone levels using the following formula [33]

$$\tau_{ij} \leftarrow (1-\rho)\tau_{ij} \tag{13}$$

where ρ is the evaporation rate. Next, pheromone reinforcement is performed using the following formula [33]

$$\tau_{ij} \leftarrow \tau_{ij} + \sum_{k=1}^{m} \frac{Q}{Fitness_k} \tag{14}$$

where Q is the pheromone deposition constant, and *fitness* is the objective value.

The process of solution construction and pheromone updating is repeated until the stopping criterion is reached. The best solution obtained during the search is selected as the optimal scheduling result.

The computational complexity of the algorithms can be analyzed using Big-O notation as follows. Let T be the number of iterations, N be the population size (individuals, particles, or ants), and D be the problem dimension representing the number of scheduling variables in the HEMS.

Each iteration of GA involves population evaluation, selection, crossover, and mutation. The dominant computational cost is the fitness evaluation of all individuals, leading to a time complexity of O(TND). In PSO, each iteration updates particle velocities and positions, followed by fitness evaluation. These operations scale linearly with both the number of particles and decision variables; thus, the time complexity is O(TND). It is worth noting that although the time complexity of GA is similar to that of PSO, the additional genetic operations introduce higher computational overhead for GA. For ACO, each ant incrementally constructs a solution and updates pheromone values. Since pheromone updates often involve interactions among decision variables, the time complexity becomes O(TND2).

The GA algorithm was implemented in Python using the PyGAD library [34], whereas the PSO and ACO algorithms were implemented in Python directly (without a specific library). The algorithms are implemented on a Raspberry Pi 5, a small-sized single-board computer powered by a 2.4 GHz quad-core 64-bit ARM Cortex-A76 CPU with 8 GB RAM. This is a low-cost and easy-to-install solution for homes, making it suitable for real-time implementation.

SCADA and AI-based optimization integration using Node-RED

Node-RED is an open-source, flow-based programming language in which a flow is used to connect nodes, which are functional units that perform specific tasks. It provides a browser-based visual interface with a drag-and-drop menu for easy programming. It supports various libraries for connecting hardware to other software programming languages, including Python. Figure 6 shows the Node-RED flow for the SCADA and AI integration. The nodes in the figures are divided into four groups defined by the bounding boxes. The main groups are located in the top left and bottom parts, as described below.

The nodes in the top-left group were used to configure and display the HMI. In this application, HTML-based SCADA was developed using a UI builder. The nodes in the bottom group were used to connect Node-RED with the Python program. One of the primary benefits of Node-RED is its intuitive interface for connecting to Python programs. Once the exec node is created, the Python program can be executed and interfaced with the Node-RED environment. Therefore, the AI-based optimization, written in Python, can be accessed via the Node-RED-developed SCADA. This feature offers two benefits: first, it facilitates the development of SCADA using Node-RED; second, it facilitates the development of AI-based algorithms using Python, a leading, popular, and open-source programming language for AI applications.

PV power and home appliance power consumption dataset

The proposed system was evaluated using the PV power and home appliance power consumption datasets collected using the system described in Figure 2. The recorded data comprised PV power and home appliance power consumption, which included schedulable and non-schedulable loads. Power data were recorded in real-time at 1-minute intervals. Notably, the monitoring system shown in Figure 2 does not monitor the lighting system because installing the lighting power sensor requires changes to the cable installation, rather than simply adding a plug-based power meter for other appliances. Therefore, the lighting data were added to the dataset based on the lamp power rating, and usage time was assumed to be the same every day (from evening to the next morning).

In the evaluation, the non-schedulable power profiles followed those in the dataset. However, because the proposed approach optimizes the operation time of schedulable loads, it only uses the average power usage from the dataset, and the operation time and duration follow the previously described load-scheduling scheme. Because the time interval used in load scheduling was 10 min, the dataset was resampled to 10-minute intervals. For simplicity, without loss of generality, only six days of power data, from 04/08/2025 (Monday) to

10/08/2025 (Sunday), were used in the evaluation. Table 1 lists the PV ratings and average schedulable load powers.

RESULTS AND DISCUSSION

HMI implementation

Figure 7 shows the main display of the developed HMI. In the central part, the status and power consumption of the appliances are displayed, as shown at the top center of the figure. The lower part shows the load-scheduling time of the appliances. An HMI provides a user-friendly interface for monitoring and managing home appliances. The main menu of the HMI operation is displayed in the top-left corner and includes the home menu, SCADA HEMS (SCADA monitoring), and the device management menu.

The management device menu for the HMI is shown in Figure 8. This menu is used to manage devices and home appliances for load scheduling. The user can add a new appliance or edit an existing appliance. The appliances were categorized into non-schedulable and schedulable loads, as previously described. With the proposed SCADA system, the user can easily configure home appliances to optimize energy use.

To evaluate the usability of the HMI, user experience, system responsiveness, refresh time, and user errors are compared with those of the existing Haiwell HMI [26]. To make a fair comparison, both HMIs are accessed from the same web browser via a Local Area Network (LAN) connection. The evaluation results are listed in Table 2. The table shows that the proposed HMI performs competitively compared to the established Haiwell HMI. The proposed HMI delivers a superior user experience, with fast, easy access to information via a scrollable menu. Furthermore, refinements are needed to enhance robustness during device configuration.

Evaluation of multiobjective optimization

The GA, PSO, and ACO parameters used for energy optimization are listed in Tables 3, 4, and 5, respectively. Notably, they are implemented in Python, enabling users to write and run the program independently of the Node-RED environment. It offers flexibility for debugging, modifying, and adopting new algorithms.

To evaluate the effectiveness of the proposed load scheduling, the results of the AI-based scheduling algorithm were compared with those of a random scheduling algorithm. In random scheduling, the operational time of a schedulable load is assigned randomly. Note that the range of operation time and the duration of random scheduling follow AI-based scheduling to ensure fair comparisons.

Tables 6, 7, and 8 present comparisons of GA, PSO, ACO, and random scheduling for energy import, PV utilization, and PAR, respectively. In each table, the random scenario is used as the baseline for comparing the effectiveness of the methods, i.e., reductions in energy imports and PAR, and increases in PV utilization.

Table 1. PV power rating and average power of schedulable loads

	PV	Water heater	Washing machine	Rice cooker	Microwave oven
Power rating	2000 Wp	568 W	90 W	330 W	1250 W
Usage time duration	12 h	10 min	40 min	30 min	10 min

Table 2. Evaluation of the HMI usability

Parameter	Proposed HMI	Haiwell HMI [26]
User experience	Displayed information is easily and quickly accessible due to the scrollable menu.	Users may lack a comprehensive overview of all available information at a glance.
System responsiveness	Very responsive. The transition between menus runs smoothly and quickly	Very responsive. The transition between menus runs smoothly and quickly
HMI Refresh time	Fast (less than 1 s)	Fast (less than 1 s)
User errors	Some errors occur when adding the new devices due to the unavailable icons	No errors occurred due to the limited and straightforward menus

Table 3. GA parameters

Parameter	Value
Number of generations	100
Number of parents	20
Number of solutions per population	40
Number of genes	9
Gen type	Integer
Crossover type	Single point
Mutation type	Random
Mutation percentage of genes	10%

Table 4. PSO parameters

Parameter	Value
Number of particles	30
Maximum number of iterations	80
Number of solutions per population	40
Inertia weight (w)	0.7
Cognitive coefficient (c_1)	1.5
Social coefficient (c_2)	1.5

Table 5. ACO parameters

Parameter	Value
Number of ants	40
Maximum number of iterations	80
Pheromone evaporation rate (ρ)	0.1
Pheromone deposition constant (Q)	1.0
Pheromone influence exponent (α)	1.0
Heuristic influence exponent (β)	0.0

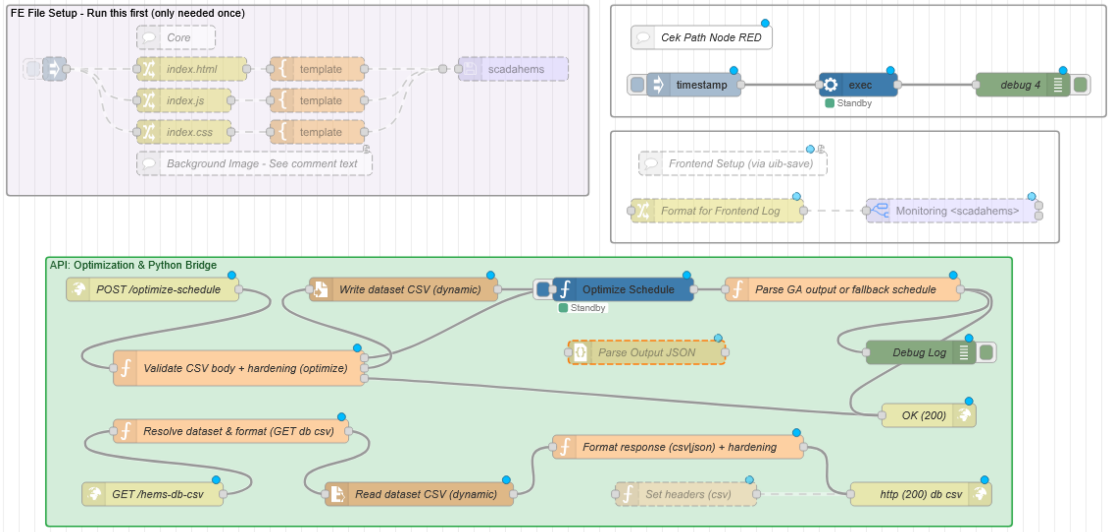

Figure 6. Node-RED flow for SCADA and GA integration

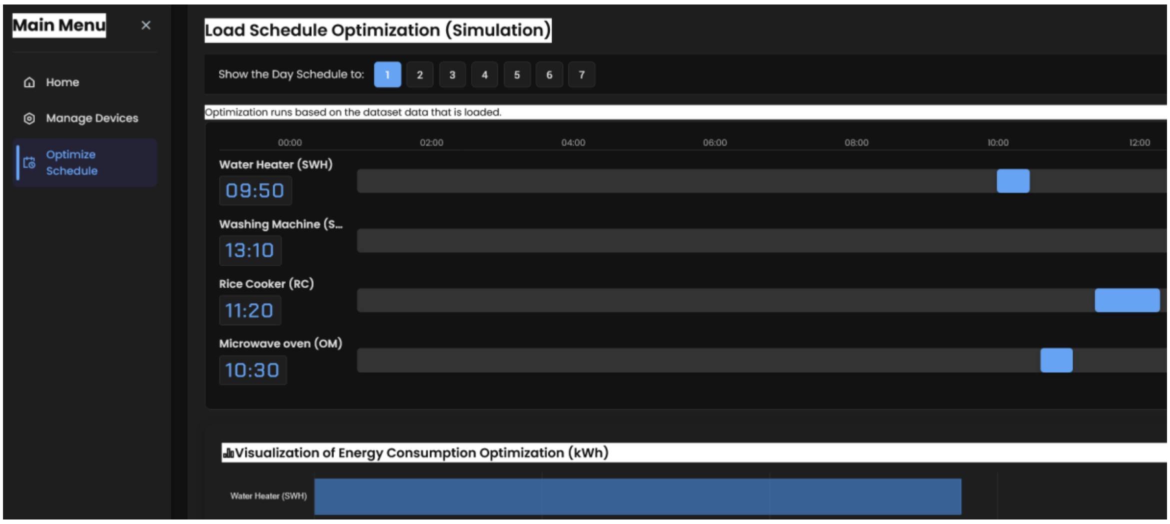

Figure 7. Main display of the HMI

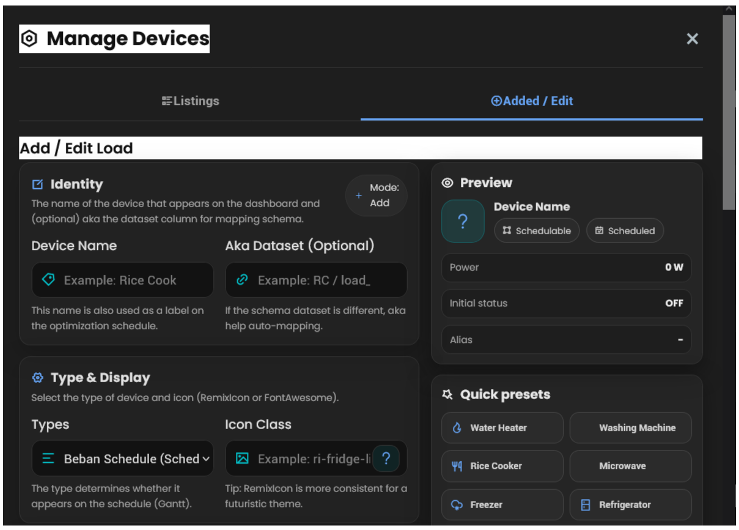

Figure 8. Managing-device menu in the HMI

Table 6 shows that all three optimization algorithms reduce the grid-imported energy compared to random scheduling. The lowest total energy import over 7 days is achieved by GA at 39.59 kWh, followed by PSO at 39.60 kWh and ACO at 40.38 kWh. The percentage reductions relative to random scheduling show that GA and PSO are nearly the same at 6.91% and 6.89%, respectively, while ACO is 5.07%. These results demonstrate that GA and PSO are more effective at minimizing grid-imported energy than ACO. These reductions exceed the 4.6% reported in [26].

Table 7 shows that the optimization algorithms increase the PV utilization, with GA, PSO, and ACO achieving average PV utilization of 34.24%, 34.23%, and 32.95%, respectively, which are higher than the random scheduling result of 29.14%. The improvement percentages are 14.91% (GA), 14.89% (PSO), and 11.57% (ACO) compared to the random scheduling. The results confirm that optimized scheduling successfully schedules appliances based on PV availability. Again, GA and PSO perform comparably, better than ACO. These improvements exceed 12.48% reported in [7].

The PAR comparisons in Table 8 demonstrate the ability of optimization algorithms to flatten the household load profile. Compared to random scheduling, GA, PSO, and ACO achieve PAR reductions of 24.44%, 23.89%, and 21.48%, respectively. The results show that the GA is superior in reducing the PAR.

Table 6. Comparison of energy import

Day	Energy import – GA (kWh)	Energy import – PSO (kWh)	Energy import – ACO (kWh)	Energy import – Random (kWh)
1	5.86	5.86	5.96	6.35
2	5.54	5.62	5.79	6.01
3	5.72	5.72	5.94	6.15
4	5.40	5.37	5.46	5.59
5	5.59	5.53	5.61	5.85
6	5.79	5.80	5.85	6.34
7	5.70	5.70	5.77	6.24
Total	39.59	39.60	40.38	42.53
Reduction (%)	6.91	6.89	5.07	-

Table 7. Comparison of PV utilization

Day	PV utilization – GA (%)	PV utilization – PSO (%)	PV utilization – ACO (%)	PV utilization – Random (%)
1	32.65	32.54	31.33	26.57
2	32.29	31.35	29.52	27.14
3	32.09	32.09	29.69	27.42
4	30.73	30.99	30.01	28.56
5	35.08	35.83	34.89	32.12
6	34.74	34.71	34.14	28.38
7	42.13	42.13	41.08	33.78
Average	34.24	34.23	32.95	29.14
Increment (%)	14.91	14.89	11.57	-

Table 8. Comparison of PAR

Day	PAR – GA	PAR – PSO	PAR – ACO	PAR – Random
1	3.72	3.91	3.91	5.40
2	3.56	3.56	3.82	4.85
3	3.85	3.85	3.85	4.95
4	3.87	3.87	4.07	5.23
5	3.68	3.68	3.91	5.09
6	4.12	4.12	4.12	4.81
7	3.58	3.58	3.74	4.58
Average	3.77	3.80	3.92	4.99
Reduction (%)	24.44	23.89	21.48	-

Figure 9 shows the average daily power load profile over 7 days, including PV power, baseline non-schedulable loads, and schedulable load power generated by four scheduling methods (GA, PSO, ACO, and random). The figure shows that the optimization algorithms shift the operation of the schedulable loads to the times when PV power is available. In contrast, the random method applies those loads independently of solar availability.

Figures 10 to 13 show the average daily schedulable load profiles for the random, GA, PSO, and ACO schedules, respectively. In the figure, the profiles of PV power and the load powers of the water heater (WH), washing machine (WM), rice cooker (RC), and microwave oven (MO) are shown. As shown in Figure 10, the RC and MO are operated in the early morning when there is no PV power by random scheduling, which does not occur in GA (Figure 11), PSO (Figure 12), or ACO (Figure 13). It clearly demonstrates the optimization algorithm's effectiveness in maximizing PV power use, thereby reducing grid-imported energy and increasing PV utilization.

From Figures 10 to 13, it is observed that load overlap occurs more frequently under random scheduling (Figure 10) than under optimization scheduling. The results provide evidence of PAR improvement achieved by the optimization algorithms.

Regarding execution time, the experimental results show that the execution times for GA, PSO, and ACO on a Raspberry Pi 5 are 0.70 s, 0.38 s, and 1.42 s, respectively, which are sufficiently fast for this application. The results comply with the time complexity discussed in the previous section.

Comparison to existing works

Table 9 presents a comparison with existing works, in which several parameters, such as the optimization objective and method, SCADA features, and implementation platform, are compared.

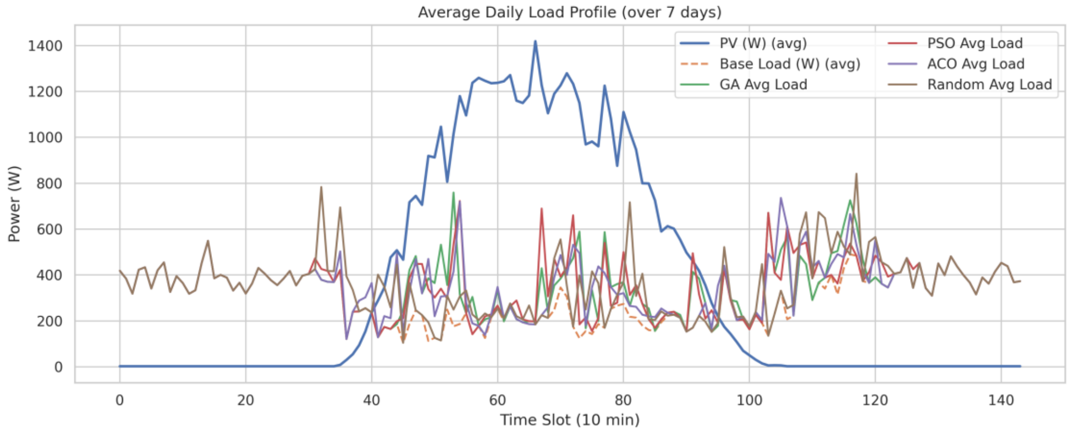

Figure 9. Average daily load power profile

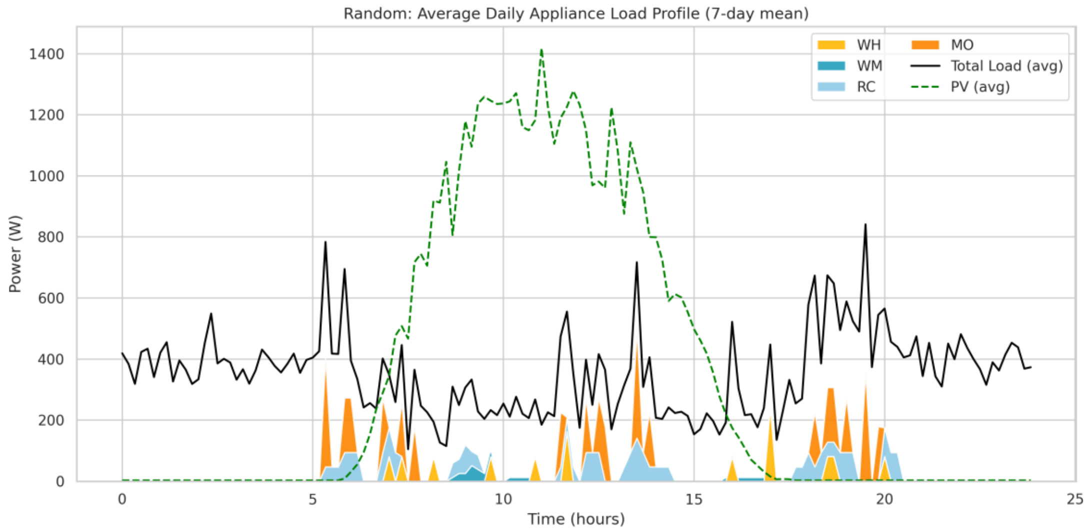

Figure 10. Average daily schedulable load power profile of random scheduling

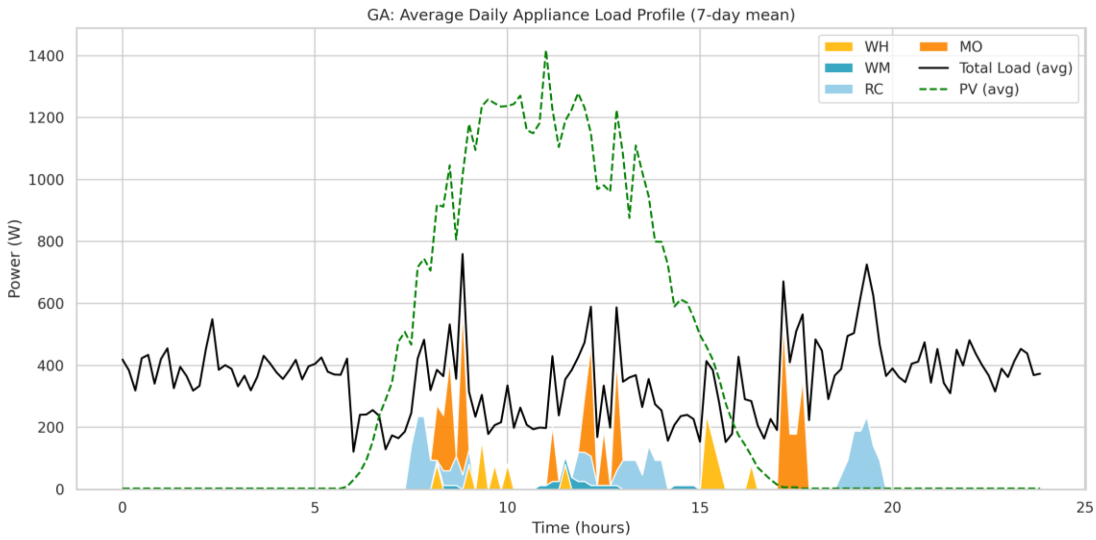

Figure 11. Average daily schedulable load power profile of GA scheduling

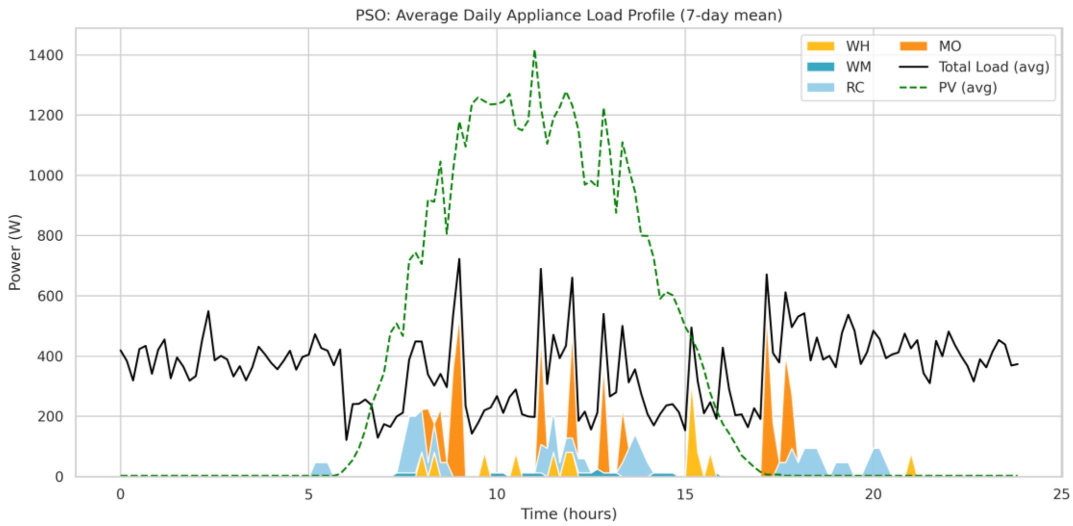

Figure 12. Average daily schedulable load power profile of PSO scheduling

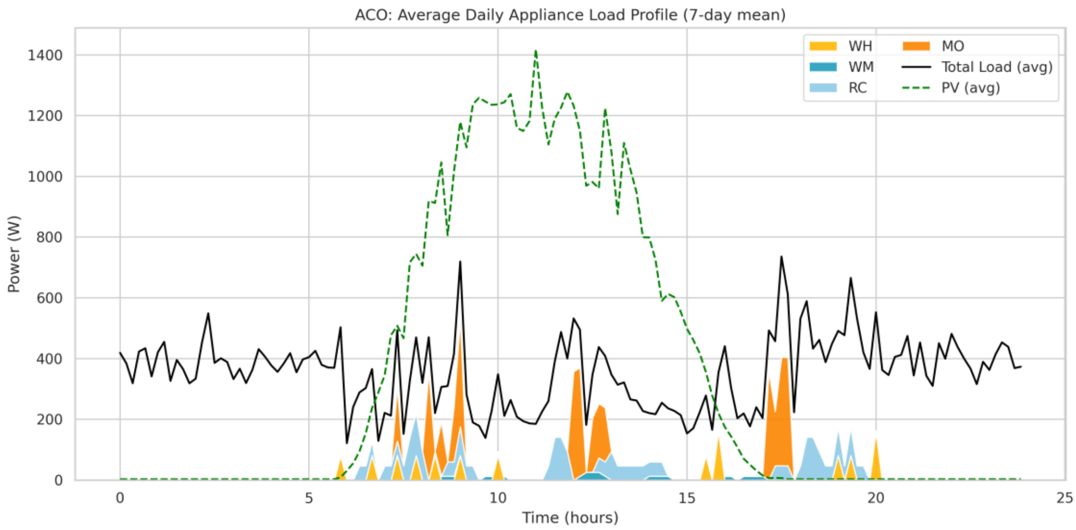

Figure 13. Average daily schedulable load power profile of ACO scheduling

In [7], multiobjective optimization is performed using metaheuristic algorithms. However, it is evaluated via simulation and lacks SCADA capability. In [12] and [14], the optimization algorithms are implemented on the hardware and IoT platforms. Since the ThingSpeak IoT platform has been adopted, the dashboard interface is more limited than that of a typical SCADA system.

The work in [26] proposed HEMS with SCADA functionality; however, it focuses on single-objective optimization and requires two devices: a Raspberry Pi for optimization and a Haiwell HMI for SCADA implementation. In contrast, the proposed approach offers all the features of multiobjective optimization using metaheuristic algorithms integrated with SCADA, implemented on a single embedded device.

Table 9. Comparison to existing works

Ref.	Optimization		SCADA feature	Implementation platform
	Objective	Method		
[7]	Energy cost, PAR, user comfort	GA, BPSO, WDO	No	Simulation
[12]	Energy cost	MIP	No	PC (C# and MATLAB)
[14]	Power cost, PAR	BSA, BPSO	No	Raspberry Pi, ThingSpeak IoT
[26]	Imported energy	Fuzzy logic	Yes (separate devices)	Raspberry Pi (Python), Haiwell HMI IoT-based SCADA
Proposed	Imported energy, PV utilization, PAR	GA, PSO, ACO	Yes (single device)	Raspberry Pi (Python, Node-RED)

CONCLUSION

In this study, multiobjective AI-based optimization was integrated with a SCADA system running on a single embedded device. The proposed system used Python and Node-RED on a Raspberry Pi platform to enable SCADA and AI integration. The experimental results show that load scheduling based on meta-heuristic algorithms, namely, GA, PSO, and ACO, effectively reduces grid-imported energy by 6.91%, 6.89%, and 5.07%; increases PV utilization by 14.91%, 14.89%, and 11.57%; and reduces PAR by 24.44%, 23.89%, and 21.48%, compared to random scheduling. These grid-imported and PAR-reduction percentages are better than those reported in previous works.

From a theoretical perspective, this work provides insight into integrating monitoring, control, and optimization within an edge-computing-based HEMS architecture. Unlike conventional approaches that separate SCADA systems from AI-based optimization modules, the proposed framework demonstrates that multiobjective optimization can be executed directly within a SCADA-enabled embedded platform while maintaining computational feasibility. This contributes to the development of practical multiobjective optimization frameworks for distributed energy management systems.

In terms of generalization capability, the proposed framework is adaptable to other residential energy systems that involve distributed energy resources, such as battery storage systems, smart appliances, and demand-response programs. Since the architecture is developed using open-source platforms and modularly integrates SCADA and AI algorithms, the system can be extended to support different optimization algorithms, larger datasets, and more complex residential or microgrid environments. This suggests that the approach can serve as a scalable solution for edge-based energy management applications.

However, several limitations should be acknowledged. First, the multiobjective optimization model adopts a weighted-sum method, which may not cover the entire Pareto-optimal solution space. Second, the system was evaluated using a limited dataset collected from a single house over a short period, which may not represent seasonal variations in PV generation and energy consumption. Third, the number of schedulable appliances is relatively limited compared to those in real smart home environments. Additionally, user behavior and comfort preferences were not explicitly modeled in the optimization process.

In the future, the proposed system will be extended to consider various home appliances, large homes, and residential areas, as well as advanced HEMS algorithms and evaluations using large datasets. Furthermore, the actual implementation at home will be investigated.

ACKNOWLEDGMENT

This work was supported by a Research Grant from the Ministry of Higher Education, Science, and Technology, Republic of Indonesia, 2025. (No.: SP DIPA- 139.04.1.693320/2025, 4th Revision, 30 April 2025)

REFERENCES

[1] S. Mischos, E. Dalagdi, and D. Vrakas, “Intelligent energy management systems: a review,” *Artificial Intelligence Review*, vol. 56, no. 10, pp. 11635–11674, 2023, doi: 10.1007/s10462-023-10441-3.

[2] A. Raza, L. Jingzhao, Y. Ghadi, M. Adnan, and M. Ali, “Smart home energy management systems: Research challenges and survey,” *Alexandria Engineering Journal*, vol. 92, pp. 117–170, 2024, doi: 10.1016/j.aej.2024.02.033.

[3] B. Han, Y. Zahraoui, M. Mubin, S. Mekhilef, M. Seyedmahmoudian, and A. Stojcevski, “Home Energy Management Systems: A Review of the Concept, Architecture, and Scheduling Strategies,” *IEEE Access*, vol. 11, pp. 19999–20025, 2023, doi: 10.1109/ACCESS.2023.3248502.

[4] M. Nutakki and S. Mandava, “Review on optimization techniques and role of Artificial Intelligence in home energy management systems,” *Engineering Applications of Artificial Intelligence*, vol. 119, pp. 1–16, 2023, doi: 10.1016/j.engappai.2022.105721.

[5] H. M. Usman, R. ElShatshat, and A. H. El-Hag, “Energy management for smart residential homes: A real-time fuzzy logic approach,” *Electric Power Systems Research*, vol. 238, pp. 1–11, 2025, doi: 10.1016/j.epsr.2024.111057.

[6] Z. Luo et al., “Optimal scheduling of smart home energy systems: A user-friendly and adaptive home intelligent agent with self-learning capability,” *Advances in Applied Energy*, vol. 15, pp. 1–20, 2024, doi: 10.1016/j.adapen.2024.100182.

[7] Z. Mahmood et al., “Efficient Scheduling of Home Energy Management Controller (HEMC) Using Heuristic Optimization Techniques,” *Sustainability*, vol. 15, no. 2, pp. 1–22, 2023, doi: 10.3390/su15021378.

[8] H. AlZaabi et al., “Intelligent Energy Consumption for Smart Homes Using Fused

Machine-Learning Technique," *Computers, Materials and Continua*, vol. 74, no. 1, pp. 2261–2278, 2023, doi: 10.32604/cmc.2023.031834.
[9] O. E. Dragomir and F. Dragomir, "Application of Scheduling Techniques for Load-Shifting in Smart Homes with Renewable-Energy-Sources Integration," *Buildings*, vol. 13, no. 1, pp. 1–14, 2023, doi: 10.3390/buildings13010134.
[10] R. Felicetti, F. Ferracuti, S. Iarlori, and A. Monteriù, "Peak shaving and self-consumption maximization in home energy management systems: A combined integer programming and reinforcement learning approach," *Computers and Electrical Engineering*, vol. 117, pp. 1–18, 2024, doi: 10.1016/j.compeleceng.2024.109283.
[11] M. Umair, M. A. Cheema, B. Afzal, and G. Shah, "Energy management of smart homes over fog-based IoT architecture," *Sustainable Computing: Informatics and Systems*, vol. 39, pp. 1–13, 2023, doi: 10.1016/j.suscom.2023.100898.
[12] H. Jo, H. Park, S. Kwon, and K. Cho, "Home Energy Management Systems (HEMSs) with Optimal Energy Management of Home Appliances Using IoT," *Energies*, vol. 17, no. 12, pp. 1–15, 2024, doi: 10.3390/en17123009.
[13] Z. Solatidehkordi, J. Ramesh, A. R. Al-Ali, A. Osman, and M. Shaaban, "An IoT deep learning-based home appliances management and classification system," *Energy Reports*, vol. 9, pp. 503–509, 2023, doi: 10.1016/j.egyr.2023.01.071.
[14] B. N. Alhasnawi et al., "A new Internet of Things based optimization scheme of residential demand side management system," *IET Renewable Power Generation*, vol. 16, no. 10, pp. 1992–2006, 2022, doi: 10.1049/rpg2.12466.
[15] M. Alanazi, A. Mahmood, and M. J. M. Chowdhury, "SCADA vulnerabilities and attacks: A review of the state of the art and open issues," *Computers and Security*, vol. 125, pp. 1–29, 2023, doi: 10.1016/j.cose.2022.103028.
[16] D. F. Larios, E. Personal, A. Parejo, S. García, A. García, and C. Leon, "Operational simulation environment for SCADA integration of renewable resources," *Energies*, vol. 16, no. 3, pp. 1–37, 2020, doi: 10.3390/en13061333.
[17] G. S. Gnana, A. Karthikeyan, K. Karthikeyan, P. Sanjeevikumar, S. Karapparambil Thomas, and A. Babu, "Critical review Of SCADA And PLC in smart buildings and energy sector," *Energy Reports*, vol. 12, pp. 1518–1530, 2024, doi: 10.1016/j.egyr.2024.07.041.
[18] J. Figueiredo and J. Sá Da Costa, "A SCADA system for energy management in intelligent buildings," *Energy and Buildings*, vol. 49, pp. 85–98, 2012, doi: 10.1016/j.enbuild.2012.01.041.
[19] M. Kermani, B. Adelmanesh, E. Shirdare, C. A. Sima, D. L. Carnì, and L. Martirano, "Intelligent energy management based on SCADA system in a real Microgrid for smart building applications," *Renewable Energy*, vol. 171, pp. 1115–1127, 2021, doi: 10.1016/j.renene.2021.03.008.
[20] O. Laayati, M. Bouzi, and A. Chebak, "Smart Energy Management System: Design of a Monitoring and Peak Load Forecasting System for an Experimental Open-Pit Mine," *Applied System Innovation*, vol. 5, no. 1, pp. 1–18, 2022, doi: 10.3390/asi5010018.
[21] Y. S. Yang, S. H. Lee, W. C. Chen, C. S. Yang, Y. M. Huang, and T. W. Hou, "Securing SCADA Energy Management System under DDos Attacks Using Token Verification Approach," *Applied Sciences (Switzerland)*, vol. 12, no. 1, pp. 1–15, 2022, doi: 10.3390/app12010530.
[22] Y. S. Yang, S. H. Lee, W. C. Chen, C. S. Yang, Y. M. Huang, and T. W. Hou, "TTAS: Trusted token authentication service of securing scada network in energy management system for industrial internet of things," *Sensors*, vol. 21, no. 8, pp. 1–22, 2021, doi: 10.3390/s21082685.
[23] S. Ferlito, S. Ippolito, C. Santagata, P. Schiattarella, and G. Di Francia, "A Study on an IoT-Based SCADA System for Photovoltaic Utility Plants," *Electronics (Switzerland)*, vol. 13, no. 11, pp. 1–18, 2024, doi: 10.3390/electronics13112065.
[24] S. Priyanto, A. Soetedjo, and I. B. Sulistiawati, "Monitoring the Power Consumption of Home Appliances Using an IoT-Based SCADA System," in *2024 8th International Conference on Information Technology, Information Systems and Electrical Engineering, ICITISEE 2024*, Yogyakarta, Indonesia: IEEE, 2024, pp. 441–446. doi: 10.1109/ICITISEE63424.2024.10730065.
[25] I. B. Sulistiawati, A. Soetedjo, D. B. Susanti, and G. Sukowiyono, "A Software Testbed for Developing the Energy Management System in Lecture Rooms Using IoT- based SCADA," in *2024 8th International Conference on Information Technology, Information Systems and Electrical*

Engineering (ICITISEE), Yogyakarta, Indonesia: IEEE, 2024, pp. 411–416. doi: 10.1109/ICITISEE63424.2024.10730127.

[26] S. Priyanto, A. Soetedjo, and I. B. Sulistiawati, “Integration of SCADA and embedded fuzzy-based load scheduling in Home Energy Management System with grid-connected PV,” *International Journal on Smart Sensing and Intelligent Systems*, vol. 18, no. 1, pp. 1–24, 2025, doi: 10.2478/ijssis-2025-0020.

[27] “Node-RED: Low-code programming for event-driven applications.” Accessed: Aug. 01, 2025. [Online]. Available: https://nodered.org/

[28] C. Chang and P.-F. Tsai, “Multiobjective Decision-Making Model for Power Scheduling Problem in Smart Homes,” *Sustainability*, vol. 14, no. 19, pp. 1–13, 2022, doi: 10.3390/su141911867.

[29] S. B. Utilization, “Many-Objective HEMS Based on Multi-Scale Occupant Satisfaction Modelling and,” *IEEE Transactions on Sustainable Energy*, vol. 13, no. 2, pp. 934–947, 2022, doi: 10.1109/TSTE.2022. 3140765.

[30] G.-T. Lin, W.-Y. Chiu, C.-F. Wu, A. Nazari, and D. Thiruvady, "Multiobjective model predictive control for residential demand response management under uncertainty," *Energy*, vol. 328, pp. 1–14, 2025, doi: 10.1016/j.energy.2025.136249.

[31] O. Al-Ani and S. Das, “Reinforcement Learning: Theory and Applications in HEMS,” *Energies*, vol. 15, no. 17, pp. 1–33, 2022, doi: 10.3390/en15176392.

[32] C. Menos-aikateriniadis, I. Lamprinos, and P. S. Georgilakis, “Particle Swarm Optimization in Residential Demand-Side Management: A Review on Scheduling and Control Algorithms,” *Energies*, vol. 15, no. 2, pp. 1–26, 2022, doi: https://doi.org/10.3390/en15062211.

[33] A. Métwalli, F. Fathy, E. Khatab, and O. Shalash, “ER-ACO: A Real-Time Ant Colony Optimization Framework for Emergency Medical Services Routing and Hospital Resource Scheduling,” *Algorithms*, vol. 19, no. 2, pp. 1–17, 2026, doi: 10.3390/a19020102.

[34] “PyGAD - Python Genetic Algorithm.” Accessed: Aug. 01, 2025. [Online]. Available: https://pygad.readthedocs.io/en/latest/