



Aalborg Universitet

AALBORG UNIVERSITY
DENMARK

Multiobjective Optimal Scheduling Framework for HVAC Devices in Energy-Efficient Buildings

Chakraborty, Nilotpal; Mondal, Arijit; Mondal, Samrat

Published in:
IEEE Systems Journal

DOI (link to publication from Publisher):
[10.1109/JSYST.2019.2933308](https://doi.org/10.1109/JSYST.2019.2933308)

Creative Commons License
CC BY 4.0

Publication date:
2019

Document Version
Accepted author manuscript, peer reviewed version

[Link to publication from Aalborg University](#)

Citation for published version (APA):
Chakraborty, N., Mondal, A., & Mondal, S. (2019). Multiobjective Optimal Scheduling Framework for HVAC Devices in Energy-Efficient Buildings. *IEEE Systems Journal*, 13(4), 4398-4409. Article 8809384. <https://doi.org/10.1109/JSYST.2019.2933308>

General rights


Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal -

Take down policy

If you believe that this document breaches copyright please contact us at vbn@aub.aau.dk providing details, and we will remove access to the work immediately and investigate your claim.

Multiobjective Optimal Scheduling Framework for HVAC Devices in Energy-Efficient Buildings

Nilotpal Chakraborty , *Student Member, IEEE*, Arijit Mondal, *Member, IEEE*,
and Samrat Mondal, *Senior Member, IEEE*

Abstract—The worldwide energy consumption has been growing in aggregate at a tremendous rate, and a majority of the same is due to heating ventilation air conditioning (HVAC) loads in urban buildings. With the help of the recent advances in energy management and optimization techniques, the operations and functioning of these devices can now be managed and controlled efficiently for an improved energy consumption scenario and thereby reducing cost. In this article, we propose a multiobjective optimal scheduling framework based on Johnson’s elementary circuit finding algorithm for controlling HVAC devices, specifically for buildings that require continuous thermal comfort maintenance. Two primary objectives addressed in this article are: minimizing power fluctuation and maximizing thermal comfortability of the users. We use standard comfortability indices to quantify thermal comfortability. To reduce the computation time, we also propose two intelligent improvement schemes that prune the exponential search space of Johnson’s algorithm. Furthermore, a new greedy scheduling algorithm has been proposed to obtain near-optimal solutions efficiently. All the proposed approaches have been studied in a simulated environment depicting a real-world scenario to evaluate their efficiency and effectiveness for practical implementations, including a comparative analysis with Karp’s minimum mean cycle algorithm in this problem setup.

Index Terms—Greedy algorithm, heating ventilation air conditioning (HVAC) scheduling, Pareto optimization, smart building (SB), thermal comfort.

NOMENCLATURE

AC	Air conditioners.
GNSA	Greedy nondominated scheduling algorithm.
HVAC	Heating ventilation air conditioning.
MOSA	Multiobjective optimal scheduling approach.
RES	Renewable energy sources.
RCMA	Remaining cumulative minimum average.
RMA	Remaining minimum average.
SB	Smart building.
SEC	Smart energy controller.

Manuscript received July 8, 2018; revised February 26, 2019, June 14, 2019, and July 28, 2019; accepted July 29, 2019. The work of A. Mondal and S. Mondal was supported in part by the IMPRINT-2 initiative undertaken by MHRD and DST through SERB, Govt. of India. (Corresponding author: Nilotpal Chakraborty.)

N. Chakraborty was with the Department of Computer Science and Engineering, Indian Institute of Technology Patna, Patna 801106 India. He is now with the Department of Computer Science, Aalborg University, Aalborg 9220 Aalborg, Denmark (e-mail: nilotpalc@cs.aau.dk).

A. Mondal and S. Mondal are with the Department of Computer Science and Engineering, Indian Institute of Technology Patna, Patna 801106, India (e-mail: arijit@iitp.ac.in; samrat@iitp.ac.in).

Digital Object Identifier 10.1109/JSYST.2019.2933308

TCD	Thermostatically controlled devices.
TCB	Thermal comfort band.
T_i^+	Upper band limit of i th AC.
T_i^-	Lower band limit of i th AC.
$T_i(t)$	Temperature maintained by the i th AC at time t .
P_i	Power consumption for i per unit time.
\mathcal{T}	Total duration of operation.
$T_a(t)$	Ambient temperature.
R_i	Thermal resistance of the zone maintained by i .
Q_i	Heat capacity of i .
H_i	Heat input rate of the zone maintained by i .
Δt	Time interval.
$s_i(t)$	State of i .
$P_{\text{tot}}(t)$	Total power consumption at time t .
P^*	Power consumption set-point.
T_u	Neutral temperature set-point.
$\mathcal{D}^{T_i(t)}$	Discomfort due to temperature $T_i(t)$.

I. INTRODUCTION

IN THE current worldwide scenario, a large portion of generated energy gets consumed by residential and commercial buildings. According to the U.S. Energy Information Administration, about 40% (or about 39 quadrillion British thermal units) of the total U.S. energy consumption was due to residential and commercial sectors in 2016 [1]. As per the International Energy Outlook, published in September 2017 [2], building energy consumption is expected to increase by 35% between 2015 to 2040. Moreover, as per the report published by the U.S. Department of Energy (Annual Energy Outlook, 2017), about 35% of energy consumption in residential, commercial buildings is due to heating ventilation air conditioning (HVAC) loads [3].

Most of the HVAC loads are predominantly thermostatically controlled devices (TCD), such as air conditioners (AC), room heaters, refrigerators, etc., whose sole objective is to maintain the temperature of its operating zone according to users’ desired thermal comfort level. To manage and maintain the desired thermal comfort, the thermostats residing in these TCDs follow a certain pattern of ON–OFF cycles, hovering around a certain temperature band, called thermal comfort band (TCB). For example, in case of an AC, the thermostat is switched OFF when the temperature hits the lower thermal limit (T_i^-), and it is switched ON immediately when the temperature reaches the upper thermal limit (T_i^+). Similar behavior is being depicted in Fig. 1 [4], [5].

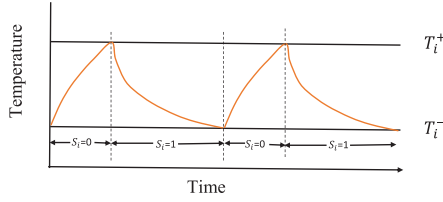


Fig. 1. Behavior of a thermostatic device (an AC).

At first glance, this process of switching may seem to be quite simple; however, this may possess a significant impact on the total energy consumption and peak load demand. For example, consider an office building having multiple HVAC units, each of which are required to maintain different TCBs based on the preferences of different users. Without a proper controlling or scheduling mechanism, it might well be possible to arrive at a situation when all the devices are switched ON to keep their respective zonal temperatures within the specified TCBs. This will not only increase energy consumption but will also increase peak load consumption significantly [6]. One naive way to solve this problem would be to limit the power consumption up to a specific threshold value. However, this may create a situation where the TCBs of all the ACs are not maintained, since now, only a few of them can be switched ON at a time. Thus, a smart and intelligent thermostatic device scheduling mechanism must be developed to mitigate these problems.

In practice, there are a number of challenging issues that the power grid system often encounters including system stability, load-generation management, network reliability, etc., and the increase in peak load demand, even for a smaller period of time, puts huge pressure on the overall performance of the system [7], [8]. Moreover, power stability may also give rise to privacy and security concerns for the users at multiple levels that must be considered for the successful grid operation [9], [10]. Although, all the issues and challenges must be addressed for the optimal functioning of the grid, in this article, we address the problem of managing and scheduling high power consuming appliances, such as ACs, in order to optimize the load profile. We also devise an intelligent scheduling mechanism for TCDs, specifically ACs, in buildings, such as large data centers, hospitals, or commercial spaces where these devices need to be operated round the clock. Given a set of ACs, along with their power consumption and the user desired temperatures for each of the devices, our objective is to obtain a repeatable schedule for the devices such that, first, the users comfort is maximized, and second, the obtained load profile is smoother for the total duration of operations. A smoother load profile not only helps in stabilizing the grid, but also in peak load reduction, and consequently reducing the total cost of operations and increasing users' financial gain [11], [12]. Systematically obtaining a schedule for the ACs that results in a smooth load profile, along with improving users' thermal comfortability, is the crux of this article. The novelty and highlights of this article are listed as follows.

- 1) Given the particular need to operate ACs in buildings for a longer time, we propose a graph-based multiobjective optimal scheduling approach (MOSA) to manage these devices, with the target to obtain smoother load profile and

maximizing users' thermal comfortability, where thermal comfort has been quantified according to the ASHRAE standard 55. We present a detailed discussion on the time complexity of the proposed MOSA along with its optimality.

- 2) Due to the exponential time complexity of MOSA, we propose two improvement schemes. The improvement schemes use a couple of intelligent branch and bound conditions that prunes a significant portion of the state space required to be explored by MOSA while extracting Pareto optimal solutions, thereby reducing the actual time to obtain the solution.
- 3) The proposed optimal approach and the improvement schemes have been experimented to schedule up to 50 devices, where we numerically evaluate the performance of the algorithms, both in terms of the obtained Pareto optimal solution set and execution time.
- 4) We further propose an efficient greedy search-based mechanism, named greedy nondominated scheduling algorithm (GNSA), to obtain near-optimal solutions quickly. We discuss its time complexity and verify its performance in comparison to the MOSA using well known performance evaluation metrics for multiobjective Pareto optimal solutions. During experimentation, GNSA is observed to produce results within the order of a few seconds in contrast to the results produced by MOSA, which took hours to execute.
- 5) The results obtained are highly encouraging, both in terms of efficiency and effectiveness. In particular, MOSA and the two improvement schemes were tested for up to 50 ACs to produce Pareto optimal solutions, and the computation time for MOSA was significantly reduced by the improvement schemes. We evaluated GNSA in comparison to the optimal solutions using performance metrics, where it is observed that about 70%–80% of the solutions resulted from the greedy algorithm where Pareto is the optimal set. We also evaluated GNSA using weighted sum method (WSM) where we compared it with Karp's minimum mean cycle algorithm (KMMC) and observed an average deviation of 7% only.

The rest of the article is organized as follows. In Section II, we provide an overview of the existing literature on thermostatic device scheduling. In Section III, we present the system model and formalize the problem mathematically. In Section IV, we present the proposed optimal scheduling approach for maximizing users' comfort and minimizing power fluctuation. In Section V, two improvement schemes are presented, and in Section VI, we discuss a greedy scheduling algorithm. Simulation results are presented and discussed in Section VII, and finally some concluding remarks have been provided in Section VIII.

II. RELATED WORK

Recently, quite a number of works have been reported on building energy management system in smart grid [7], [13]–[15] and a significant portion of the same is devoted on thermostatic device scheduling for minimizing total energy consumption and consequently peak load demand. Wu *et al.* [7] presented

a stochastic approach of energy management in residential buildings with renewable energy sources (RES), with the primary target to reduce cost due to the charging of electric vehicles. Wu *et al.* [8] targeted similar applications, however, with the additional objective to achieve optimal sizing of various components in a residential microgrid. Lu and Katipamula [4] presented a number of such controlling strategies for thermostatic devices, with water heaters as an example. Lu [5] presented an HVAC scheduling mechanism providing load balancing services efficiently. An efficient controlling mechanism for water heaters was also proposed by Shah *et al.* [16], where they controlled these devices by considering them as elastic loads, in a time-of-use pricing market. Martini *et al.* [17] proposed scheduling mechanisms of home appliances from the perspective of a cyberphysical energy system. Jindal *et al.* [18] proposed an HVAC scheduling approach to minimize total energy consumption targeting university buildings. Du *et al.* [19] proposed a distributed optimal approach for HVAC units based on statistics on weather forecasting. Similarly, distributed optimization was also targeted by Radhakrishnan *et al.* [20], where they proposed a learning-based HVAC scheduling mechanism for large commercial buildings.

Some controlling and scheduling mechanisms have also been proposed specifically for air-conditioning devices. Bashash and Fathy [21] presented a partial differential equation based model for efficient real-time scheduling and controlling of thermostatic cooling systems. Wai *et al.* [22] presented an efficient demand response mechanism for cooling devices such as ACs and freezers. Their performance evaluation on a practical dataset of Calgary, Canada show the effectiveness of their proposed mechanism. An optimal thermostatic programming solution has been suggested by Kamyar and Peet [23], having combined heating/cooling loads with a time-of-use pricing scheme. Karmakar *et al.* in [6] proposed a thermostatically controlled band management algorithm for ACs. Their primary objective is to maintain TCB for each of the ACs in the system under the constraint of peak load consumption. Chakraborty *et al.* [24] proposed a graph-theoretic-based scheduling approach that utilizes KMMC [25] for thermostatic devices that are operated for longer hours.

There are a number of multiobjective optimization problems (MOPs) addressed in the recent past on smart grid, where more than one objectives are addressed that are generally conflicting in nature. However, very limited works exist that address the concerns of thermostatic devices, such as maximizing thermal comfort, minimizing peak load consumption, improving load profile, etc. Li *et al.* [26] presented the multiobjective optimization problem (MOP) of sizing RES in order to support an island microgrid. They targeted optimizing costs of grid components including the installation costs for the RESs and improving the reliability of the grid. Shakouri [27] presented a MOP of reducing energy consumption and peak load demand. However, users preferences and comfortability are not a part of their optimization function. Similarly, Li *et al.* [28] discussed a MOP of minimizing the operational costs for the utility along with maximizing the aggregators' benefits, whereas consumers' operational requirements are not prioritized. However, selection of the final solution from the Pareto optimal front suffers from various drawbacks [29], [30].

Despite the amount of work done on various aspects of scheduling and controlling in the smart grid, only a limited amount of work has been pursued to address users' thermal comfortability improvement, that too for thermostatic devices. Providing a healthy and comfortable environment is one of the fundamental aspects of building energy systems, specifically HVAC units. On the other hand, thermal comfort varies from person to person and depends on quite a lot of factors such as metabolic rate, air temperature, clothing insulation, humidity, etc., as per the ASHRAE standard 55 [31]. Thus, the MOP of maximizing users' thermal comfort, while minimizing power consumption, is a highly challenging problem.

Compared to the existing literature, the work proposed here is one of its kind since we address two conflicting objectives of maximizing users' thermal comfort and minimizing load fluctuation by smoothening load profile due to thermostatic devices, simultaneously. We formulate the above-mentioned multiobjective conflicting optimization problem as a directed weighted graph and propose intelligent scheduling techniques for TCDs to obtain Pareto optimal solutions.

III. SYSTEM MODELING AND PROBLEM DESCRIPTION

Let us consider a smart building (SB) that has a smart energy controller (SEC) for managing and controlling all the appliances in the building. Let there be a total of n thermostatic devices (specifically ACs) in the SB that operate simultaneously to maintain their respective zonal temperatures within the desired limits. Each of the thermal zones maintained by these devices has specific values of heat input rates, thermal resistance, and heat capacity. Let the TCB maintained by the i th device be $[T_i^-, T_i^+]$, where T_i^+ denotes the upper band limit and T_i^- denotes the lower band limit. The TCB may be provided by the user or may be computed by the SEC according to the needs. A device i , while switched ON, consumes a P_i amount of power per unit time, whereas power consumption is 0 if switched OFF. Let $T_i(t)$ denotes the temperature maintained by the i th AC at time t . Then, throughout the duration of operations (\mathcal{T}) of the devices, we must have

$$T_i^- \leq T_i(t) \leq T_i^+ \quad \forall t \in \mathcal{T}. \quad (1)$$

As depicted in Fig. 1, change in temperature inside a room typically follows an exponential curve. This change depends on many factors including outdoor temperature and heat transfer coefficients associated with different heat transfer modes viz., conduction, convection, and radiation. However, for simplicity, we consider a unified deterministic offline model combining all the heat transfer modes as is done in practice [6], [24]. When switched ON, the temperature maintained by i th device at time t is described by

$$T_i(t) = T_a(t) - (T_a(t) - T_i(t-1)) e^{-\frac{\Delta t}{R_i Q_i}} \quad (2)$$

where $T_a(t)$ is the ambient temperature, measured in °F or °C, R_i represents thermal resistance of the zone maintained by the i th AC, Q_i is the heat capacity of i th AC, and $\Delta t (= t - (t-1))$ is the time interval. When a device is switched OFF, the change

in the temperature can be described by

$$T_i(t) = T_a(t) - (T_a(t) - T_i(t-1)) e^{-\frac{\Delta t}{R_i Q_i}} + H_i R_i \left(1 - e^{-\frac{\Delta t}{R_i Q_i}}\right) \quad (3)$$

where H_i is the heat input rate for the thermal zone maintained by the i th AC. Once the thermal models are formulated for the devices, we can move toward formalizing the scheduling problem we want to solve.

In this article, we address the challenging issue of scheduling ACs to optimize power consumption for a smooth load profile, as well as thermal comfortability of the users simultaneously. In the following sections, we present the mathematical formulations for each of these objectives and, subsequently, present the overall MOP.

A. Load Profile Optimization

As discussed earlier, there are a number of parameters needed to be optimized for the smooth functioning of the grid. In this article, we consider a relatively simpler model with the primary focus on smoothening of the load profile as one of the objectives that influence the grid operations. Let P_i be the instantaneous power consumption by the i th device whenever it is switched ON, and let $s_i(t)$ be a binary variable representing the state of i th device (0 when switched OFF and 1 when switched ON). Then, we must have

$$P_{\text{tot}}(t) = \sum_{i=1}^n s_i(t) \times P_i \quad \forall t \in \mathcal{T} \quad (4)$$

where $P_{\text{tot}}(t)$ is the total power consumption at time t . With \mathcal{T} being the total time of operation for the devices, the problem of scheduling for a smooth load profile can be formulated as

$$\text{Minimize : } \sum_{t=1}^{\mathcal{T}} |P_{\text{tot}}(t) - P^*| \quad (5)$$

where P^* is the desired total load consumption set-point for all $t \in \mathcal{T}$, the value of which can be considered to be supplied by the utility or may be locally computed by the SEC. The value of P^* is the ideal power consumption to make the load profile flat and is generally chosen depending on the grid conditions and local power consumption scenario. Here, we consider it as an input to the system.

Equation (5) represents our objective of scheduling for a smooth load profile by minimizing the absolute power consumption deviation from a given power consumption set-point (P^*) across the total duration of the operation. Therefore, the more is the absolute deviation of power consumption from the set-point, the higher is the fluctuation in load profile and thereby increasing the absolute deviation cost [12], which we seek to minimize. Apart from improving the operations of the grid, minimizing load variation also improves the privacy aspects of the consumers, thereby gaining their trust into the system [9], [10]. However, since (5) is monotonically increasing with respect to \mathcal{T} , solving it in its current form is difficult. Therefore, similar to [24], we try optimize the average absolute power consumption deviation (ρ) from the given set-point for every time slot $t \in \mathcal{T}$. Incorporating this, the objective function can

now be restated as

$$\text{Minimize : } \rho = \frac{1}{\mathcal{T}} \sum_{t=1}^{\mathcal{T}} |P_{\text{tot}}(t) - P^*|. \quad (6)$$

B. Comfort Optimization

Apart from power optimization, we also propose to maximize users' thermal comfort in this article. Comfort is a state of satisfaction that varies from person to person depending on various factors including metabolic rate, mean air temperature, humidity, clothing insulation, etc., and is, therefore, difficult to measure. ASHRAE Standard 55 [31] suggests the use of predictive mean vote, as a quantifiable thermal comfort index, where human sensation has been scaled from -3 to $+3$ corresponding to the categories: "cold," "cool," "slight cool," "neutral," "slight warm," "warm," and "hot."

Thus, maximizing users' comfort essentially means maximizing the time quantum for which the user senses "neutral." Given the TCB $[T_i^-, T_i^+]$ of an AC i around the user desired "neutral" temperature set-point T_u , we classify the comfort band into multiple thermal sensation zones as follows:

- 1) "cold" ($T_{\text{cold}} = [T_i^-, T_u - 2.5]$);
- 2) "cool" ($T_C = [T_u - 2.5, T_u - 1.5]$);
- 3) "slight cool" ($T_{SC} = [T_u - 1.5, T_u - 0.5]$);
- 4) "neutral" ($T_N = [T_u - 0.5, T_u + 0.5]$);
- 5) "slight warm" ($T_{SW} = [T_u + 0.5, T_u + 1.5]$);
- 6) "warm" ($T_W = [T_u + 1.5, T_u + 2.5]$);
- 7) "hot" ($T_{\text{hot}} = [T_u + 2.5, T_i^+]$),

where $[\bullet, \bullet]$, (\bullet, \bullet) , $[\bullet, \bullet)$, $(\bullet, \bullet]$ have their usual mathematical meanings.

Since we are looking to maximize users' thermal comfort, we ensure that a user senses "neutral" as long as possible by keeping the temperature in the predefined "neutral zone" (T_N) for maximum duration. In other words, we seek to minimize users' discomfort by minimizing the temperature deviation from the T_N zone. Below, we introduce a cost metric that defines the discomfort cost for the appliance i due to its temperature ($T_i(t)$) at time t as

$$\mathcal{D}^{T_i(t)} = \begin{cases} 0, & \text{if } T_i(t) \in T_N \\ 1, & \text{if } T_i(t) \in T_{SC} \vee T_i(t) \in T_{SW} \\ 2, & \text{if } T_i(t) \in T_C \vee T_i(t) \in T_W \\ 3, & \text{if } T_i(t) \in T_{\text{cold}} \vee T_i(t) \in T_{\text{hot}}. \end{cases} \quad (7)$$

Thus, the "neutral" zone ($T_u + 0.5 \leq T_i(t) \leq T_u - 0.5$) has a discomfort cost of 0, and as we deviate from that zone, cost starts increasing. The ± 0.5 difference between any two sensation levels is obtained from [31]. Note that discomfort cost cannot exceed a value of 3. This is to keep parity with the real-world scenario [31], [32]. Our objective is to minimize the discomfort for the total duration of operation, which we formulate here as a mean squared error function, as follows:

$$\text{Minimize : } \delta = \frac{1}{\mathcal{T}} \sum_{t=1}^{\mathcal{T}} \sum_{i=1}^n \left(\mathcal{D}^{T_i(t)}\right)^2. \quad (8)$$

C. Multiobjective Optimization Problem

Inherently, (6) and (8) are conflicting in nature, and we are looking for a tradeoff solution that optimizes both power consumption deviation and discomfort simultaneously. Therefore, the overall objective, combining (6) and (8), can be stated as

$$\text{Minimize : } (\rho, \delta). \quad (9)$$

Equation (9) represents the objective of the MOP that is solved, subjected to (1), (2), (3), (4), and (7). The results obtained from solving (9) will be a set of two-dimensional (2-D) points, where each point can be considered as a vector of two elements. To measure the quality of the obtained solutions, we utilize the concept of dominance relationship [33], which is defined as follows.

Definition 1: Let $S^a = [S_1^a, S_2^a]$ and $S^b = [S_1^b, S_2^b]$ represent two $2 - D$ vectors. Then for a minimization problem, S^a is said to dominate S^b if both the following conditions hold true.

- 1) $S_i^a \leq S_i^b \quad \forall i \in \{1, 2\}$.
- 2) $S_j^a < S_j^b \quad \exists j \in \{1, 2\}$.

A solution is said to be Pareto optimal, if there exists no other solution that dominates it. In the following section, we propose an intelligent scheduling approach that yields the complete Pareto optimal set of solutions for a given set of thermostatic devices. It is noteworthy to be mentioned here that all the modeling parameters discussed above are assumed to be known to the system, and therefore, the proposed scheduling approach in this article is essentially an offline mechanism.

IV. PROPOSED MULTIOBJECTIVE OPTIMAL SCHEDULING APPROACH

In this section, we present and discuss an intelligent MOSA for a set of thermostatic devices to achieve the solutions to the aforementioned MOP. The proposed MOSA is essentially a graph-based solution, similar to that of [24]; however, there are significant differences because of the multiple conflicting objectives targeted in this article. We start with describing the graph abstraction of the problem and, subsequently, describe ways to obtain the schedule that minimizes both average thermal discomfort and power fluctuation.

A. Graph Abstraction of the Scheduling Problem

With the given set of aforementioned inputs, we create a directed weighted graph $G = (V, E)$, whose each node $v_p \in V$ represents a tuple of possible temperature values for all the n devices, given as $v_p = \langle T_1^p, T_2^p, \dots, T_n^p \rangle$. A directed edge $e_{pq} \in E$ represents a transition from v_p to v_q . The temperature changes for devices from a node v_p to v_q depend on whether they are switched ON or OFF during the transition and is calculated using (2) and (3), respectively. An edge from v_p to v_q is labeled with an edge weight $\mathbb{W}_{pq} = \langle \mathbb{P}_{pq}, \mathbb{D}_{pq} \rangle$, where \mathbb{P}_{pq} describes the total power consumption deviation from the given set-point for the transition e_{pq} , mathematically expressed as

$$\mathbb{P}_{pq} = |P_{\text{tot}}(e_{pq}) - P^*|. \quad (10)$$

The second parameter in the edge weight is \mathbb{D}_{pq} that represents the square sum of the total discomfort for all the devices during

that transition. This is one of the significant difference of this model than the one in [24], where the edge weights in the graph has scalar weights (the total power consumption value). For an edge $e_{pe} \in E$ from v_p to v_q , each of the n devices will have a certain discomfort value based on the target node temperature, which we consider as a vector: $[D^{T_1^q}, D^{T_2^q}, \dots, D^{T_n^q}]$, where $D^{T_i^q}$ is computed based on T_i^q using (7). From this vector, the quantity \mathbb{D}_{pq} is computed as follows:

$$\mathbb{D}_{pq} = \left(D^{T_1^q}\right)^2 + \left(D^{T_2^q}\right)^2 + \dots + \left(D^{T_n^q}\right)^2. \quad (11)$$

Starting with the initial vertex $v_0 = \langle T_1^0, T_2^0, \dots, T_n^0 \rangle$, we explore all the unique nodes that can be created satisfying the temperature constraint of each of the ACs, as depicted in (1). After the nodes are constructed, we create possible connections among them using directed edges with appropriate edge weights. More precisely, a directed edge is created from a node v_p to v_q only if, $\forall T_i^p \in v_p$, we have $T_i^{p+1} \in v_q$ such that (1) is satisfied. However, in the worst case there can be a total of 2^n edges going out of a node v_p that satisfy (1) depending on the switching status during the transitions, which in return would result in an intractable graph size. Therefore, to keep the graph size manageable, we put a limit on the absolute power consumption deviation for each transition for all the ACs from P^* . This keeps the total power consumption within a known bounded value where the power fluctuation can only vary within the upper and lower limits of the bounded space. Let P_{lt} be that limit for every transition, and therefore, during the transition from v_p to v_q , the total power consumption \mathbb{P}_{pq} is given as

$$P_{\text{low}} \leq \mathbb{P}_{pq} \leq P_{\text{high}} \quad (12)$$

where $P_{\text{low}} = P^* - P_{lt}$ and $P_{\text{high}} = P^* + P_{lt}$ are the lower and upper power consumption deviation limits.

Thus, the final graph that is constructed contains nodes whose temperature values for all the devices are within their respective TCBs and edges labeled with appropriate weights, satisfying both (1) and (12).

B. Scheduling for Optimized Power and Comfort

We devise a scheduling mechanism that can be used to operate the devices for significantly longer periods of time, for example, 16–18 h in case of shopping malls, 24×7 for commercial and industrial office buildings, data centers, etc. It is, therefore, viable for us to obtain a schedule that can be repeated over time while optimizing (9).

To obtain such a repeatable schedule, we utilize the graph constructed in Section IV-A, in which we look for a cycle. Since the graph contains only those nodes and edges that satisfy all the constraints, the presence of a cycle will ensure that there exists a schedule that can be repeated as per the temperature combinations in the nodes of the cycle, and thus, the devices can be scheduled to operate for a long period of time. Detecting a cycle, therefore, depicts the feasibility of the given specifications.

Once such a graph is obtained, we seek to find out a cycle from the graph that optimizes (9), i.e., the cycle for which both average power consumption deviation and average discomfort is minimized. The average weight of a cycle (C) is computed

as the average of the individual weight components of all the edges in the cycle, i.e., $\overline{\mathbb{W}}_C = \langle \rho_C = \frac{\sum_{e \in C} \mathbb{P}_e}{|C|}, \delta_C = \frac{\sum_{e \in C} \mathbb{D}_e}{|C|} \rangle$, where $|C|$ indicates length of the cycle (measured by the number of edges), e is an edge in the cycle, and \mathbb{P}_e and \mathbb{D}_e are the weights for e . However, there may be multiple cycles in the feasible graph and selecting one of them is difficult since the targeted objectives are conflicting in nature and we are looking for a tradeoff solution. To address this issue and to determine the best cycle, we utilize the concept of nondominance in graphs similar to *Definition 1*, and define a nondominated cycle as below.

Definition 2: A cycle C_a , having average weight $\overline{\mathbb{W}}_{C_a} = \langle \rho_a, \delta_a \rangle$ is said to dominate another cycle C_b with an average weight of $\overline{\mathbb{W}}_{C_b} = \langle \rho_b, \delta_b \rangle$ if any one of the following conditions hold true.

- 1) $\rho_a < \rho_b$ and $\delta_a < \delta_b$.
- 2) $\rho_a \leq \rho_b$ and $\delta_a < \delta_b$.
- 3) $\rho_a < \rho_b$ and $\delta_a \leq \delta_b$.

Thus, a nondominated cycle is the one that minimizes both average power consumption deviation and average discomfort simultaneously, and therefore, is our desired cycle. Since there can be more than one such nondominated cycles in the graph, each one of them is a Pareto optimal solution [34], and here, we seek to find all of them to form the complete Pareto optimal solution set.

There exists a number of algorithms to find out all the cycles in a graph and a comprehensive list of them all has been presented by Mateti and Deo in [35]. Here, we utilize Johnson's all elementary circuit finding algorithm for directed graphs [36], one of the efficient and fastest known algorithms for obtaining cycles in a graph. Johnson's algorithm takes a directed graph as input, where it considers the nodes to be represented using integers "1 through n ." In a nutshell, the algorithm starts with an initial vertex s (generally the vertex with the least indexed number) and proceeds to find out all the elementary paths from s until the initial vertex is reached again, resulting in a cycle. Thus, all the cycles are constructed in the subgraph induced by s and nodes "larger than s " in the ordering of the nodes. The algorithm avoids duplicate cycles by blocking a vertex v_p when it is added to some elementary path beginning s . v_p is kept blocked till all the paths from v_p to s intersects the current elementary path at a node other than s . Furthermore, a node does not become an initial node for constructing elementary paths unless it is the least indexed vertex in at least one elementary cycle. Thus, finally what we have is a set of unique elementary cycles in the original directed graph.

On the graph constructed in Section IV-A, we apply Johnson's algorithm to determine all the cycles in it, where each cycle have their individual average weights. From these cycles, we obtain all the nondominant cycles by a one-to-one comparison among them. This eventually leads to our desired set of cyclic schedules, minimizing both average power consumption deviation from a given set-point and average discomfort.

The proposed approach is summarized in Algorithm 1. In Step 1, the comfort zones for each of the given ACs are determined based on their supplied TCBs. Step 2 handles the graph abstraction phase discussed above, where nodes represent temperature combinations of the devices, and an edge between

Algorithm 1: Proposed MOSA.

Input: The set of n ACs with their individual power consumption values P_i and thermal comfort band $[T_i^-, T_i^+]$, power consumption set-point (P^*).

Output: **ND:** the set of Pareto optimal (non dominated) solutions, where each solution represent the average cost of a cyclic schedule.

- 1: From the given thermal comfort band for each of the ACs, calculate the comfort zones.
 - 2: Construct a graph G having nodes representing temperature combinations of the devices and connecting them are the edges representing a transition from one temperature combination to another, guided by (2) and (3) while satisfying (1) and (12). Each edge $e_{pq} \in E$ is labeled with weight $\mathbb{W}_{pq} = \langle \mathbb{P}_{pq}, \mathbb{D}_{pq} \rangle$, where \mathbb{P}_{pq} is obtained using (10) and \mathbb{D}_{pq} is obtained using (11).
 - 3: On G , detect the presence of a cycle.
 - 4: If no cycle found, input specifications are infeasible. Exit.
 - 5: Else, apply Johnson's algorithm to obtain the set of all the possible cycles $C = C_1, C_2, \dots, C_c$.
 - 6: For each $C_i \in C$, do:
 - 7: Obtain the average cost parameters $\langle \rho_i, \delta_i \rangle$.
 - 8: For each cycle $C_i \in C$, do:
 - 9: Compare the cost parameters $\langle \rho_j, \delta_j \rangle$ of C_j with $\langle \rho_i, \delta_i \rangle$, $\forall C_j \in C$ and $C_j \neq C_i$.
 - 10: If $\langle \rho_i, \delta_i \rangle$ is non dominated as per *Definition 2*:
 - 11: Add C_i in **ND**, i.e., **ND** = **ND** \cup C_i .
 - 12: Return **ND**.
-

any nodes is constructed if the transition satisfies (1) and (12). The transition indicating the rate of change in temperature values between two adjacent nodes is guided by (2) and (3). We label the edges with appropriate power consumption and discomfort values using (10) and (11). In step 3, the proposed algorithm detects the presence of a cycle, and the process halts if no cycle is found in step 4, considering the specifications to be infeasible. However, for feasible specifications, the algorithm executes Step 5 to obtain all the cycles in the graph. Once we have the set of all possible cycles in $G(V, E)$, Steps 6 through 11 determine all the nondominated cycles that are stored in the list **ND**. Finally in Step 12, the list of all the nondominated cycles are returned in the form of **ND**.

C. Time Complexity and Optimality of MOSA

The two primary phases involved in the proposed MOSA are constructing the graph and obtaining all the cycles in the graph. Since each of the nodes in the graph is a combination of temperatures for all the devices, in the worst case, the total number of valid combinations can be $|T_i^- - T_i^+|^n$, where $|T_i^- - T_i^+|$ indicates the total number of temperature points within the TCB and n is the number of devices. Therefore, the graph size can also grow exponentially with the increase in the number of devices and TCB.

In the second stage, we utilize Johnson's algorithm to obtain all the cycles, the time complexity of which is known

to be $O((V + E)(c + 1))$, where V is the number of nodes, E is the number of edges, and c is the number of cycles. In order to obtain all the nondominated cycles, we iterate over all the cycles that requires $O(c^2)$ computations. Thus, the overall time complexity to obtain all the nondominated cycles is $O((V + E)(c + 1) + c^2)$.

The optimality of the proposed MOSA is guaranteed due to the fact that it determines the nondominated cycles by extracting all the cycles from the graph, and the underlying algorithm is Johnson's algorithm, which is known to be optimal [36]. Therefore, the proposed MOSA produces optimal results depending on the input graph.

In order to produce the nondominated cycles, MOSA utilizes Johnson's algorithm, which converges after all the cycles in the graph are obtained. The proposed MOSA, therefore, converges intrinsically.

V. INTELLIGENT IMPROVEMENTS OVER MOSA

One major issue with the aforesaid MOSA is its exponential time complexity, which is predominantly due to the possibility of an exponential number of cycles in a graph. Moreover, the size of the graph varies exponentially with the number of devices. Although the graph construction phase for the optimal solution is inevitable, extracting out all the possible cycles from the graph may be avoided since not all of them are nondominated cycles. In order to address this issue of MOSA, in this article, we present two intelligent branch and bound techniques that exploit the key properties of the graph to avoid exploration of cycles that are dominated, thereby pruning the search space and improving the efficiency of the optimal approach, while producing the same set of Pareto optimal solutions as MOSA.

The improvement schemes are elegant extensions of MOSA, which work on the same principle of Johnson's algorithm with some suitable modifications to address the issue of exponential computational overhead in case of Johnson's algorithm. Similar to Johnson's algorithm, both the improvement schemes maintain a set of data structures including a stack to keep track of the path starting from the initial node, and a block set containing the nodes that are currently being explored. The schemes also maintain two lists: OPEN and CLOSED. The OPEN list is used to store the nodes that are candidates for expansion and have the possibility to produce nondominated cycles. The CLOSED list consists of the nodes that have been expanded already. However, instead of exploring all the nodes, as is done in Johnson's algorithm, these improvement schemes use branch- and bound-based evaluation functions at each node to estimate the possibility of a nondominated cycle ahead of that node.

Let us consider that starting from some initial node s in the graph containing m nodes, we have traversed k edges to reach the node r , accruing a total of power consumption deviation TC_P^r and total cost due to discomfort TC_D^r . We employ these information and other graph properties to estimate a lower bound on the average cost of a nondominated cycle at r and compare it to the costs of already obtained nondominated cycles. Next, we describe two improvement schemes that use branch and bound conditions to prune the search space. The first one is based on computing remaining minimum average (RMA) and

the second one is based on remaining cumulative minimum average (RCMA).

A. RMA-Based Improvement Scheme

This improvement scheme is based on estimating the minimum average cost of nondominated cycles, where nodes that may not result in an optimal solution are pruned. Similar to Johnson's algorithm, this scheme begins with some initial vertex s , and it tries to compute a lower bound on the average cost of a nondominated cycle based on the unexplored part of the graph and the average cost till the explored node r , $\bar{P}_r = \frac{TC_P^r}{k}$ and $\bar{D}_r = \frac{TC_D^r}{k}$.

It starts with determining the minimum weight edge in the subgraph spanned by the unexplored nodes in the graph. Let P_{\min} be the minimum power consumption cost and D_{\min} be minimum discomfort cost among all the untraversed edges from nodes in the OPEN list. Then an estimation on the minimum average cost of power consumption deviation for a nondominated cycle via r can be $\frac{k\bar{P}_r + P_{\min}}{k+1}$, which can also be rewritten as $\bar{P}_r + \frac{P_{\min} - \bar{P}_r}{k+1}$ if cycle is obtained by adding only 1 edge after r . Likewise, the estimation would be $\bar{P}_r + \frac{2(P_{\min} - \bar{P}_r)}{k+2}$ if cycle is obtained by adding two edges after r , $\bar{P}_r + \frac{3(P_{\min} - \bar{P}_r)}{k+3}$ if cycle is obtained with three more edges after r , and so on. Considering the graph to have a total of m nodes, the estimation on the minimum average power consumption deviation if cycle is obtained at the last node would be $\bar{P}_r + \frac{(m-k)(P_{\min} - \bar{P}_r)}{m}$. Similarly, we estimate the minimum average cost of discomfort for a nondominated cycle at the node r .

In the above-mentioned lower bound estimation, the terms $\frac{1}{k+1}, \frac{2}{k+2}, \frac{3}{k+3}, \dots, \frac{m-k}{m}$ are increasing and \bar{P}_r is constant at the node r . Therefore, if $P_{\min} - \bar{P}_r > 0$, then the lower bound on power consumption deviation must be $\bar{P}_r + \frac{P_{\min} - \bar{P}_r}{k+1}$ and if $P_{\min} - \bar{P}_r < 0$, then we must have the lower bound estimation to be $\bar{P}_r + \frac{(m-k)(P_{\min} - \bar{P}_r)}{m}$ for the average power consumption deviation. Combining these, we can write the overall lower bound on the average cost of a nondominated cycle (C_r^{RMA}) at the node r as follows:

$$C_r^{\text{RMA}} = \langle P_{\text{est}}^r, D_{\text{est}}^r \rangle \quad (13)$$

where $P_{\text{est}}^r = \bar{P}_r + \min\left\{\frac{P_{\min} - \bar{P}_r}{k+1}, \frac{(m-k)(P_{\min} - \bar{P}_r)}{m}\right\}$ and $D_{\text{est}}^r = \bar{D}_r + \min\left\{\frac{D_{\min} - \bar{D}_r}{k+1}, \frac{(m-k)(D_{\min} - \bar{D}_r)}{m}\right\}$.

Therefore, exploration from node r is pruned if C_r^{RMA} in (13) is more than the already obtained best results, otherwise, we backtrack from r and all the paths leading from r remain unexplored. The same is carried out for all the nodes in the OPEN list. Since in the multiobjective space, there may exist more than one nondominated cost path leading from a node r , it becomes necessary to allow all such nondominated paths from r to obtain the complete Pareto optimal set. To capture multiple nondominated cycles during the course of the search, this scheme maintains a list of already obtained nondominated cycles along with their average costs, which is updated only if a new nondominated cycle is found.

B. RCMA-Based Improvement Scheme

This improvement scheme essentially operates similarly as that of the RMA improvement scheme; however, it offers a different perspective in obtaining the lower bound on the average cost of a nondominated cycle. After reaching the node r starting from the initial node s , this scheme obtains the least weight power consumption deviation (\mathbb{P}_i^{\min}) and discomfort (\mathbb{D}_i^{\min}) values among the outgoing edges for each of the remaining nodes in the OPEN list, including r . The obtained sets $\{\mathbb{P}_i^{\min}\}$ and $\{\mathbb{D}_i^{\min}\}$ are then sorted increasingly.

Let the sorted sequence of the least weight power consumption deviation values be $\mathbb{P}_j^{\min}, \mathbb{P}_{j+1}^{\min}, \dots, \mathbb{P}_{m-k}^{\min}$ and similarly the sorted sequence of discomfort is $\mathbb{D}_j^{\min}, \mathbb{D}_{j+1}^{\min}, \dots, \mathbb{D}_{m-k}^{\min}$. Once the sorted sequence is obtained, we perform a cumulative average sum of these weight components as follows: $\bar{P}_r + \frac{\mathbb{P}_j^{\min} - \bar{P}_r}{k+1}, \bar{P}_r + \frac{\mathbb{P}_j^{\min} + \mathbb{P}_{j+1}^{\min} - \bar{P}_r}{k+2}, \dots, \bar{P}_r + \frac{\mathbb{P}_j^{\min} + \mathbb{P}_{j+1}^{\min} + \dots + \mathbb{P}_{m-k}^{\min} - \bar{P}_r}{m}$ for power consumption deviation and $\bar{D}_r + \frac{\mathbb{D}_j^{\min} - \bar{D}_r}{k+1}, \bar{D}_r + \frac{\mathbb{D}_j^{\min} + \mathbb{D}_{j+1}^{\min} - \bar{D}_r}{k+2}, \dots, \bar{D}_r + \frac{\mathbb{D}_j^{\min} + \mathbb{D}_{j+1}^{\min} + \dots + \mathbb{D}_{m-k}^{\min} - \bar{D}_r}{m}$ for discomfort. The minimum of these cumulative sum of the components gives us a lower bound on the average cost of a nondominated cycle. More formally, the estimated lower bound on the average cost of a nondominated cycle ($\mathcal{C}_r^{\text{RCMA}}$) at node r by RCMA scheme is given as

$$\mathcal{C}_r^{\text{RCMA}} = \left\langle \min_j \left\{ \bar{P}_r + \frac{\sum_{j=1}^{m-k} \mathbb{P}_j^{\min} - \bar{P}_r}{k+j} \right\}, \min_j \left\{ \bar{D}_r + \frac{\sum_{j=1}^{m-k} \mathbb{D}_j^{\min} - \bar{D}_r}{k+j} \right\} \right\rangle. \quad (14)$$

The primary difference between the proposed two improvement schemes is the evaluation functions that they utilize in order to expand a node. Although both the schemes explore all the remaining edges in the graph, the RMA scheme considers the minimum of the individual costs among all the untraversed edges in (13), whereas RCMA scheme considers the minimum of the individual costs among the outgoing edges per node in (14).

Nevertheless, the basic iterations of both these schemes consist of selecting a node from the unexplored set of nodes, generation of its successors, and entering them into either the OPEN or CLOSED list. Using branch and bound conditions, these two schemes try to obtain lower bounds on the average cost for nondominated cycles to restrict the expansion of nodes that yield unnecessary cycles. It may be observed that if we omit the evaluation functions that are checked at every node, both these improvement schemes essentially boil down to Johnson's algorithm. Nonetheless, with the introduction of these evaluation functions, the number of nodes expanded is greatly reduced, since now, only those paths are explored that have the possibility of resulting in nondominated cycles, thereby reducing the computation time as compared to Johnson's algorithm.

Algorithm 2: Proposed GNSA.

Input: The directed graph $G(V, E)$ constructed in Section IV-A.

Output: **GND:** a set of greedy non dominated solutions, where each solution depicts a cycle.

▷ /*Special notations used in the algorithm*/

UN: the set of unexplored nodes, initially containing all the nodes V from $G(V, E)$.

ON: the set of open nodes, initially empty.

CN: the set of closed nodes, initially empty.

ξ : the set of non dominated edges, initially empty.

dfs (a): DFS traversal from the node a .

▷ /*Phase 1: Exploration*/

1: For each $v_i \in V$, do:

2: Obtain the set of non dominated outgoing edges

$\varepsilon_i = \{\bar{e}_{ij}\}$ from v_i .

3: Update $\xi = \xi \cup \varepsilon_i$.

▷ /*Phase 2: Obtaining desired cycles*/

4: Repeat the following, until **UN** is empty:

5: Select a node v_i from **UN**.

6: **dfs**(v_i)

7: Return **GND**.

▷ /*Description of the DFS traversal*/

8: **Procedure** **dfs**(v_i):

9: Update **ON** = **ON** \cup v_i and **UN** = **UN** \setminus v_i .

10: For each $\bar{e}_{ij} \in \xi$, do:

11: If the target node v_j from v_i is in **ON**:

12: Cycle is detected. Store the cycle in **GND**.

13: If the target node v_j from v_i is in **UN**:

14: **dfs**(v_j).

15: Update **CN** = **CN** \cup v_i .

16: Return.

VI. GREEDY NONDOMINATED SCHEDULING ALGORITHM

Although the improvement schemes can reduce the computational overhead of Johnson's algorithm, the total number of nondominated cycles in the graph can still be huge, making the approach difficult to implement in a real-world environment. In such a scenario, it is safe to assume that a method that produces results quickly, although suboptimal, is desirable since the time required to produce the optimal solutions is exponentially huge with respect to the number of devices. We, therefore, propose a new GNSA that produces approximated nondominated cycles efficiently, without the need to explore the complete graph and all the cycles.

The pseudocode of the proposed GNSA is shown in Algorithm 2. Instead of exploring all the cycles in the graph, it tries to obtain cycles by exploring only the nondominated edges. The algorithm primarily has two phases, each of them are described in details below.

Phase 1—Graph exploration: In this phase, GNSA processes the graph $G(V, E)$ constructed in Section IV-A and explores all the nondominated outgoing edges from each of the nodes in the graph. All the obtained nondominated outgoing edges are marked and stored in a list ξ , which is then forwarded to the

TABLE I
INPUT SPECIFICATIONS

Parameters	Ranges
Desired temperature	$[18 - 24] ^\circ C$
Initial temperature	$[18 - 24] ^\circ C$
Heat flow rate	$[18.7 - 24.4] W/m^2$
Thermal resistance	$[0.0096 - 0.028] (m^2 K)/W$
Heat capacity	$[1.2 - 3.5] J/K$
Ambient temperature	$22.5 ^\circ C$
Power consumption	$[10 - 20] \times 10^2 W$

next phase. The steps 1 through 3 in Algorithm 2 represent this phase.

Phase 2—Obtaining the nondominated cycles: To obtain the approximated nondominated cycles, the proposed GNSA utilizes the depth-first search (DFS) mechanism on the induced graph containing only the nondominating outgoing edges from Phase 1. The algorithm maintains a set **UN** that initially contains all the nodes from the graph G . It further maintains two more lists **ON** and **CN** in order to track the nodes that are to be visited and the nodes that have already been visited, respectively. At every iteration, the algorithm selects and removes a node v_i from the set **UN**, adds it to **ON**, and traverses the induced graph $G(V, \xi)$ from v_i using the DFS algorithm, indicated by $\text{dfs}(v_i)$ in Algorithm 2. Within $\text{dfs}(v_i)$, the algorithm recursively explores the graph from v_i and records a cycle whenever a node v_j during the traversal is found to be in **ON**. The algorithm maintains a set **GND** to track all the cycles encountered during the graph traversal. Finally, when v_i is explored completely, it is added into **CN**. The above-mentioned steps are repeated until **UN** is empty. The desired set of nondominated cycles is finally obtained from **GND**. In Algorithm 2, the Steps 4 to 7 depict Phase 2, whereas the DFS traversal procedure is shown in Steps 8 through 16.

The time complexity of the proposed GNSA depends on the number of nodes and the number of edges explored, which can be $O(V + E)$ at the worst on the graph $G(V, E)$. Once the node exploration phase is complete, it runs DFS to traverse through the set all the nodes on the graph, keeping track of all the cycles, which are effectively approximated nondominated cycles, while processing. Thus, the time complexity of this phase is $O(V + E)$. Therefore, the effective time complexity of the GNSA algorithm is $O(V + E)$. Thus, this heuristic strategy significantly reduces the time taken to obtain the cyclic schedules as compared to the optimal approach, and thereby, making it a highly desirable and efficient solution for real-world large-scale implementation. Since the number of nodes in the graph is finite, the execution of GNSA eventually halts when all the potential unexplored nodes are enumerated.

VII. EXPERIMENTAL RESULTS AND DISCUSSIONS

The performance of the scheduling methodologies proposed in this article has been evaluated using test cases generated from real-world parameters as input specifications, as listed in Table I, where the values of the parameters are in their typical range for office buildings in Asia and the U.S. [32], [37], [38]. The implementation was done in Java on a computer having Intel Xeon processor E5-2609 v3 (15 MB Cache, 1.90 GHz, 6 cores) with 132 GB of main memory.

TABLE II
RESULTS OBTAINED THROUGH PROPOSED MOSA

#Devices	Set-point	Graph size		#Time	#Cycles	#ND cycles
		# Nodes	# Edges			
10	100	241	1091	183.40	1917	13
15	150	996	14982	889.81	8556	29
20	200	5214	117186	2366.33	47061	117
25	250	10635	177739	2825.36	133914	103
30	300	18341	225678	3406.12	453429	129
35	350	24989	271998	3718.50	1389618	153
40	400	34835	358227	4431.87	3841973	88
45	450	41942	391646	4882.33	6209216	246
50	500	52746	439425	5397.20	10968303	134

Set-point = Power consumption set-point, #Nodes = average number of nodes, #Edges = average number of edges, #Time = average time in seconds, #Cycles = average number of cycles, #ND cycles = average number of nondominated cycles.

A. Evaluating MOSA and the Improvement Schemes

The MOSA, proposed in Section IV, has been experimented to schedule up to 50 devices, with a total of 500 different test cases. The proposed MOSA takes input from Table I and produces output in the form of cyclic schedules that have nondominated power consumption and discomfort parameters. The dynamics of the devices when switched ON or OFF vary according to (2) and (3). Since we want to minimize the power consumption deviation from a certain power consumption set-point, here for the experimental purpose, we have considered a feasible window of $\pm 5\%$ from the set-point. For example, if the power consumption set-point is 100, we allow the total power consumption to vary within the range [95, 105] and anything outside this range is considered to be infeasible. This is to ensure safety and reliability in grid operations as observed in real-life scenarios [39], [40]. Furthermore, we have approximated the temperatures of the devices to take values rounded off to the nearest 0.5 value, i.e., if the resulting temperature value is 23.34, we make it 23.5. This is adapted to restrict the number of temperature values that a device can take, and consequently, restricting the graph size and computational overhead.

Table II lists the summary of the results obtained using MOSA, depicting the average size of the graphs in terms of the number of nodes and edges, and the average number of nondominated cycles. It also shows the input power consumption set-point for each set of devices, which is considered ten times the number of appliances. For a particular number of devices, the proposed optimal scheduling approach obtains the complete Pareto optimal set of solutions (the nondominated cycles), which can then be used to operate the devices. The advantage of MOSA is that it obtains the complete Pareto optimal set, which provides high flexibility and an in-depth decision-making process for the controller to opt for the best of the available solutions, since now it has a large number of equally good solutions at its disposal and can choose any one of them as per the requirements.

As can be observed from Table II, the average time required increases enormously with the increase in the number of devices, and as the number of devices reaches a mark of 50, the computation time becomes unacceptable for implementing MOSA in a real-world scenario. This is also justified by the size of the graph, represented in terms of the number of nodes and edges in Table II. However, the intelligent improvement schemes on the optimal approach proposed in this article address this issue by producing results significantly quicker as shown in Fig. 2. Results presented

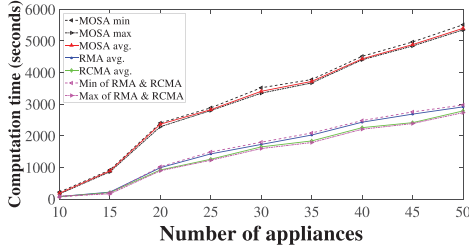


Fig. 2. Comparison of execution time among different algorithms.

in Fig. 2 clearly depicts the efficacy of the improvement schemes in reducing the time taken to produce Pareto optimal results for up to 50 devices. The dotted lines in black indicate the maximum and minimum amount of time taken by MOSA, whereas the solid red line represents the average time taken. On the other hand, the solid lines in blue and green represent the time taken by the RMA scheme and RCMA scheme, respectively. The dotted lines in magenta indicate the range of the required computation time by the improvement schemes.

One can observe from Fig. 2 that the range of computation time for the improvement schemes is significantly less than that of MOSA, even though all of them produce the same Pareto optimal set of solutions. In fact, RMA and RCMA schemes have been able to reduce the time consumption by more than 48.5% as compared to MOSA, which indicates the efficiency of the two improvement schemes without compromising the quality of solutions.

B. Evaluating GNSA

Here, we evaluate the performance of the proposed GNSA. Being a multiobjective solution to the scheduling problem at hand, GNSA results in a collection of vectors instead of a scalar quantity, which complicates the process of evaluating the solutions and understanding the quality of solutions with respect to the optimal solution set. In this regards, we utilize the existing performance evaluation and quality measures for different nondominated solution sets. Although a large number of such performance metrics are available in the literature [41], [42], here in particular, we use the following quality measures in this article: *Error Ratio (ER)*, *Generational Distance (GD)*, *Maximum Pareto Front Error (MPFE)*, *Schott's Spacing metric (SS)*, and *Hyper-volume ratio (HR)*. The specific reason behind choosing these metrics as performance measures is because of their Pareto compatibility, which try to minimize the distance of the results obtained from the Pareto optimal set. Thus, lower the values of these metrics, the closer the results are to the Pareto optimal solutions.

The aforementioned performance metrics have been utilized to evaluate GNSA, results of which are listed in Table III that comprises of results averaging over all the 500 test cases. All the performance metrics consider the Pareto optimal set as their reference nondominated optimal set, which we have already obtained using MOSA, RMA scheme, and RCMA scheme. The optimal values for the various performance metrics, listed in Table III, depict the desired distance of results obtained by the greedy algorithm from the optimal set. For example, the optimal

TABLE III
EVALUATION OF GNSA ALGORITHM USING VARIOUS PERFORMANCE METRICS

#devices	ER	GD	MPFE	SS	HR
	Optimum value				
	0	0	0	0	1
10	0.20	0.73	1.15	0.67	1.12
15	0.33	0.65	1.34	0.82	1.05
20	0.37	0.52	1.15	0.73	1.14
25	0.24	0.48	1.38	1.16	1.17
30	0.16	0.53	1.29	1.09	1.06
35	0.25	0.67	1.32	0.91	1.21
40	0.27	0.67	1.46	1.02	1.14
45	0.18	0.86	1.28	0.96	1.07
50	0.20	0.58	1.33	1.04	1.16

value for ER metric is 0, which indicates that the closer the values obtained by GNSA are to 0 using ER, the better they are in terms of producing Pareto optimal solutions with respect to the ER metric. Similarly, we can evaluate the performance of GNSA using the other metrics with respect to their optimal values as listed in the table.

Table III depicts the fact that the performance of GNSA has been closer to the optimal methodologies for the majority of the performance metrics. In particular, the average ER has varied within a significantly small range of [0.16–0.37] for up to 50 numbers of devices. This indicates that the number of unwanted solutions obtained by GNSA is significantly less and about 70% of the cycles obtained are in the Pareto optimal set. Similarly, the average GD for GNSA algorithm has varied within a range of [0.48–0.86], which also portrays its performance in obtaining closer to optimal results. Similar behavior is also observed using MPFE and HR metrics. Although for SS metric, the performance of GNSA is observed to be weaker in comparison to the other metrics, overall, GNSA has been able to produce most of the Pareto optimal solutions as confirmed by most of the performance metrics, making it a highly effective greedy scheduling algorithm.

C. Solving MOP Using WSM

It is to be noted that during the graph construction phase, if we allow the edges to have only one of the parameters as weights (power consumption deviation or discomfort), then the problem of scheduling being studied can essentially be solved using the KMMC [25]. However, since we are targeting to address both the objectives simultaneously, KMMC does not yield the desired solutions. Nevertheless, a traditional way of solving any MOP has been to solve it as a single objective optimization problem by assigning weights to the individual objectives, e.g., the WSM [43]. With the help of WSM, we can now apply KMMC for the targeted scheduling problem and use the obtained results to further assess the performance of GNSA. The WSM is also useful to arrive at a particular solution out of the complete Pareto front of solutions [29], [30], [44]. To study such a scenario for the algorithms proposed in this article, we define a unified cost function (UCF) for a schedule/cycle (\mathcal{C}), incorporating the provision of prioritizing the two objectives over each other, which is mathematically expressed as

$$\text{UCF} = \mathcal{W}_p \times \overline{P_C} + \mathcal{W}_{dc} \times \overline{DC_C} \quad (15)$$

where $\overline{P_C}$ is the overall average power deviation and $\overline{DC_C}$ is the overall average discomfort, for \mathcal{C} . $\mathcal{W}_p \geq 0$ and $\mathcal{W}_{dc} \geq 0$

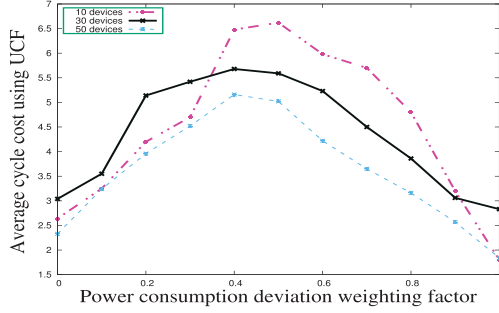


Fig. 3. Performance evaluation of GNSA using UCF.

are the weighting factors for power deviation and discomfort, respectively, and $\mathcal{W}_p + \mathcal{W}_{dc} = 1$. According to these weighting factors for the two objectives, the decision maker can now choose the best one among the obtained multiple Pareto optimal solutions. For example, consider two Pareto optimal cyclic schedules, where the first schedule has an average power consumption deviation of 3 and discomfort value of 2.5, whereas the second schedule has an average power consumption deviation of 5 and a discomfort value of 1. Assume that the user provides a weight of 0.3 to power consumption deviation and 0.7 to discomfort. Then the UCF for the first schedule will be: $0.3 \times 3 + 0.7 \times 2.5 = 2.65$, and UCF for the second schedule is: $0.3 \times 5 + 0.7 \times 1 = 2.2$. Since our objectives are of minimization type, we select the schedule having the least UCF, i.e., the second schedule in this example. In the cases of multiple such cycles having the least UCF, any one of them may be selected as the final schedule.

We start utilizing UCF to evaluate the performance of GNSA. Given certain values for \mathcal{W}_p and \mathcal{W}_{dc} , the graph constructed in Section IV-A will have edges with scalar values now. This graph is then fed to GNSA to obtain the minimum mean cyclic schedule. The weights are varied within 0 and 1 such that $\mathcal{W}_p + \mathcal{W}_{dc} = 1$. We have also normalized the weighting parameters for the edges to take values within [0–5] for simplicity. Following this procedure, we obtained results for all the 500 test cases and evaluated the performance of the GNSA algorithm. Fig. 3 shows results for three sets of devices (i.e., number of devices = 10, 30, 50), where the x -axis represents \mathcal{W}_p and the y -axis represents UCF. It can be observed from the figure that as the weighting factors tend to be equal (i.e., $\mathcal{W}_p \rightarrow \mathcal{W}_{dc}$ or vice versa), the average UCF, as per (15) has been maximum for almost all the test cases. This reveals the fact that cost increases when we target multiple objectives with equal weights. However, as we focus on only one particular objective, the cost tends to decrease. We have further observed that out of the two objectives, giving higher priority to \mathcal{W}_p results in a lesser overall cost. Thus, efficient power consumption management yields a better result for the user in terms of financial savings, while compromising on the comfort level a bit. It is worth mentioning that despite of its own drawbacks [45], we have observed that a majority of the results produced using the above-mentioned WSM scheme (the UCF) are from the “knee” zone of the Pareto front, which is the desired solution zone for MOPs. This observation has encouraged us in opting for the UCF to evaluate GNSA in comparison to KMMC.

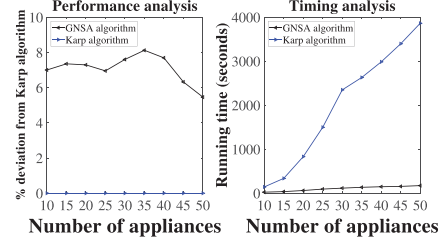
Fig. 4. Comparison of GNSA with Karp's algorithm for $\mathcal{W}_p = \mathcal{W}_{dc} = 0.5$.

Fig. 4 presents the comparative analysis of GNSA with that of KMMC in terms of % deviation in obtaining a cyclic schedule with minimum mean cost, and average running time, for a case when $\mathcal{W}_p = \mathcal{W}_{dc} = 0.5$. GNSA, being a heuristic, resulted in higher average costs of cycles than that of KMMC, which is known to be optimal. In particular, the % deviation of GNSA varied within [5.46–8.12]% as compared to KMMC and the average deviation has been noted to be 7.086% only, for this particular case. However, the advantage of GNSA over KMMC is realized in the timing analysis, where KMMC takes significantly longer time to produce results than that of GNSA. The time difference becomes notable for larger number of devices. We can also observe that the rate of computation time increase for GNSA is much smaller than that for KMMC. For instance, the average computation time taken by KMMC to schedule 50 ACs is in the order of hours, whereas GNSA takes only 172.34 s, as can be observed in Fig. 4. Similar behavior has been observed for other weight combinations as well. It is worth mentioning that computation time depends on the size of the graph, which in turn depends on the number of devices, and the running time computation shown in Fig. 4 does not include the time for graph construction. Nevertheless, it can be concluded that, for a relatively smaller number of ACs (or smaller graph sizes), we may use KMMC, but for a substantial number of devices, GNSA is more efficient in obtaining cyclic schedules.

VIII. CONCLUSION AND FUTURE WORK

In this article, we have addressed the problem of scheduling thermostatically devices with the objective to minimize power fluctuation and maximize users' thermal comfort. We have proposed an optimal scheduling strategy by modeling the problem into a graph and obtaining all the nondominated cycles from it, corresponding to the Pareto optimal front. Furthermore, we have also proposed a couple of intelligent improvement schemes and a fast greedy algorithm to obtain solutions quickly. The proposed methods have been validated using test cases generated from a real-world scenario, where the improvement schemes have been able to reduce the computation overhead of the optimal strategy by more than 48.5%, and the greedy mechanism has been observed to yield the majority of its solutions from the Pareto optimal solution set. Comparative performance analysis of the greedy approach was also carried out with that of KMMC for graphs with scalar edge weights. A possible extension of this article is to optimize the power consumption set-point, which is our future work. The system parameters are assumed to be deterministic in this article. Developing a model without

such constraints along with the consideration of dynamic load variation and an integrated power-temperature model is a part of our future work.

REFERENCES

- [1] "Energy Consumption in U.S. Residential and Commercial Buildings," 2016. [Online]. Available: <https://www.eia.gov/tools/faqs/>
- [2] "International Energy Outlook 2017, Energy Information Administration, U.S.," [Online]. Available: <https://www.eia.gov/outlooks/ieo/>
- [3] "Annual energy Outlook 2017, Energy Information Administration, U.S.," [Online]. Available: <https://www.eia.gov/outlooks/aeo/>
- [4] N. Lu and S. Katipamula, "Control strategies of thermostatically controlled appliances in a competitive electricity market," in *Proc. IEEE Power Eng. Soc. Gen. Meeting*, 2005, pp. 202–207.
- [5] N. Lu, "An evaluation of the HVAC load potential for providing load balancing service," *IEEE Trans. Smart Grid*, vol. 3, no. 3, pp. 1263–1270, Sep. 2012.
- [6] G. Karmakar, A. Kabra, and K. Ramamritham, "Coordinated scheduling of thermostatically controlled real-time systems under peak power constraint," in *Proc. IEEE 19th Real-Time Embedded Technol. Appl. Symp.*, Apr. 2013, pp. 33–42.
- [7] X. Wu, X. Hu, X. Yin, and S. J. Moura, "Stochastic optimal energy management of smart home with PEV energy storage," *IEEE Trans. Smart Grid*, vol. 9, no. 3, pp. 2065–2075, May 2018.
- [8] X. Wu, X. Hu, Y. Teng, S. Qian, and R. Cheng, "Optimal integration of a hybrid solar-battery power source into smart home nanogrid with plug-in electric vehicle," *J. Power Sources*, vol. 363, pp. 277–283, 2017.
- [9] H. Chang, W. Chiu, H. Sun, and C. Chen, "User-centric multiobjective approach to privacy preservation and energy cost minimization in smart home," *IEEE Syst. J.*, vol. 13, no. 1, pp. 1030–1041, Mar. 2019.
- [10] O. Tan, J. Gomez-Vilardebo, and D. Gunduz, "Privacy-cost trade-offs in demand-side management with storage," *IEEE Trans. Inf. Forensics Security*, vol. 12, no. 6, pp. 1458–1469, Jun. 2017.
- [11] A. Mohsenian-Rad, V. W. S. Wong, J. Jatskevich, R. Schober, and A. Leon-Garcia, "Autonomous demand-side management based on game-theoretic energy consumption scheduling for the future smart grid," *IEEE Trans. Smart Grid*, vol. 1, no. 3, pp. 320–331, Dec. 2010.
- [12] Q. Wang, X. Liu, J. Du, and F. Kong, "Smart charging for electric vehicles: A survey from the algorithmic perspective," *IEEE Commun. Surveys Tut.*, vol. 18, no. 2, pp. 1500–1517, Second quart. 2016.
- [13] A. Mondal, S. Misra, and M. S. Obaidat, "Distributed home energy management system with storage in smart grid using game theory," *IEEE Syst. J.*, vol. 11, no. 3, pp. 1857–1866, Sep. 2017.
- [14] V. Pradhan, V. S. K. M. Balijepalli, and S. A. Khaparde, "An effective model for demand response management systems of residential electricity consumers," *IEEE Syst. J.*, vol. 10, no. 2, pp. 434–445, Jun. 2016.
- [15] A. Safdarian, M. Fotuhi-Firuzabad, and M. Lehtonen, "A distributed algorithm for managing residential demand response in smart grids," *IEEE Trans. Ind. Informat.*, vol. 10, no. 4, pp. 2385–2393, Nov. 2014.
- [16] J. J. Shah, M. C. Nielsen, T. S. Shaffer, and R. L. Fittro, "Cost-optimal consumption-aware electric water heating via thermal storage under time-of-use pricing," *IEEE Trans. Smart Grid*, vol. 7, no. 2, pp. 592–599, Mar. 2016.
- [17] D. D. Martini, G. Benetti, M. L. D. Vedova, and T. Facchinetti, "Adaptive real-time scheduling of cyber-physical energy systems," *ACM Trans. Cyber-Phys. Syst.*, vol. 1, no. 4, pp. 20:1–20:25, Sep. 2017.
- [18] A. Jindal, N. Kumar, and J. J. P. C. Rodrigues, "A heuristic-based smart HVAC energy management scheme for university buildings," *IEEE Trans. Ind. Informat.*, vol. 14, no. 11, pp. 5074–5086, Nov. 2018.
- [19] Y. F. Du, L. Jiang, C. Duan, Y. Z. Li, and J. S. Smith, "Energy consumption scheduling of HVAC considering weather forecast error through the distributionally robust approach," *IEEE Trans. Ind. Informat.*, vol. 14, no. 3, pp. 846–857, Mar. 2018.
- [20] N. Radhakrishnan, S. Srinivasan, R. Su, and K. Poolla, "Learning-based hierarchical distributed HVAC scheduling with operational constraints," *IEEE Trans. Control Syst. Technol.*, vol. 26, no. 5, pp. 1892–1900, Sep. 2018.
- [21] S. Bashash and H. K. Fathy, "Modeling and control of aggregate air conditioning loads for robust renewable power management," *IEEE Trans. Control Syst. Technol.*, vol. 21, no. 4, pp. 1318–1327, Jul. 2013.
- [22] C. H. Wai, M. Beaudin, H. Zareipour, A. Schellenberg, and N. Lu, "Cooling devices in demand response: A comparison of control methods," *IEEE Trans. Smart Grid*, vol. 6, no. 1, pp. 249–260, Jan. 2015.
- [23] R. Kamyar and M. M. Peet, "Optimal thermostat programming for time-of-use and demand charges with thermal energy storage and optimal pricing for regulated utilities," *IEEE Trans. Power Syst.*, vol. 32, no. 4, pp. 2714–2723, Jul. 2017.
- [24] N. Chakraborty, A. Mondal, and S. Mondal, "Intelligent scheduling of thermostatic devices for efficient energy management in smart grid," *IEEE Trans. Ind. Informat.*, vol. 13, no. 6, pp. 2899–2910, Dec. 2017.
- [25] R. M. Karp, "A characterization of the minimum cycle mean in a digraph," *Discrete Math.*, vol. 23, no. 3, pp. 309–311, 1978.
- [26] P. Li, R. Li, Y. Cao, D. Li, and G. Xie, "Multiobjective sizing optimization for island microgrids using a triangular aggregation model and the levy-harmony algorithm," *IEEE Trans. Ind. Informat.*, vol. 14, no. 8, pp. 3495–3505, Aug. 2018.
- [27] G. H. Shakouri and A. Kazemi, "Multi-objective cost-load optimization for demand side management of a residential area in smart grids," *Sustain. Cities Soc.*, vol. 32, pp. 171–180, 2017.
- [28] D. Li, W. Chiu, H. Sun, and H. V. Poor, "Multiobjective optimization for demand side management program in smart grid," *IEEE Trans. Ind. Informat.*, vol. 14, no. 4, pp. 1482–1490, Apr. 2018.
- [29] S. Sudeng and N. Wattanapongsakorn, "A decomposition-based approach for knee solution approximation in multi-objective optimization," in *Proc. IEEE Congr. Evol. Comput.*, Jul. 2016, pp. 3710–3717.
- [30] W. Chiu, G. G. Yen, and T. Juan, "Minimum Manhattan distance approach to multiple criteria decision making in multiobjective optimization problems," *IEEE Trans. Evol. Comput.*, vol. 20, no. 6, pp. 972–985, Dec. 2016.
- [31] ASHRAE Standard 55. Thermal Environmental Conditions for Human Occupancy. 2013. [Online]. Available: <https://bit.ly/2T5nvZf>
- [32] S. Manu, Y. Shukla, R. Rawal, L. E. Thomas, and R. de Dear, "Field studies of thermal comfort across multiple climate zones for the subcontinent: India model for adaptive comfort (IMAC)," *Building Environ.*, vol. 98, pp. 55–70, 2016.
- [33] P. Dasgupta, P. Chakrabarti, and S. DeSarkar, "Multiobjective heuristic search in and/or graphs," *J. Algorithms*, vol. 20, no. 2, pp. 282–311, 1996.
- [34] B. S. Stewart and C. C. White, III, "Multiobjective a^* ," *J. ACM*, vol. 38, no. 4, pp. 775–814, Oct. 1991.
- [35] P. Mateti and N. Deo, "On algorithms for enumerating all circuits of a graph," *SIAM J. Comput.*, vol. 5, no. 1, pp. 90–99, 1976. [Online]. Available: <https://doi.org/10.1137/0205007>
- [36] D. Johnson, "Finding all the elementary circuits of a directed graph," *SIAM J. Comput.*, vol. 4, no. 1, pp. 77–84, 1975. [Online]. Available: <https://doi.org/10.1137/0204007>
- [37] C. Peng and Z. Wu, "In situ measuring and evaluating the thermal resistance of building construction," *Energy Buildings*, vol. 40, no. 11, pp. 2076–2082, 2008.
- [38] M. Vanouni and N. Lu, "A reward allocation mechanism for thermostatically controlled loads participating in intra-hour ancillary services," *IEEE Trans. Smart Grid*, vol. 9, no. 5, pp. 4209–4219, Sep. 2018.
- [39] G. Huebner, D. Shipworth, I. Hamilton, Z. Chalabi, and T. Oreszczyn, "Understanding electricity consumption: A comparative contribution of building factors, socio-demographics, appliances, behaviours and attitudes," *Appl. Energy*, vol. 177, pp. 692–702, 2016.
- [40] B. Anderson, S. Lin, A. Newing, A. Bahaj, and P. James, "Electricity consumption and household characteristics: Implications for census-taking in a smart metered future," *Comput., Environ. Urban Syst.* (spatial analysis with census data: Emerging issues and innovative approaches), vol. 63, pp. 58–67, 2017.
- [41] J. Knowles and D. Corne, "On metrics for comparing nondominated sets," in *Proc. Congr. Evol. Comput.*, 2002, pp. 711–716.
- [42] K. Deb, *Multi-Objective Optimization Using Evolutionary Algorithms*. New York, NY, USA: Wiley, 2001.
- [43] E. Zitzler and L. Thiele, "Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach," *IEEE Trans. Evol. Comput.*, vol. 3, no. 4, pp. 257–271, Nov. 1999.
- [44] S. Bechikh, L. B. Said, and K. Ghédira, "Searching for knee regions in multi-objective optimization using mobile reference points," in *Proc. ACM Symp. Appl. Comput.*, 2010, pp. 1118–1125.
- [45] I. Das and J. E. Dennis, "A closer look at drawbacks of minimizing weighted sums of objectives for Pareto set generation in multicriteria optimization problems," *Structural Optim.*, vol. 14, no. 1, pp. 63–69, Aug. 1997.