A Scheduler for Smart Homes with Probabilistic User Preferences

Van Nguyen[1], William Yeoh[2], Tran Cao Son[1], Vladik Kreinovich[3(✉)], and Tiep Le[1,4]

[1] New Mexico State University, Las Cruces, NM, USA
{vnguyen,tson,tile}@cs.nmsu.edu
[2] Washington University in St. Louis, St. Louis, MO, USA
wyeoh@wustl.edu
[3] University of Texas at El Paso, El Paso, TX, USA
vladik@utep.edu
[4] Viome, Inc., Santa Clara, CA, USA
lebatiep@gmail.com

Abstract. Scheduling appliances is a challenging and interesting problem aimed at reducing energy consumption at a residential level. Previous work on appliance scheduling for smart homes assumes that user preferences have no uncertainty. In this paper, we study two approaches to address this problem when user preferences are uncertain. More specifically, we assume that user preferences in turning on or off a device are represented by Normal distributions. The first approach uses sample average approximation, a mathematical model, in computing a schedule. The second one relies on the fact that a scheduling problem could be viewed as a constraint satisfaction problem and uses depth-first search to identify a solution. We also conduct an experimental evaluation of the two approaches to investigate the scalability of each approach in different problem variants. We conclude by discussing computational challenges of our approaches and some possible directions for future work.

Keywords: Smart Home Scheduling · Probabilistic user preference

1 Introduction

Demand Side Management (DSM) is a portfolio of measures to improve the energy system at the consumption side. The initial goal of DSM is to cut the cost or energy consumption from the power grid, and that goal is a well-studied subject in smart grids (Department of Energy and Climate Change 2009a; 2009b). In recent years, with the rapid growth of technology and engineering such as Internet of Things, smart devices and ubiquitous computing, appliances in a household can communicate with each others. This creates a new environment in which each appliance can be considered as an agent, and the team of agents is able to collaborate to achieve a specific goal; for example, to execute a precomputed schedule of the appliances. It is not difficult to envision that in the near

M. Baldoni et al. (Eds.): PRIMA 2019, LNAI 11873, pp. 138–152, 2019.
https://doi.org/10.1007/978-3-030-33792-6_9

future, these agents can be controlled by a central server (or multiple servers) that can generate schedules of the appliances on the fly to improve the users' comfort while keeping energy consumption minimal.

In this paper, we aim at developing a scalable and efficient scheduling system for smart homes. The main difference between the proposed system and contemporary ones (see Sect. 6 for a discussion of current approaches) lies in that the former system will take user preferences into consideration under the assumption that the preferences are uncertain. This assumption is realistic since, as shown in the literature, user preferences could be approximately but not completely learned (see, e.g., [6,13,14,16]).

The present work could be considered as bridging the preference elicitation research and the development of smart home schedulers. This also provides a core component for the development of a comprehensive energy management system for smart grids in which individuals (e.g., homes, companies, etc.) can control their own energy consumption and, at the same time, coordinate with each other to lower the overall energy consumption, contributing to improved sustainability.

The main contributions of this paper are the following: (*i*) We provide a definition of a multi-objective *Smart Home Scheduling Problem* (SHSP) with probabilistic user preferences; (*ii*) We propose two approaches to solve SHSP, one based on *Sample Average Approximation* (SAA) and the other based on depth-first search; (*iii*) We present an empirical evaluation of the two approaches. Our empirical evaluation shows that the depth-first search based approach performs better than the sampling-based approach and thus provides a viable system for SHSP.

2 Smart Home Scheduling Problem

In this section, we define the *Smart Home Scheduling Problem* (SHSP) with probabilistic user preferences and its solutions.

Definition 1. *A* scheduling problem P *is a tuple* $\langle A, E, T, C, L, D\rangle$*, where*

- A *is a set of* ***appliances*** *(or* ***devices****), usually written as the set of integers* $\{1, \ldots, |A|\}$.
- $E = (e_1, e_2, ..., e_{|A|})$ *is a vector of positive real numbers, where each* e_i *represents the* ***energy consumption*** *of device* i.
- T *is a set of* ***time slots****, usually written as the set of integers* $\{1, \ldots, |T|\}$.
- $C = (c_1, c_2, ..., c_{|T|})$ *and* $L = (l_1, l_2, ..., l_{|T|})$ *are vectors of non-negative real numbers, where* c_i *and* l_i *represent the* ***cost*** *of 1 kWh and the maximum* ***permissible load*** *of all the devices at time* i*, respectively.*
- D *is an* $|A| \times |A|$ *matrix, called* ***dependency matrix****, each cell* $D(i, j)$ *represents a hard constraint between devices* i *and* j*. The relations/constraints can be one of the following types:*
  - ***before*** *(resp.* ***after****) means that the device* i *must be turned on before (resp. after) device* j.

- ***parallel*** *(resp.* ***not-parallel****) means that the device i must run in parallel (resp. must not run in parallel) with the device j.*
- ***nil*** *if the usage of the device i is independent from that of the device j.*

Intuitively, a scheduling problem P represents the problem of when to turn on devices. P is said to have no dependency if every element in D is **nil**. In this paper, without the loss of generality, we will assume that each device in P is turned on exactly once within $|T|$ time slots. For simplicity of the presentation, we will also assume that each device is active for only one time slot. The definitions and propositions in this paper can easily adapted for systems with appliances that work in multiple time slots (e.g., the washing machine runs for two hours) or need to be turn off (e.g., the light bulbs). Furthermore, we assume that the matrix D is symmetric in the following sense: (*i*) If $D(i,j) =$ **before** (resp. $D(i,j) =$ **after**), then $D(j,i) =$ **after** (resp. $D(j,i) =$ **before**); (*ii*) If $D(i,j) =$ **parallel** (resp. $D(i,j) =$ **not-parallel**), then $D(j,i) =$ **parallel** (resp. $D(j,i) =$ **not-parallel**); and (*iii*) If $D(i,j) =$ **nil**, then $D(j,i) =$ **nil**. A user preference for a scheduling problem is defined as follows.

Definition 2. *A* probabilistic user preference *over a scheduling problem $P = \langle A, E, T, C, L, D\rangle$ is a tuple $\mathcal{C} = \langle N, \alpha, \beta, \lambda\rangle$, where*

- *N is an $|A| \times |T|$ matrix, called* ***preference matrix****, where each cell $N(i,j)$ is a Normal distribution $\mathcal{N}(\mu_{ij}, \sigma_{ij})$ representing the probability distribution of the user's preference in turning the device i on at time slot j.*
- *α, called the* ***cumulative satisfaction threshold****, is a number representing the minimum acquired cumulative preference required by a user from a schedule.*
- *β is a number in the interval $[0,1]$ representing the* ***probability threshold****, which indicates the threshold of the probability that α will be achieved given a schedule in order for a user to accept that schedule.*
- *λ is a number indicating the* ***cost threshold*** *that a user could accept.*

Table 1 presents an example of preference matrix for $|A| = 4$ and $|T| = 3$.

Scheduling problems with probabilistic preferences are defined next.

Definition 3. *A* Smart Home Scheduling Problem *(SHSP) with probabilistic user preferences (or* p-scheduling problem*, for short) is a pair $(P, \mathcal{C})$, where P is a scheduling problem and $\mathcal{C}$ is a probabilistic preference over P.*

In this paper, for brevity, when we refer to SHSPs, we mean SHSPs with probabilistic user preferences.

We next define the notions of a schedule for P and when a schedule for P satisfies constraints in **dependency matrix** in D.

Definition 4. *Given a scheduling problem $P = \langle A, E, T, C, L, D\rangle$, a* ***schedule*** *for P is an $|A| \times |T|$ matrix H, where each cell $H(i,j)$ is either 0 (*off*) or 1 (*on*), representing the status of the device i at time slot j, and*

Table 1. An example of preference matrix N for $|A| = 4$ and $|T| = 3$. The two numbers in each cell of the table represent the Normal distribution for user preference to turn on an appliance at a time slot. For example, the preference to turn on device 1 at time 1 is $\mathcal{N}(10, 0.2)$.

		Time slots		
		1	2	3
Appliances	1	10, 0.2	9, 0.1	6, 0.15
	2	6, 0.01	8, 0.05	2, 0.78
	3	6, 0.51	7, 0.2	7, 0.99
	4	2, 0.41	6, 0.67	6, 0.09

- *if* $D(i,j) =$ ***before****, then* $H(i,t) = 1$ *implies that* $H(j,k) = 0$ *for every* $k \leq t$*;*
- *if* $D(i,j) =$ ***after****, then* $H(i,t) = 1$ *implies that* $H(j,k) = 0$ *for every* $k \geq t$*;*
- *if* $D(i,j) =$ ***parallel****, then* $H(i,t) = 1$ *implies that* $H(j,t) = 1$*; and*
- *if* $D(i,j) =$ ***not-parallel****, then* $H(i,t) = 1$ *implies that* $H(j,t) = 0$*.*

It is easy to see that the following observation holds under the assumption that each appliance is turned on exactly once among $|T|$ time slots.

Observation 1. *For each schedule* H *of a problem* $P = \langle A, E, T, C, L, D\rangle$

$$\forall i \in [1, |A|], \exists j \in [1, |T|] \text{ such that } H(i,j) = 1$$

$$\text{and } \forall j_1, j_2 \in [1, |T|], H(i, j_1) = H(i, j_2) = 1 \implies j_1 = j_2.$$

Definition 5. *The* ***Normal distribution of a schedule*** H *for a p-schedule problem* $(P, \mathcal{C})$*, with* $P = \langle A, E, T, C, L, D\rangle$ *and* $\mathcal{C} = \langle N, \alpha, \beta, \lambda\rangle$*, is defined by* $\mathcal{N}_H(\mu_H, \sigma_H)$ *, where*

- $\mu_H = \sum_{i=1}^{|A|} \sum_{j=1}^{|T|} \mu_{ij} \cdot H(i,j)$ *and*
- $\sigma_H = \sum_{i=1}^{|A|} \sum_{j=1}^{|T|} \sigma_{ij} \cdot H(i,j)$.

Intuitively, the distribution $\mathcal{N}_H(\mu_H, \sigma_H)$ represents the satisfactory of a user given the schedule H. Given a schedule H of a problem P, we define:

- the total energy consumption of all appliances at a given time slot j as:

$$e^H(j) = \sum_{i=1}^{|A|} e_i \cdot H(i,j). \tag{1}$$

- the cost at a given time slot j is

$$c^H(j) = c_j \cdot e^H(j). \tag{2}$$

Recall that the *complementary cumulative distribution function* (ccdf) of a real-valued random variable X is defined as $F_X(x) = Prob(X \geq x)$. We thus use $F_{\mathcal{N}_H(\mu_H, \sigma_H)}(\alpha)$ to represent the probability that the cumulative preference acquired by schedule H is greater than or equal to a cumulative satisfaction threshold α.

We are now ready to define the notion of satisfaction of a user preference by a schedule.

Definition 6. *Given a p-schedule problem* $\mathcal{P} = (P, \mathcal{C})$*, with* $P = \langle A, E, T, C, L, D \rangle$ *and* $\mathcal{C} = \langle N, \alpha, \beta, \lambda \rangle$ *over* P*, a schedule* H *satisfies* $\mathcal{C}$ *if it meets the following conditions:*

- ***Power Safety****:*

$$\forall j \in [1, |T|], e^H(j) \leq l_j. \tag{3}$$

- ***User Preference****:*

$$F_{\mathcal{N}_H(\mu_H, \sigma_H)}(\alpha) \geq \beta. \tag{4}$$

- ***Cost Efficiency****:*

$$f_c^H = \sum_{j=1}^{|T|} c^H(j) \leq \lambda \tag{5}$$

We say that a schedule H^* *is* optimal *if*

$$H^* = \underset{H \in \mathcal{H}}{\operatorname{argmin}} \sum_{j=1}^{|T|} c^H(j) \tag{6}$$

where $\mathcal{H}$ *is the set of schedules satisfying Conditions* (3)*,* (4)*, and* (5)*.*

Intuitively, Condition (3) requires that at any given time slot j, the total energy consumption of all appliances (i.e., $e^H(j)$) is at most equal to the given maximum load (i.e., l_j). Condition (4) states that the probability in which the cumulative preference acquired by H meets the cumulative satisfaction threshold (i.e., that accumulate preference is at least α) is at least β. Finally, Condition (5) indicates that the cost of the schedule must be at most λ.

In the next sections, we propose two methods to solve the SHSP. The first method is based on *Sample Average Assumption* (SAA) and the second method is based on depth-first search.

3 Solving SHSPs Using Sample Average Approximation

Sample Average Approximation (SAA) [5] is a method to solve an optimization problem of the form

$$\min_{x \in \Theta} \tilde{f}(x), \tag{7}$$

where $\Theta \subseteq \mathbb{R}^d$ $(d < \infty)$ and the real-valued function $\tilde{f}(\cdot)$ cannot be computed exactly, but can be estimated through a (stochastic) simulation.

Throughout this section, $(P, \mathcal{C})$ denotes a p-scheduling problem, where $P = \langle A, E, T, C, L, D\rangle$ is a scheduling problem and $\mathcal{C} = \langle N, \alpha, \beta, \lambda\rangle$ is a constraint over P.

Observe that in SHSPs, if the user preferences are deterministic (i.e., each cell of the matrix N is a real number), then a schedule that maximally satisfies the user preferences can be easily computed (e.g., as proposed in [18]). So, one way to use SAA in computing a solution of a SHSP $\mathcal{P} = (P, \mathcal{C})$ is to randomly generate deterministic samples from the given $\mathcal{P}$, compute their solutions, and take the average of these solutions as the solution of $\mathcal{P}$. From this realization, in order to solve a SHSP using SAA approach, we aim at formalizing the SHSP as a *Mixed Integer Linear Programming* (MILP). We start with some extra notations.

Definition 7. *Let N be a preference matrix of the size $|A| \times |T|$. A **sample** of N is an $|A| \times |T|$ matrix where the cell (i, j) is a value generated from the Normal distribution $\mathcal{N}(\mu_{ij}, \sigma_{ij})$.*

Given a sample and a schedule, the projection of the schedule on the sample is defined as follows.

Definition 8. *Let s be a sample matrix of N. The **projection** of a schedule H on s, denoted by $\mathfrak{H}$, is an $|A| \times |T|$ matrix, where, for each cell (i, j)*

$$\mathfrak{H}(i,j) = \begin{cases} s(i,j) \text{ if } H(i,j) = 1 \\ 0 \text{ otherwise} \end{cases} \tag{8}$$

Definition 9. *Let s be a sample matrix of N. The **cumulative preference** of a schedule H in s, denoted by C_s, is defined by*

$$C_s = \sum_{i=1}^{|A|} \sum_{j=1}^{|T|} \mathfrak{H}(i,j). \tag{9}$$

We define the indicator function for a sample s as follows.

$$f(s) = \begin{cases} 1 \text{ if } C_s \geq \alpha \\ 0 \text{ otherwise} \end{cases} \tag{10}$$

Let n be an integer and $\mathcal{S} = \{s_k\}_{k=1}^n$ be a sequence of samples of N. A schedule H is said to satisfy the user preference condition in $\mathcal{S}$ if

$$\sum_{k=1}^{n} f(s_k) \geq \frac{\beta \times n}{100} \tag{11}$$

The above allows us to transform a p-scheduling problem $\mathcal{P} = (P, \mathcal{C})$ to a MILP,[1] denoted as program Π:

$$\mathcal{S} = \{s_k\}_{k=1}^{n} \text{ is a sequence of } n \text{ samples of } N$$

$$\text{maximize} \sum_{k=1}^{n} f(s_k) \text{ subject to}$$

$$\sum_{j=1}^{|T|} H(i,j) = 1 \qquad \forall i = 1, ..., |A| \qquad \text{(from Definition 4)}$$

$$e(j) \leq l_j \qquad \forall j = 1, ..., |T| \qquad \text{(from Eq. 3)}$$

$$\sum_{j=1}^{|T|} c(j) \leq \lambda \qquad \text{(from Eq. 5)}$$

$$\sum_{i=1}^{|A|} \sum_{j=1}^{|T|} H(i,j) \cdot s_k(i,j) \geq \alpha \qquad \forall k = 1, ..., n$$

$$\sum_{k=1}^{n} f(s_k) \geq \frac{\beta \times n}{100} \qquad \text{(from Eq. 11)}$$

4 Solving SHSP Using Depth-First Search

In this section, we propose an approach to solving p-scheduling problems using *depth-first search* (DFS).[2] As in the previous section, $\mathcal{P} = (P, \mathcal{C})$ denotes a p-scheduling problem, where $P = \langle A, E, T, C, L, D \rangle$ is a scheduling problem and $\mathcal{C} = \langle N, \alpha, \beta, \lambda \rangle$ is a preference over P. Due to Observation 1, we can view $\mathcal{P}$ as a *Constraint Satisfaction Problem* (CSP), denoted by $csp(\mathcal{P})$, whose set of variables $\widetilde{H}_1, \ldots, \widetilde{H}_{|A|}$ and the domain of each variable is $\{1, \ldots, |T|\}$. Intuitively, each variable $\widetilde{H}_i$ encodes a schedule of the appliance i. It is easy to see that there is a one-to-one correspondence between a complete variable assignment $\widetilde{H} = \{\widetilde{H}_i = v_i \mid i = 1, \ldots, |A|\}$ of $csp(\mathcal{P})$ and a schedule H of $\mathcal{P}$ defined by

$$H(i,j) = 1 \quad \text{iff} \quad \widetilde{H}_i = j. \tag{12}$$

For this reason, we often use H and $\widetilde{H}$ interchangeably. We will begin with a theorem that helps in choosing the value of a variable in the expansion phase as well as pruning the search tree.

[1] Our proposed transformation from a p-scheduling problem $\mathcal{P} = (P, \mathcal{C})$ to a MILP does satisfy Conditions (3), (4), and (5), but ignores the dependency matrix D in P. We leave a proposal for a complete transformation for future work.

[2] One could also use other search algorithms as well because our core contribution here is to formulate the problem into a search problem and propose a number of pruning conditions that can be used with any search strategy.

Theorem 1. *Let* $\mathcal{N}(\mu_1, \sigma_1), \mathcal{N}(\mu_2, \sigma_2), ..., \mathcal{N}(\mu_n, \sigma_n)$ *be the Normal distributions of the random variables* $x_1, x_2, ..., x_n$*, respectively,* α *be a number, and* $K \in [1, n]$*. Then, the probability* $x_K \geq \alpha$ *is maximal when* $\frac{\mu_K - \alpha}{\sigma_K}$ *is maximal among* $\frac{\mu_1 - \alpha}{\sigma_1}, \ldots, \frac{\mu_n - \alpha}{\sigma_n}$.

Proof. Denote $\phi(\xi) = Prob(x < \xi)$ for a standard Normal distribution. It holds that

$$\begin{aligned} Prob(\mathcal{N}(\mu_K, \sigma_K) \geq \alpha) &= Prob(\sigma_K \mathcal{N}(0,1) + \mu_K \geq \alpha) \\ &= Prob\Big(\mathcal{N}(0,1) \geq \frac{\alpha - \mu_K}{\sigma_K}\Big) = 1 - \phi\Big(\frac{\alpha - \mu_K}{\sigma_K}\Big) \end{aligned} \tag{13}$$

This implies that:

$$\begin{aligned} \max_K(Prob(\mathcal{N}(\mu_K, \sigma_K) \geq \alpha)) &\Leftrightarrow \max_K \Big(1 - \phi\Big(\frac{\alpha - \mu_K}{\sigma_K}\Big)\Big) \\ &\Leftrightarrow \min_K \Big(\phi\Big(\frac{\alpha - \mu_K}{\sigma_K}\Big)\Big) \\ &\Leftrightarrow \min_K \Big(\frac{\alpha - \mu_K}{\sigma_K}\Big) \\ &\Leftrightarrow \max_K \Big(\frac{\mu_K - \alpha}{\sigma_K}\Big) \end{aligned} \tag{14}$$

How can Theorem 1 help us in searching for a schedule? Observed that if we were to use DFS to solve the CSP of $\mathcal{P}$, then each possible solution H corresponds to a complete assignment of the variables $\widetilde{H}_i$ $(1 \leq i \leq |A|)$. So, Theorem 1 indicates that the schedule H with $\mathcal{N}(\mu_H, \sigma_H)$ that has large μ_H and small σ_H has the best chance to satisfy the user preferences than other schedules. More precisely, from Eqs. 4 and 13, we have the following:

$$\begin{aligned} F_{\mathcal{N}(\mu_H, \sigma_H)}(\alpha) \geq \beta &\Leftrightarrow Prob(\mathcal{N}(\mu_H, \sigma_H) \geq \alpha) \geq \beta \\ &\Leftrightarrow 1 - \phi(\frac{\alpha - \mu_H}{\sigma_H}) \geq \beta \Leftrightarrow \phi(\frac{\alpha - \mu_H}{\sigma_H}) \leq 1 - \beta \end{aligned} \tag{15}$$

Consider a partial assignment $\widetilde{H}'$ and assume that $\widetilde{H}$ is a completion of $\widetilde{H}'$, i.e., $\widetilde{H}' \subseteq \widetilde{H}$. Let

$$\mu^{max}_{\widetilde{H}'} = \sum_{\widetilde{H}_i = v_i \in \widetilde{H}'} \mu(i, v_i) + \sum_{\widetilde{H}_i = v_i \in \widetilde{H} \setminus \widetilde{H}', i \neq k} \max_{j=1,\ldots,|T|} \mu(i, j) \tag{16}$$

$$\sigma^{max}_{\widetilde{H}'} = \sum_{\widetilde{H}_i = v_i \in \widetilde{H}'} \sigma(i, v_i) + \sum_{\widetilde{H}_i = v_i \in \widetilde{H} \setminus \widetilde{H}'} \max_{j=1,\ldots,|T|} \sigma(i, j) \tag{17}$$

Assume that H is the schedule equivalent to $\widetilde{H}$ as defined in (12). Clearly,

$$\mu^{max}_{\widetilde{H}'} \geq \mu_H \text{ and } \sigma^{max}_{\widetilde{H}'} \geq \sigma_H \tag{18}$$

Therefore,

$$\frac{\alpha - \mu^{max}_{\widetilde{H}'}}{\sigma^{max}_{\widetilde{H}'}} \leq \frac{\alpha - \mu_H}{\sigma_H} \tag{19}$$

which implies

$$1 - \phi\Big(\frac{\alpha - \mu^{max}_{\widetilde{H}'}}{\sigma^{max}_{\widetilde{H}'}}\Big) \geq 1 - \phi\Big(\frac{\alpha - \mu_H}{\sigma_H}\Big) \tag{20}$$

That leads to the following theorem, which is also the pruning condition based on user preference.

Theorem 2. *Assuming that $\widetilde{H}'$ is a partial assignment of the variables in the CSP of a p-scheduling problem $\mathcal{P}$. If*

$$1 - \phi\Big(\frac{\alpha - \mu^{max}_{\widetilde{H}'}}{\sigma^{max}_{\widetilde{H}'}}\Big) < \beta$$

and $\widetilde{H}$ is a completion assignment such that $\widetilde{H}' \subseteq \widetilde{H}$, then the schedule H corresponding to $\widetilde{H}$ does not satisfy Condition (4).

Theorem 2 can then be used to eliminate a partial assignment $\widetilde{H}'$ from consideration in the search for a solution. This is used in the algorithm that we present next.

Given a problem $P = \langle A, E, T, C, L, D\rangle$, we say that the number of dependencies of an appliance i, denoted by $d(i)$, is the number of elements in row i of D whose value differs from **nil**. Without loss of generality, we will assume that the appliances in A are listed in decreasing order of dependencies, i.e., if $1 \leq i < j \leq |A|$, then $d(i) \geq d(j)$. For a partial assignment $\widetilde{H}'$, let

$$cost(\widetilde{H}') = \sum_{\widetilde{H}_i = v_i \in \widetilde{H}'} c_{v_i} \cdot \left(\sum_{i=1}^{|A|} e_i \times |\{i \mid \widetilde{H}_t = v_i \in \widetilde{H}'\}| \right)$$

Notice that $cost(\widetilde{H}')$ is the energy consumption of all appliances specified by $\widetilde{H}'$. Due to the space limitation, we will only present the algorithm for computing an optimal solution for a p-scheduling problem $\mathcal{P} = (P, \mathcal{C})$.

In Algorithm 1, the function $ok(\widetilde{H})$ returns **false** if one of the following conditions is satisfied: (*i*) $cost(\widetilde{H}) > \lambda$ (cost efficiency requirement violated); (*ii*) the dependencies among current scheduled appliances do not satisfy the conditions in Definition 4 (dependency violated); and (*iii*) $1 - \phi\Big(\frac{\alpha - \mu^{max}_{\widetilde{H}}}{\sigma^{max}_{\widetilde{H}}}\Big) < \beta$ (user preference violated).

Intuitively, Algorithm 1 implements DFS by selecting a time slot for an appliance in each iteration of the overall **while-loop** (Lines 6–36), in the order $1, \ldots, |A|$. If all the time slots of the first appliance have been considered, then the search is complete (Line 8–10). For each device, the algorithm starts with the time slot whose preference distribution has maximal mean over other time

slots that have not been considered (Line 18–19). When a time slot is assigned to an appliance, the algorithm uses Theorem 2 and other checks ($ok(.)$) to rule out whether the search should be continued or backtracked (the **loop** command, Lines 21–23) for a different time slot of the appliance or the previous appliance (Lines 21–23). When a backtrack to the previously considered appliance (Lines 11–17), the assignment of the current appliance is removed from the schedule and its set of time slots is reseted to **false** (Lines 12–15). If all appliances have been assigned some time slots, then we need to check whether the generated schedule satisfies the user preference and optimal (Lines 26–30).

Observe that if we add "**return** $\widetilde{H}$" to Line 29 of Algorithm 1, then it returns the first satisfiable schedule. Furthermore, additional bookkeeping on the dependencies (e.g., removing all time slots that violate the dependencies in the schedule from the set $c(i)$) could help prune certain selections. We did implement this measure in our implementation. Due to the verifications in Lines 21 and 26, it is easy to see that the following theorem holds.

Theorem 3. *Algorithm 1 is sound and complete.*

5 Experiments

We performed an empirical evaluation of the two proposed methods (labeled DFS and SAA) on randomly-generated problems (i.e., problems with randomly-generated energy consumption vectors E and preference matrices N).

We implemented DFS method using Python, and we used MATLAB Release 2017a, for SAA method, to solve the mixed integer linear programming proposed in program Π. The number of samples in SAA is 100. In our experiments, we investigate the runtime and success rates of the two approaches in the following four SHSP variants:

- *Variant 1*: There are 10 dependencies between the appliances and the goal is to find a satisfiable solution.
- *Variant 2*: All appliances are independent from each other and the goal is to find a satisfiable solution.
- *Variant 3*: There are 10 dependencies between the appliances and the goal is to find an optimal solution.
- *Variant 4*: All appliances are independent from each other and the goal is to find an optimal solution.

For each variant, we generated problems varying the number of appliances $|A| = 20, 25, 30, \ldots, 65$, set the horizon $|T| = 24$, and set $\alpha = 6.5 \cdot |A|$ and $\beta = 0.8$. Finally, we use costs from the literature [12].

We set a time limit of 10 min and 1 hour for problems whose goal is to find satisfiable and optimal solutions, respectively, and we report average runtimes and success rates (=number of instances successfully solved) of the two approaches.

Input : A p-scheduling problem $\mathcal{P} = (P, \mathcal{C})$
Output: An optimal schedule of $\mathcal{P}$

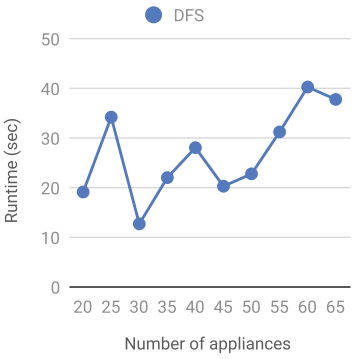
optimalValue $= +\infty$
optimalCandidate = **nil**
Let $\widetilde{H} = \emptyset$
Let *checked* be a Boolean $|A| \times |T|$ matrix, initialized with **false**
i $= 1$
while true do
  Let $c(i) = \{k \mid checked(i,k) = \textbf{false}\}$
  if $c(i) = \emptyset \wedge i = 1$ **then**
    break
  end
  if $c(i) = \emptyset \wedge i > 1$ **then**
    Set $checked(i,k) = \textbf{false}$ for $k = 1, \ldots, |T|$
    Identify x such that $\widetilde{H}_i = x$ belongs to $\widetilde{H}$
    $\widetilde{H} = \widetilde{H} \setminus \{\widetilde{H}_i = x\}$
    $i = i - 1$
    loop
  end
  Let $j \in c(i)$ such that $\mu(i,j) = \max_{k \in c(i)} \mu(i,k)$
  $checked(i,j) = \textbf{true}$
  $\widetilde{H} = \widetilde{H} \cup \{\widetilde{H}_i = j\}$
  if $\neg ok(\widetilde{H})$ **then**
    loop
  end
  if $i = |A|$ **then**
    Let H be the schedule correspond to $\widetilde{H}$
    if $F_{\mathcal{N}_H(\mu_H, \sigma_H)}(\alpha) \geq \beta \wedge f_c^H <$ optimalValue **then**
      optimalCandidate $= \widetilde{H}$
      optimalValue $= f_c^H$
      % **return** $\widetilde{H}$ if only satisfiable schedule is needed
    end
    Identify x such that $\widetilde{H}_i = x$ belongs to $\widetilde{H}$
    $\widetilde{H} = \widetilde{H} \setminus \{\widetilde{H}_i = x\}$
  else
    i $=$ i $+1$
  end
end
if optimalCandidate $=$ **nil then**
  return no optimal schedule found
end
return optimalCandidate;

Algorithm 1: Computing an Optimal Schedule for $\mathcal{P} = (P, \mathcal{C})$

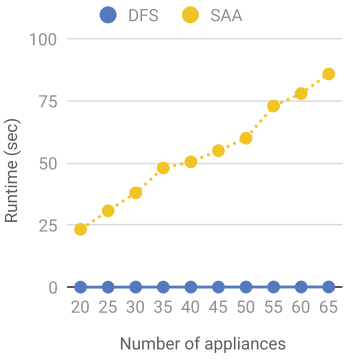
Figure 1 shows the runtimes (in seconds) and Table 2 tabulates the success rates (in percentages) of the two approaches on the four variants. We only ran

SAA for Variant 2 because our formulation does not take into account dependencies between appliances (and is thus inapplicable for Variants 1 and 3) and it is not guaranteed to find optimal solutions since it is an approximation approach (and is thus inapplicable for Variants 3 and 4). Results for the number of appliances $|A| > 35$ for Variants 3 and 4 are not shown because none of the approaches successfully solved a single instance for those large problems within the time limit.

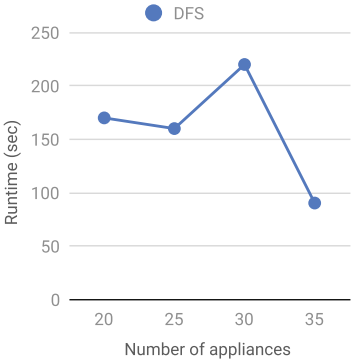
The results show that DFS is faster than SAA in Variant 2 and, thus, it is more scalable than SAA. In Variants 1 and 2, DFS maintains acceptable runtimes of within 40 seconds and success rates of approximately 80%. Not surprisingly, DFS is slower when solving the optimization problems of Variants 3 and 4 compared to the satisfaction problems of Variants 1 and 2. Similarly, DFS also has smaller success rates on the optimization problems.



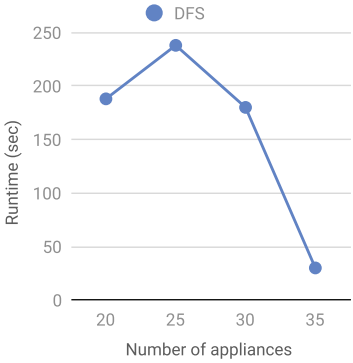
(a) Variant 1



(b) Variant 2



(c) Variant 3



(d) Variant 4

Fig. 1. Average runtimes of solved instances (in seconds)

6 Related Work

In the literature, existing works to solve SHSPs can be divided into three categories: *mathematical optimization*, *meta-heuristic search* and *heuristic search* [2]. Between these three categories, most of the work is in mathematical optimization, especially ones using MILP-based approaches. Some recent works [1,3,10,12] used MILP, while others have formulated the problem in convex programming [17] and quadratic programming [15] in which the cost or the energy consumption function is optimized. To a certain extent, our proposed SAA model can be considered as the first work that applies SAA mathematical model into solving SHSPs.

Table 2. Success rates (in percentages)

Variant	Approach	Number of appliances $\|A\|$									
		20	25	30	35	40	45	50	55	60	65
1	DFS	75	79	84	80	83	72	85	79	81	81
2	DFS	100	100	100	100	100	100	100	100	100	100
	SAA	100	100	100	100	100	95	86	81	69	46
3	DFS	42	25	14	5	0	0	0	0	0	0
4	DFS	35	23	15	7	0	0	0	0	0	0

There exist also several studies in applying search techniques in managing residential energy usage [4,7,9]. For example, Misra *et al.* presented a scheduling mechanism based on *Markov Decision Processes* (MDPs) for reducing energy expenses, which differs from our approach as we do not use MDPs [9]. Our work is similar to the work by Lee *et al.* [7], where they introduced a backtracking-based scheduling approach. The key differences between their work and ours are the following: (*i*) They do not take into account user preferences; (*ii*) Their objective is to reduce the peak energy load of homes or buildings, while ours is to minimize the cost of energy usage while satisfying user preference threshold.

Another work that is worth mentioning is the one by Fioretto *et al.* [4], where the authors described a mapping of SHSPs to distributed constraint optimization problems and proposed a distributed algorithm to solve it. The key difference between their work and ours is that they solve the bi-objective optimization problem by minimizing the weighted sum of both energy cost and user discomfort (opposite of user preference). Instead, we seek to only minimize energy cost while ensuring that the user preference (or user discomfort) is within some acceptable threshold.

With respect to the last general area of meta-heuristic search, researchers have proposed a scheduling method using genetic algorithm [8] and studied the application of particle swarm optimization [11] in determining a near-optimal solution for a multi-objective optimization problem like SHSPs.

7 Conclusions and Future Work

In this paper, we introduced two approaches to solve the *smart home scheduling problem* with *probabilistic user preferences*. More precisely, we consider the scheduling problem when user preferences for using (turning on/off) a device are Normal distributions. The first approach relies on *sample average approximation* (SAA) and the second approach uses *depth-first search* (DFS). We also propose pruning strategies, which we applied to our DFS algorithm. As these strategies are general for search-based approaches, they can also be applied to other heuristic search approaches aside from DFS. Our experimental results show that DFS is faster and scales better than SAA.

Future work includes more comprehensive evaluations, where we vary the α and β parameters of DFS and SAA as well as the degree of dependencies of the devices. We also plan to investigate improved optimization techniques for SAA. Finally, we also plan to consider an online extension of the problem, where some subset of preferences are elicited and schedules are provided in a repeated and interactive manner with users.

Acknowledgment. This research is partially supported by NSF grants 1242122, 1345232, 1619273, 1623190, 1757207, 1812618, and 1812619. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the sponsoring organizations, agencies, or the U.S. government. We would also like to thank Long Tran-Thanh for initial discussions that influenced the direction of this research.

References

1. Agnetis, A., De Pascale, G., Detti, P., Vicino, A.: Load scheduling for household energy consumption optimization. IEEE Trans. Smart Grid **4**(4), 2364–2373 (2013)
2. Beaudin, M., Zareipour, H.: Home energy management systems: a review of modelling and complexity. Renew. Sustain. Energy Rev. **45**, 318–335 (2015)
3. Costanzo, G.T., Zhu, G., Anjos, M.F., Savard, G.: A system architecture for autonomous demand side load management in smart buildings. IEEE Trans. Smart Grid **3**(4), 2157–2165 (2012)
4. Fioretto, F., Yeoh, W., Pontelli, E.: A multiagent system approach to scheduling devices in smart homes. In: Proceedings of the International Conference on Autonomous Agents and Multi-Agent Systems, pp. 981–989 (2017)
5. Kleywegt, A.J., Shapiro, A., Homem-de Mello, T.: The sample average approximation method for stochastic discrete optimization. SIAM J. Optim. **12**(2), 479–502 (2002)
6. Le, T., Tabakhi, A.M., Tran-Thanh, L., Yeoh, W., Son, T.C.: Preference elicitation with interdependency and user bother cost. In: Proceedings of the International Conference on Autonomous Agents and Multi-Agent Systems, pp. 1459–1875 (2018)
7. Lee, J., Kim, H.J., Park, G.L., Kang, M.: Energy consumption scheduler for demand response systems in the smart grid. J. Inf. Sci. Eng. **28**(5), 955–969 (2012)

8. Lujano-Rojas, J.M., Monteiro, C., Dufo-López, R., Bernal-Agustín, J.L.: Optimum residential load management strategy for real time pricing (RTP) demand response programs. Energy Policy **45**, 671–679 (2012)
9. Misra, S., Mondal, A., Banik, S., Khatua, M., Bera, S., Obaidat, M.S.: Residential energy management in smart grid: a Markov decision process-based approach. In: Proceedings of the IEEE International Conference on Green Computing and Communications; IEEE Internet of Things; and IEEE Cyber, Physical and Social Computing, pp. 1152–1157 (2013)
10. Molderink, A., Bakker, V., Bosman, M.G., Hurink, J.L., Smit, G.J.: Domestic energy management methodology for optimizing efficiency in smart grids. In: Proceedings of the IEEE Bucharest PowerTech, pp. 1–7 (2009)
11. Pedrasa, M.A.A., Spooner, T.D., MacGill, I.F.: Scheduling of demand side resources using binary particle swarm optimization. IEEE Trans. Power Syst. **24**(3), 1173–1181 (2009)
12. Sou, K.C., Weimer, J., Sandberg, H., Karl Henrik, J.: Scheduling smart home appliances using mixed integer linear programming. In: Proceedings of the IEEE Conference on Decision and Control and European Control Conference, pp. 5144–5149 (2011)
13. Tabakhi, A.M., Le, T., Fioretto, F., Yeoh, W.: Preference elicitation for DCOPs. In: Proceedings of the International Conference on Principles and Practice of Constraint Programming, pp. 278–296 (2017)
14. Tabakhi, A.M., Yeoh, W., Yokoo, M.: Parameterized heuristics for incomplete weighted CSPs with elicitation costs. In: Proceedings of the International Conference on Autonomous Agents and Multi-Agent Systems, pp. 476–484 (2019)
15. Tarasak, P.: Optimal real-time pricing under load uncertainty based on utility maximization for smart grid. In: Proceedings of the IEEE International Conference on Smart Grid Communications, pp. 321–326 (2011)
16. Truong, N.C., Baarslag, T., Ramchurn, G., Tran-Thanh, L.: Interactive scheduling of appliance usage in the home. In: Proceedings of the International Joint Conference on Artificial Intelligence, pp. 869–1467 (2016)
17. Tsui, K.M., Chan, S.C.: Demand response optimization for smart home scheduling under real-time pricing. IEEE Trans. Smart Grid **3**(4), 1812–1821 (2012)
18. Zhu, Z., Tang, J., Lambotharan, S., Chin, W.H., Fan, Z.: An integer linear programming based optimization for home demand-side management in smart grid. In: Proceedings of the IEEE PES Innovative Smart Grid Technologies, pp. 1–5 (2012)