Uncertain FlexOffers

a scalable, uncertainty-aware model for energy flexibility

Lilliu, Fabio; Pedersen, Torben Bach; Siksnys, Laurynas; Neupane, Bijay

Published in:
e-Energy 2023 - Proceedings of the 2023 14th ACM International Conference on Future Energy Systems

DOI (link to publication from Publisher):
10.1145/3575813.3576873

Publication date:
2023

Document Version
Publisher's PDF, also known as Version of record

Uncertain FlexOffers: a scalable, uncertainty-aware model for energy flexibility

Fabio Lilliu
Torben Bach Pedersen
Aalborg University
Aalborg, Denmark

Laurynas Šikšnys
Bijay Neupane
Aalborg University
Aalborg, Denmark

ABSTRACT

As the usage of Renewable Energy Sources (RES) in electricity grids increases in popularity, energy flexibility has a crucial role. The most common weaknesses of current flexibility models are: i) being hard-coded for specific devices, ii) not scaling for long time horizons and many devices, iii) losing a lot of flexibility if the model is approximated, and iv) not considering the uncertainty affecting flexibility representations, which causes the model to capture too much excess flexibility when imbalance penalties are high. The FlexOffer (FO) model can perform approximations of flexibility with good accuracy across different devices, and scales well to long time horizons and many devices: this work extends FOs to **uncertain FOs** (UFOs), which keep the good properties while capturing uncertainty. We show that UFOs are very fast by performing optimization in under 5.27 seconds for a 24 hours time horizon, while exact models use more than 29.05 hours for even a 6 hours 15 minutes time horizon, making them totally infeasible in practice. UFOs can capture more flexibility than other uncertain models: UFOs considering energy dependencies can model flexibility without losses for a charging battery, and retain 86.8% of the total flexibility for batteries and 87.5% for EVs when imbalance penalties are high, compared to 79.6% and 74.4% respectively for other models. UFOs allow to aggregate up to 6000 loads for up to 96 time units while retaining 90.5% of the total flexibility: exact models fail already for 330 loads or 21 time units.

CCS CONCEPTS

• **Hardware → Energy generation and storage**.

KEYWORDS

energy flexibility, uncertainty, battery, EV

ACM Reference Format:
Fabio Lilliu, Torben Bach Pedersen, Laurynas Šikšnys, and Bijay Neupane. 2023. Uncertain FlexOffers: a scalable, uncertainty-aware model for energy flexibility . In *The 14th ACM International Conference on Future Energy Systems (e-Energy '23), June 20–23, 2023, Orlando, FL, USA.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3575813.3576873

e-Energy '23, June 20–23, 2023, Orlando, FL, USA

ACM ISBN 979-8-4007-0032-3/23/06.
https://doi.org/10.1145/3575813.3576873

1 INTRODUCTION

In the last decades, the use of Renewable Energy Sources is becoming more prominent in electrical grids. The capability of adjusting energy demand to RES production is therefore very valuable; to this purpose, some grid users (*prosumers*) can change their energy consumption in time and amount. This ability is called *flexibility*. Description of flexibility has been treated extensively in literature [18]. Flexibility can be used for optimization towards several objectives, such as minimizing energy costs [24, 26] or CO2 emissions [11], maximizing renewable energy consumption [31], peak shaving [4, 9], matching demand for power with supply (demand response) [5], ancillary services [34] and avoiding local grid congestions [8]. In some cases, those objectives are related, e.g. energy prices may vary to encourage demand response [37]. Flexibility also enables prosumers to participate to spot (day-ahead, intraday) and balancing energy markets, either on their own or joining their flexibility with other prosumers. Thus, various mathematical models have been created in order to describe flexibility, with different properties depending on the considered cases [22]. We want to create a model that can *capture* flexibility from many different types of devices [13], *optimize* the flexibility for generic purposes (in this paper we will consider the specific use case of cost reduction), *aggregate* the flexibility from many small energy loads into a few bigger ones [7], and do the opposite process (*disaggregation*) [27] in order to control the actual loads. To achieve this, the flexibility model needs to have the following properties: i) it has to model flexibility from different device types in an unified format; ii) it has to be scalable with respect to optimization for long time horizons, and aggregation of many loads; iii) it has to capture most/all of the total available flexibility, and iv) it has to consider that flexibility over long time horizons is subject to uncertainty, e.g. the prosumer might be unable to deliver it as promised. Considering uncertainty is especially important when approximations are penalized for overestimating flexibility, e.g. when imbalance penalties are high. Therefore, the scenario of high imbalance penalties is the primary use case of this work. For representing flexibility in a unified format, the models from [32] and [6] provide some examples. Regarding accuracy, linear time invariant (LTI) state-space models [3, 15] are very precise in representing batteries and heat pumps. The model from [14] accurately represents flexibility for building heating systems and water towers, and can model uncertainty; however, being a state-space model, it does not address ii). Existing uncertainty representations consider external variables like weather [19] and user behavior [12], but not the intrinsic uncertainty of the approximation itself. About scalability, [16] proposes a scalable and approximate geometrical model for flexibility aggregation. In particular, FlexOffer (FO) [29] is a model which generates

Parameter	Description
$SoC(t)$	State of charge at time t.
SoC_{min}	Minimum state of charge.
SoC_{max}	Maximum state of charge.
E_{min}	Maximum energy taken from the device.
E_{max}	Maximum energy given to the device.
K	Fraction of energy that is kept at each operation.
P_0	Probability threshold for UFOs
P_t	Time uncertainty
f_t	Amount uncertainty
PF	Profit function.
Spr	Spot market prices.
Ipr	Imbalance market prices.

Table 1: Table of symbols used through this paper.

good approximations of flexibility for many different types of loads, which can be aggregated and optimized in a scalable way [33], thus effectively addressing properties i) - iii). However, property iv) has not been considered yet: our contribution is to build a model that extends FOs and considers uncertainty, while still being device-independent, scalable and retaining most of the available flexibility, so that it addresses all of the properties i) - iv). This work extends the work done in [21], by describing in detail the definition of UFOs, the algorithms for optimization and aggregation, and the experiments done for validating them.

We call our proposal *uncertain FlexOffers* (UFOs). Throughout the paper, when describing examples, we will use short time horizons and 1 hour time units for simplicity; however, as we will explain in Section 4, when discussing real-life feasibility, we will refer to time horizons of 96 time units, 15 minutes each (i.e., 24 hours), as upcoming energy market regulations dictate. UFOs scale better than exact models: optimization for 96 time units is performed in 5.27 seconds, while exact models need more than 29.05 hours for 25 or more time units (6 hours 15 minutes), making them infeasible in practice. UFOs are capable to model flexibility without losses for a charging battery, and capture 86.8% of the total flexibility when imbalance penalties are high, compared to the 79.6% for other uncertain models. Regarding EVs, UFOs can retain up to 87.5% of the total flexibility when imbalance penalties are high, while other uncertain models can retain up to 74.4% of the total flexibility. UFOs also allow to aggregate up to 6000 loads or up to 96 time units while retaining more than 90.5% of the total flexibility for batteries, and up to 86.8% for EVs; exact models fail already for more than 330 loads or 21 time units, and the uncertain baseline only retains 82% of flexibility for batteries, and 84.5% for EVs. The remainder of the paper is organized as follows: Section 2 describes the FO model, Section 3 outlines our proposed model, Section 4 describes our experiments and Section 5 concludes the paper.

2 PRELIMINARIES

2.1 Running example

In this subsection we describe a specific energy load, which will be used through the paper as a running example. We will consider a *Tesla Powerwall* battery with capacity 14 kWh, maximum charging/discharging power of 5 kW, and round-trip efficiency of 90%. We use one hour time units, i.e., the battery can either be charged or discharged up to 5 kWh at each time unit. For describing the functioning of the battery, we use Coulomb counting [25]. At each time unit t, we write the state of charge (SoC) of the battery as

$$\begin{aligned} SoC(t) &= SoC(t-1) + K \cdot e_t^+ + K^{-1} \cdot e_t^- \\ SoC_{min} &\leq SoC(t) \leq SoC_{max}; E_{min} \leq e_t \leq E_{max}. \end{aligned} \tag{1}$$

Here, $SoC(t)$ is the amount of energy in the battery at time unit t, in kWh. e_t is the amount of energy that the prosumer gives to/receives from the battery at time t, in kWh: e_t is positive if the battery is being charged, negative otherwise. e_t^+ is $\max\{e_t, 0\}$, e_t^- is $\min\{e_t, 0\}$. $K \in [0,1]$ measures how much energy is *kept* while charging/discharging the battery. If losses are not equal for the charging or discharging processes, Eq. 1 can be generalized by replacing K and K^{-1} with $K_1 \in [0,1]$ and $K_2 \in [1,\infty)$ respectively; however, for simplicity, in this paper we will assume $K_2 = K_1^{-1}$. SoC_{min} and SoC_{max} are the minimum and maximum state of charge of the battery in kWh, respectively. E_{min} and E_{max} are the minimum and maximum amounts of energy (in kWh) that can be taken from/given to the battery in one time unit, from the perspective of the prosumer. In the running example: $SoC_{min} = 0$ kWh, $SoC_{max} = 14$ kWh, $E_{min} = -5$ kWh, $E_{max} = 5$ kWh, $K = \sqrt{0.9} \sim 0.948$. We will consider two use cases: in the first, $SoC(1) = 0$ kWh and the battery can only be charged (*charging only* case), in the second, $SoC(1) = 7$ kWh and the battery can be either charged or discharged at each time unit (*switching* case).

2.2 FlexOffer life-cycle

The baseline for this work is the FO model [29]. Suppose we want to model flexibility for a certain device: an FO can be seen as a set of constraints on the values of consumable energy for the upcoming time units, which describe the flexibility available from said device. Figure 1 shows the life-cycle of an FO. Two main parties are involved: the prosumer, who generates and executes the FO, and the aggregator [28], who processes and issues schedules for the FO. The tasks on the prosumer's side are performed automatically by an agent, which operates according to the prosumer's requirements. First, the prosumer agent forecasts flexibility for the devices, and generates FOs according to that. Each FO is then sent to the aggregator, which determines if the FO is useful for its needs, decides whether to accept the FO or not and informs the prosumer of the response. If the FO is rejected, it is not executed and the cycle ends here. Otherwise, the aggregator processes it (e.g., aggregating it with other FOs, performing optimization), and establishes a schedule for each FO. FO schedules are then sent back to the prosumer agent, which will execute them by controlling the devices.

2.3 Description of a FlexOffer

An FO is an approximation of the available flexibility, expressed in terms of constraints over the usable amount of energy at each time unit. In this section, call e_t the amount of energy consumed at time t, and T the time horizon we are considering. There are many types of constraints that have been used to define FOs: the most simple ones are *slice* (energy) *constraints*. An energy constraint establishes, at each time unit, the minimum and maximum amount

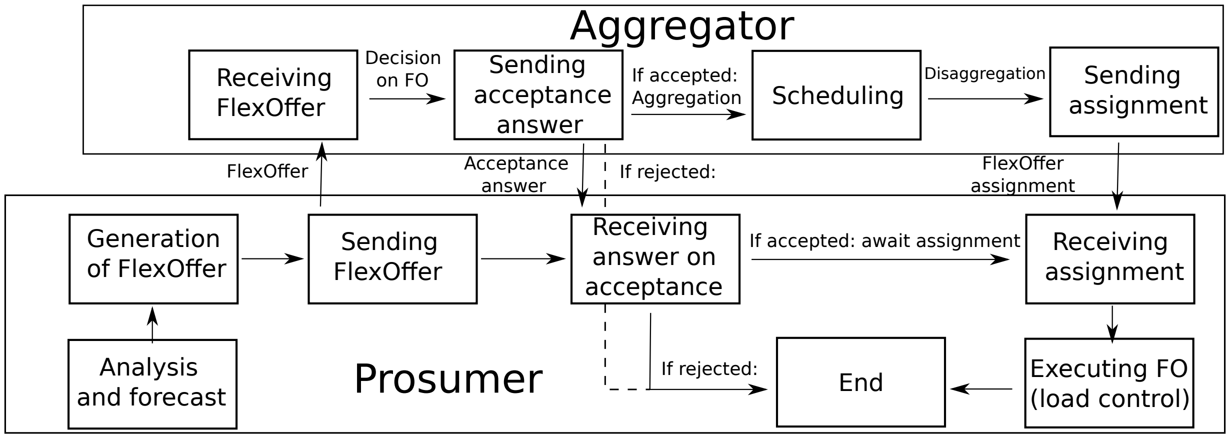

Figure 1: A schematic description of the FO life-cycle.

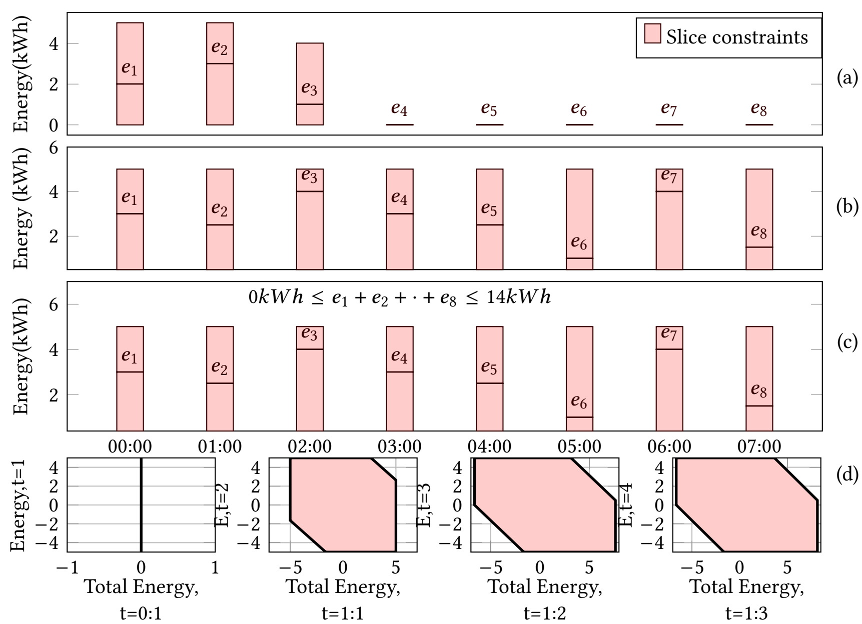

Figure 2: Inner (a), outer (b), TEC (c) SFOs, and DFOs (d)

of energy that the load can consume: at every time t, the energy constraint specifies a lower and an upper bound $emin_t$ and $emax_t$ such that $emin_t \leq e_t \leq emax_t$. A *standard FO* (SFO) is an FO whose constraints are all slice constraints. Another type of constraint is the *total energy constraint* (TEC), which specifies the lower (TE_{min}) and upper (TE_{max}) bounds for the energy that can be consumed over the considered time horizon. A *total energy constraint standard FO* (TEC-SFO) is an FO with slice and total energy constraints.A further type of constraint is the *dependent energy constraint.* This constraint specifies at each time unit t a lower and an upper bound on the amount of energy that can be consumed, depending on the total amount of energy that has been consumed before time unit t. Formally, this means that there are three real numbers a, b, c such that $a \cdot (e_1 + \ldots + e_{t-1}) + b \cdot e_t \leq c$. A *dependency FO* (DFO) is an FO with dependency energy constraints.

There are two main types of approximations: *inner* and *outer*. An inner approximation models less flexibility than the actually available amount, while an outer approximation models more. Outer FOs generate more flexibility compared to inner FOs, but some of the modeled configurations may actually be infeasible. In our use case, , high imbalance penalties, an outer approximation with excessive constraint violation is severely penalized. Figure 2 shows the constraints described in this section for the *charging only case* of the running example from Section 2.1: (a) represents an inner approximation SFO and (b) an outer approximation SFO. At each time unit, the column describes the amount of energy that can be used, and the horizontal line is an example schedule.A TEC-SFO is represented in (c), with the TEC shown above the slice constraints, and (d) shows a DFO: for each time unit t, the x axis indicates how much energy has been consumed in total before time t, and the y axis indicates how much energy can be consumed based on the value on the x axis. For the *switching* case the battery can be both charged and discharged, and as a consequence the FO slices can also represent negative values of energy, which correspond to discharging the battery. However, in Figure 2 FOs we show the *charging only* case for simplicity: in this case only positive values for energy are shown, since the battery can only be charged.

3 UNCERTAIN FLEXOFFERS

3.1 Types of uncertainty

The purpose of this work is to propose a new type of FO, which takes uncertainty into account. We consider two types of uncertainty over flexible loads: **time uncertainty**, and **amount uncertainty**. Time uncertainty is the uncertainty about if and when the flexible load will be available, while amount uncertainty is the uncertainty about how much energy will be consumed by the load per time slice and overall. Consider FOs as energy profiles: time uncertainty is the uncertainty about being able to deliver the energy described in the profile, and amount uncertainty is the uncertainty on the amount of energy of the profile. For example, suppose a prosumer wants to recharge an electric vehicle (EV) overnight, in order to use it the next day. At each time unit, time uncertainty refers to the probability of the user having the EV plugged in for recharge at that time, and amount uncertainty refers to the amount of energy that can be given to/taken from the EV at that time. Table 2 shows which types of uncertainty affect each device. **Wet devices** (i.e., washing machines, dishwashers) can only provide flexibility when the prosumer loads them, and sets the device to *ready* [10], therefore they have time uncertainty, depending on if and when the prosumer will set the devices to *ready*. However, since their energy profiles are pre-determined, there is no amount flexibility. **Heat pumps** do not have time uncertainty, as they are always plugged in (if inactive, their consumption is simply zero); they however have amount uncertainty, as their energy consumption is not pre-determined. **EVs** have both types of uncertainty, as shown above. Finally, **home batteries** are always plugged in, and therefore have no time uncertainty; they do however have amount uncertainty, as the amount of charge or discharge is not pre-determined.

Device	Time	Amount
Wet device	✓	
Heat pump		✓
EV	✓	✓
Home battery		✓

Table 2: Type of uncertainty for each device.

Throughout this paper we will talk about two main use cases: a battery, and an EV. We start with batteries because they only have amount uncertainty, which is the most general type to model. After this, we describe EVs because they are affected by all two types of uncertainty, and therefore are the most general case. This means that if we can model uncertainty for EVs, we can do the same for all other types of devices. We start by explaining how amount uncertainty works for batteries, and then we will show how, in addition to that, time uncertainty is modeled for EVs.

3.2 Definition of uncertain FlexOffers

To create an uncertainty-aware FO, we first need to model uncertainty. For simplicity, we start by considering amount uncertainty, and we consider the case of a battery. With a slight abuse of notation, in this section we will denote by e_t the amount of energy that the battery receives from/gives to the prosumer. Call T the time horizon we are considering. Since the flexibility available at time t depends on $SoC(t)$, we will have to model uncertainty for the value $SoC(t)$ at each $t \in \{1, \dots, T\}$. $SoC(1)$ can be measured, which means that for $t = 1$ we can determine the available flexibility with certainty. However, for $t > 1$, $SoC(t)$ depends on the amount of energy used before: therefore, the uncertainty for $SoC(t+1)$ depends on the uncertainty for e_t. Suppose that we know the probability distribution function (PDF) that describes the probability for e_t to assume each possible value in $[E_{min}, E_{max}]$, for each $t \in \{1, \dots, T\}$, and call this PDF $\overline{e}_t$: it is then possible to determine the PDF $\overline{SoC}_{t+1}$ that describes the probability for $SoC(t+1)$ to assume each possible value in $[SoC_{min}, SoC_{max}]$. We already know $SoC(1)$, so $\overline{SoC}(1) = \delta_{SoC(1)}$, the Dirac delta distribution concentrated at $SoC(1)$. The random variable with PDF $\overline{SoC}_{t+1}$ has to be the sum of the random variable with PDF $\overline{SoC}_t$ and the one with PDF $\overline{e}_t$, which can be written as:

$$\overline{SoC}_{t+1}(x) = \int_{E_{min}}^{E_{max}} \overline{SoC}_t(x-r) \cdot \overline{e}_t(r) dr. \tag{2}$$

We now know the probability distributions related to SoC, and we want to determine the uncertainty in value flexibility. At time t, call I_t the set of all values that e_t can possibly assume. We know that I_t is an interval and that $I_t \subseteq [E_{min}, E_{max}]$. Our purpose is to build a function $f_t : [E_{min}, E_{max}] \to [0, 1]$ that, for every $x \in [E_{min}, E_{max}]$, describes the probability that $x \in I_t$. f_t is not a PDF: $f_t(x)$ does not describe the probability that e_t will assume the value x when the schedule is determined. $f_t(x)$ instead describes the probability that x belongs to the set of the feasible values for e_t before the schedule is chosen. We define f_t as

$$f_t(x) = \int_{SoC_{min}-x}^{SoC_{max}-x} \overline{SoC}_t(r) dr \tag{3}$$

This is because $x \in I$ if and only if $SoC_{min} \leq SoC(t) + x \leq SoC_{max}$, or equivalently $SoC_{min} - x \leq SoC(t) \leq SoC_{max} - x$: the integral from Eq. 3 describes the probability for this condition to be respected. Since it is difficult from a computational perspective to create pointwise-defined functions as in Eq. 2 and 3, we created a discrete approximation approach. We assume that $SoC(t)$ and e_t can only have values which are multiples of gr (a number describing the granularity of the approximation), and therefore $\overline{e}_t$ and $\overline{SoC}_t$ are discrete distributions.

Figure 3 shows $\overline{SoC}_t$ for the running example, and Figure 4 shows f_t. It can be noticed that at time t, $\overline{SoC}_t$ behaves locally like a polynomial of degree $t-1$.

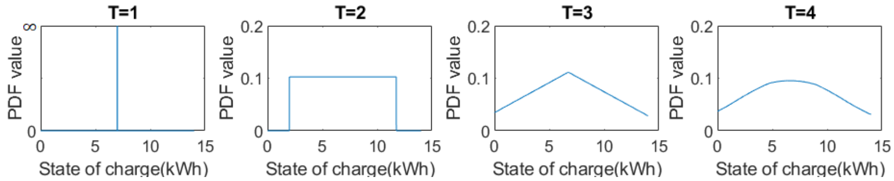

Figure 3: SoC probability distributions.

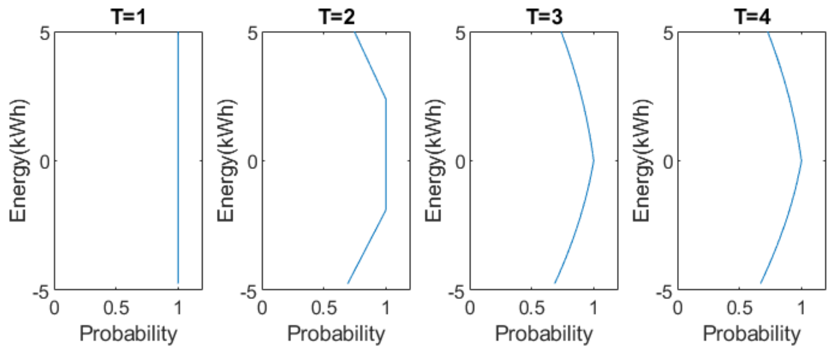

Figure 4: Flexibility probability functions.

We now want to describe time uncertainty. For this, we define an example that will be denoted through the paper as the *EV charging only example*. Consider the case of a prosumer who recharges an EV overnight. Suppose that the probability of the user plugging in the EV for a certain day is 95% if it is a weekday, and 20% if it is a weekend day. Suppose also that, when the recharging happens, it always happens overnight, at a time between 17 of the considered day and 8 of the following day. In our example, plug-in and plug-out times are modeled by lognormal distributions, tuned as in the graph from [1]. Finally, the SoC at the plug-in is modeled by a uniform distribution [17], with a mean value of 40% [35].

At each time t, we denote by P_t the probability that the EV will be plugged in at that time: in our example, during the weekdays we have $P_{17} = 0$, $P_{18} = 0.475$ and $P_{19} = 0.83$, and during the weekend it results $P_{17} = 0$, $P_{18} = 0.1$ and $P_{19} = 0.17$. Lastly, as shown in the battery case, at each time t we create the function f_t describing amount uncertainty for each possible energy value. In this case, amount uncertainty comes from the inability to know the state of charge of the EV at plug-in time when we issue the UFO, which is many hours before the EV is plugged in. If flexibility cannot be provided, we denote by d_t the default consumption for the device at time t. In the EV *charging only* example, when the prosumer plugs in at 19, we have $d_{18} = 0$ kWh.

With this notation, we can represent the combined uncertainties by some functions $g_1, \ldots, g_T$, where

$$g_k(e_k) = \begin{cases} f_k(e_k) & \text{if } e_k = d_k \\ P_t(F) \cdot f_k(e_k) & \text{otherwise.} \end{cases}$$

In general, we define an **uncertain FlexOffer** F as a tuple $\{g_1, g_2 \ldots, g_T\}$, where, $g_1 \ldots, g_T$ are functions from $\mathbf{R}$ to $[0, 1]$ such that $g_t(e_t)$ represents the probability for the device to be able to consume the amount of energy e_k at time t.

Additionally, in the EV case, a TEC is being considered, since the EV needs to be charged by at least a certain amount for a user-defined deadline. Denoting by SoC_{final} the desired final amount of charge for the EV, the constraint can be expressed as $K \ldots \sum_{t=1}^{T} e_t = SoC_{final} - SoC(0)$. The user-defined deadline is modeled by the time uncertainty: P_t describes the probability that the EV can be charged at time t: this includes both the EV being plugged in and user settings allowing the charging operation, and in particular being before the charging deadline.

3.3 Optimization

FOs described in Section 2 can be seen as a set of constraints on the energy consumed by the modeled device: in particular, it is possible to use those constraints for solving optimization problems, which are defined by an objective function, e.g., cost minimization, and a set of constraints, defined by the FO over the energy variables e_t. In this subsection, we describe how UFOs can be used for optimization.

A *schedule* $s = (s_1, \ldots, s_T)$ for energy consumption is a vector of size T, such that its t-th component s_t describes the energy consumption for the time unit t. Given a UFO F and a schedule s, we define $P(s)$ as the probability for the schedule to be supported by F. Our idea for optimizing energy consumption is to set a probability threshold P_0, and use $P(e_1, \ldots, e_T) \geq P_0$ as a constraint for the optimization. However, since this is a nonlinear constraint, we chose another approach: create an SFO F_{P_0} whose solution space respects this constraint. Algorithm 1 shows how to create F_{P_0}. First, we compare P_t to P_0 (Line 2). If $P_t < P_0$, the procedure ends and the schedule $(d_1, \ldots, d_T)$ is generated (Lines 3-5), since it is the only possible schedule s such that $P(s) \geq P_0$. Otherwise, we define $P_r = \frac{P_0}{P_t}$ as the residual uncertainty left (Line 6). Now, for every $t \in \{1, \ldots, T\}$, we divide the residual uncertainty in T parts: this

Algorithm 1: Generating an SFO from a UFO

Input: T - time horizon;
F - UFO (vector of length T);
P_0 - probability threshold
Output: F_{st} - A standard FO

1 **Function** *generateSFO(F, P_0)*:
2   **if** $F.P_t < P_0$ **then**
3     $F_{st}.emin \leftarrow (F.d_1, \dots, F.d_T)$
4     $F_{st}.emax \leftarrow (F.d_1, \dots, F.d_T)$
5   **return** F_{st}
6   $P_r \leftarrow \frac{P_0}{P_t}$
7   $pr_1, \dots, pr_T \leftarrow \sqrt[T]{P_r}$
8   **for** $t \leftarrow 1 : T$ **do**
9     $F_{st}.emin_t \leftarrow \arg\min_x f_t(x) \geq pr_t$
10     $F_{st}.emax_t \leftarrow \arg\max_x f_t(x) \geq pr_t$
11 **return** F_{st}

Algorithm 2: SFO from aggregating UFOs

Input: T - time horizon;
$\{F_1, \dots, F_N\}$ - UFOs(vectors of length T);
P_0 - probability threshold
Output: F_{st} - A standard FO

1 **Function** *generateSFO(F, P_0)*:
2   $P_1 \leftarrow \sqrt[N]{P_0}$
3   **for** $t \leftarrow 1 : T$ **do**
4     $F_{st}.emin_t, F_{st}.emax_t \leftarrow 0, 0$
5   **for** $k \leftarrow 1 : N$ **do**
6       $F_{st}.emin_t \leftarrow F_{st}.emin_t + \arg\min_x f_t^k(x) \geq P_1$
7       $F_{st}.emax_t \leftarrow F_{st}.emax_t + \arg\max_x f_t^k(x) \geq P_1$
8 **return** F_{st}

is done by generating T numbers $pr_1, \dots, pr_T \in (0, 1]$ such that their product is equal to P_r, and imposing for every t the condition $f_t(e_t) > pr_t$. There are many possible ways to generate the numbers pr_t: the simplest is to assign to each one of them the value $\sqrt[T]{P_r}$, thus distributing uncertainty equally over the time units, as we show in the algorithm. Another possibility is to distribute uncertainty only among some of the time units in case we are not interested about uncertainty at that time: for example, if we are not interested for $t = 1$, we impose $pr_1 = 1$. The slices at each time t represent the values which respect this condition (Lines 9-10). This condition ensures that $P(s) \geq P_0$ for every possible schedule s obtained by these constraints. The time complexity of this algorithm is O(T).

Once the SFO has been generated, we solve the optimization problem. We choose the function over the energy variables e_t that we want to optimize, and we find the minimum or the maximum of this function, depending on the objective. The constraints on the energy variables are given by the SFO that we just generated. An example of objective function is profit, which is defined as $PF(e) = -Spr \cdot e$. Here, $e = (e_1, \dots, e_T)$ is the energy consumed at each time $1, \dots, T$, and $Spr = (Spr_1, \dots, Ipr_T)$ are the spot prices at time $1, \dots, T$. An example for the constraints is the SFO shown in Figure 2(a): here we have $T = 8$, and the constraints are

$$
\begin{aligned}
&0\text{kWh} \leq e_t \leq 4\text{kWh} && \text{if } t \in \{1, 2\} \\
&0\text{kWh} \leq e_t \leq 3\text{kWh} && \text{if } t \in \{3, 4\} \\
&e_t = 0\text{kWh} && \text{if } t > 4.
\end{aligned}
$$

3.4 Aggregation

An important property of FOs is aggregation: given a number N of FOs, it is possible to generate $M \ll N$ FOs which together represent the combined flexibility of the original N FOs, with some losses. Aggregation allows single prosumers to participate in the energy market by having their flexibility aggregated with other prosumers', thus being able to meet the requirements for minimum bid size. The aggregation and the market bids are performed by the aggregator, as shown in Figure 1. Also, optimizing flexibility for thousands of prosumers would normally be a computationally hard problem: aggregation allows to manage the flexibility from many prosumer in a small number of FOs, making optimization much faster. This section shows how UFOs can be aggregated. Suppose we have N UFOs $F_1, \dots, F_N$. Without loss of generality, we make some assumptions: first, for all the UFOs, $d_t = 0$ for every $t \in \{1, \dots, T\}$. Second, for every function f_t^k, we have $f_t^k(0) = 1$, and those functions are concave and monotonous when restricted to $\mathbf{R}^-$ and $\mathbf{R}^+$. We then define the aggregated UFO F^a as follows: given $m_1, \dots, m_{N-1}, m_N$ as non-negative numbers such that $m_1 + \ldots + m_N = x$, $f_t^k(m_k) \geq f_t^{k+1}(m_{k+1})$ for every $k < N$ and m_N is minimum. We define then $g_t^a(x) = \prod_{k=1}^{N} f_t^k(m_k)$, and $F^a = \{g_1^a, \dots, g_T^a\}$ is the aggregated UFO. In order to exploit F^a for optimization, given a threshold P_0, we want to create an SFO $F_{P_0}^a$ like in the previous subsection: Algorithm 2 does this without actually computing F^a. We first define P_1 as the N−th root of P_0 (Line 2). After that, for every $t \in \{1, \dots, T\}$, we define the slice constraints $emin_t$ and $emax_t$ as the sum over k of the minimum (or maximum, respectively) values for x such that $f_t^k(x) \geq P_1$: from the definition of F^a, this guarantees $g_t^a(emin_t) \geq P_0$ and $g_t^a(emax_t) \geq P_0$ (Lines 4-7). The time complexity of this algorithm is O($N \cdot T$). Disaggregating a schedule obtained from the optimization of an aggregated UFO can be done in a similar way. Algorithm 3 shows how this is done: first, at every time t, we define P_1 as the N−th root of the aggregated probability for s_t to be feasible (Line 3). Then, for every $k \in \{1, \dots, N\}$, we define the t−th element of the respective schedule as the minimum x such that $f_t^k(x) \geq P_1$ and the constraint $s_1^k + \ldots + s_N^k = s_t$ is not violated, if $s_t \leq 0$; otherwise, we define it as the maximum x with such properties. The time complexity of this algorithm is O($N \cdot T$).

An aggregator can also decide to use a different bidding strategy, as follows. Let $F_1, \dots, F_N$ be UFOs and, with the previous notation, let P_t^k be the probability describing time uncertainty at time t for the k-th UFO. The aggregator can then do the following: first, consider $f_1, \dots, f_N$, i.e., the functions describing amount uncertainty for $F_1, \dots, F_N$, respectively. Second, aggregate $f_1, \dots, f_N$ with the procedure described in Algorithm 2, generating a UFO denoted by F^a. Third, at each time t, the aggregator will only bid a part of the available flexibility, which is proportional to the time uncertainty

Algorithm 3: Disaggregation of a UFO

Input: T - time horizon;
$\{F_1, \ldots, F_N\}$ - UFOs(vectors of length T);
$s = (s_1, \ldots, s_T)$ - schedule
Output: $s^1, \ldots, s^N$ - schedules
Function *disaggregateUncFO(F, P_0)*:
  for $t \leftarrow 1 : T$ **do**
    $P_1 \leftarrow \sqrt[N]{g_t^a(s_t)}$
    for $k \leftarrow 1 : N$ **do**
      if $s_t \leq 0$ **then**
        $s_t^k \leftarrow \underset{x | s_1 + \ldots + s_{k-1} > s_t - x}{\arg\min} \quad f_t^k(x) \geq P_1$
      else
        $s_t^k \leftarrow \underset{x | s_1 + \ldots + s_{k-1} < s_t - x}{\arg\max} \quad f_t^k(x) \geq P_1$
return $s^1, \ldots, s^N$

of each UFO $F_1, \ldots, F_N$. Denoting by FB_t^a the total amount of flexibility that the aggregator can bid at time t, and by FB_t^k the amount of flexibility provided by F_k at time t, the aggregator will bid an amount of flexibility equal to

$$\frac{\sum_{k=1}^{N} (P_t^k \cdot FB_t^k)}{\sum_{k=1}^{N} FB_t^k} \cdot FB_t^a$$

at each time t.

The rationale of this strategy is to bid only a part of the flexibility from each FO, which at each time is the *expected value* of the flexibility while taking uncertainty into account. We give a simple example to describe this strategy. Suppose we have ten UFOs, $F_1, \ldots, F_{10}$, with no amount uncertainty: in other words, we can see the functions f_t as slice constraints, defined as the points for which f_t has value 1. Suppose that, at time 1, all F_k have a slice constraint with $emin_1^k = 0$ kWh, $emax_1^k = 10$ kWh, and $P_1^k = 0.9$. By using this strategy, the aggregator will bid flexibility like an SFO with $emin_1 = 0$ kWh and $emax_1 = 10 \cdot 0.9 \cdot 10 = 90$ kWh. This corresponds to the expected value of the available flexibility: we have ten UFOs that can provide 10 kWh of flexibility each, but for each one of them flexibility has a 90% probability of being available, therefore the expected flexibility would be 90 kWh, as on average only 9 FOs out of 10 will be able to be delivered. This is a simple strategy, but performs better than just aggregating the slice constraints obtained by f_t: the aggregator bids the amount of flexibility that is available on average, thus reducing the risk for high imbalance penalties.

It is important to note that the aggregation strategies described in this section refer specifically to UFOs. If an FO does not consider uncertainty, it can be modeled as a UFO whose parameters (P_t, f_t) can only have values in $\{0, 1\}$: in this case, both this aggregation strategy and UFO aggregation described by Algorithm 2 behave like standard FO aggregation.

3.5 Correlation between probability functions

A UFO $F = \{g_1, \ldots, g_T\}$ is defined so that $g_t(e_t)$ describes the probability that, without any other information about e_k for $k \neq t$, e_t is a feasible value for energy consumption. However, if we have this kind of information, this probability may change. This dependency can be expressed in the following way. First, let e_t^{tot} be the total amount of energy consumed before time unit t. We define the two quantities $SoC_{max}(e_t^{tot})$ and $SoC_{min}(e_t^{tot})$ as the maximum and minimum possible state of charge of the battery if the total energy consumed up to that moment is e_t^{tot}. Calling $I_{SC} = [SC_{min}, SC_{max}]$ the possible interval of the state of charge, we define $f_t(e_t, e_{tot})$ as the minimum of 1 and

$$\frac{\ell(I_{SC} \cap [SoC_{min}, SoC_{max} - e_t] \cap [SoC_{min} - e_t, SoC_{max}])}{\ell(I_{SC})}$$

where ℓ is the length of the considered interval. In other words, the probability of e_t being feasible is the probability that, choosing $v \in I_{SC}$ randomly with uniform distribution, e_t does not violate the constraints of the battery if $SoC(t) = v$. In order to exploit information on correlation between the functions f_t for optimization, we propose a two-step optimization. First, we obtain a schedule by performing an optimization with outer DFOs; second, we see if at each t, $f_t(e_t, u_{tot}) \geq P_0$ holds. If yes, the schedule at time t is confirmed; if not, we choose the closest e_t^1 such that $f_t(e_t^1, u_{tot}) \geq P_0$, and use it for creating an additional slice constraint for optimization, ensuring that the resulting schedule respects the probability threshold condition. The optimization problem can be formulated as follows. Like in Section 3.3, we choose the function over the energy variables e_t that we want to optimize, and we find the minimum or maximum of this function, depending on the objective. First, this function is optimized by using the outer DFOs as constraints over the energy variables. After this, sequentially over t, the solution for e_t that has been found is checked for the inequality $f_t(e_t, u_{tot}) \geq P_0$ and, if it does not hold, an additional constraint over e_t (depending on $u_t ot$) is added and the optimization is run again, with this additional constraint. We can use the same example of Section 3.3 for the first step, except that the constraint needs to be a DFO, so we can use the one from Figure 2(d) instead. Here, for example, the constraints relative to $t = 4$ would be:

$$0\text{kWh} \leq e_4 \leq 5\text{kWh}$$
$$0\text{kWh} \leq e_1 + e_2 + e_3 + e_4 \leq 14\text{kWh}.$$

Once this optimization has been performed, for every t, we check if $f_t(e_t, u_{tot}) \geq P_0$ is true: if it is not, we find two threshold values a_t, b_t such that $f_t(x, u_{tot}) \geq P_0$ for every $x \in (a_t, b_t)$, we add $a_t \leq e_t \leq b_t$ to the list of constraints, and we perform the same process again, until $t = T$. The strong point of this optimization is that the amount of excess flexibility considered can be decreased at will, at the cost of losing some feasible flexibility: this allows UFOs to behave in either a more conservative or a more aggressive way, depending on the value of P_0. Therefore, we will choose lower values of P_0 if we want a more aggressive approach, and higher values if we want a more conservative approach. To be more specific, UFOs are *inner* approximations for $P_0 = 1$ and *outer* approximations for $P_0 = 0$. For intermediate values, they may be neither inner nor outer approximations; however, if a UFO is an inner approximation for $P_0 = a$, it is an inner approximation for $P_0 = k$ for every $k \in [a, 1]$. Similarly, if a UFO is an outer approximation for $P_0 = a$, it is an outer approximation for $P_0 = k$ for every $k \in [0, a]$.

4 SIMULATIONS AND RESULTS

4.1 Scalability and retained flexibility

In order to evaluate the performances of UFOs, we have run some experiments and measured the amount of provided flexibility. There are several metrics for this purpose [36], but in a real case, one of the most important is economic revenue [20]: it is defined by the profit function $PF(e)$, defined as in Section 3.3. We denote the imbalance prices by $Ipr = (Ipr_1, \ldots, Ipr_T)$. We simulated the battery B from Section 2.1, and the EV from Section 3.2. Data for prices have been taken from a sample in NordPool, and include spot prices and imbalance prices between January 1, 2018 and December 31, 2018.

Throughout this section we will mention profits and imbalance costs relative to single prosumers/loads. In reality, as explained when describing the FO life-cycle (Section 2.2), flexibility provided by the prosumers is collected by the aggregator. The aggregator is the actor who actually bids flexibility in the market, makes profit from it, and has to pay for imbalance when the used flexibility is different from the predicted amount. We will refer to profits and imbalance costs relative to a single load, as the contribution that the load gives to the aggregator on the profit it obtains and imbalance penalties it has to pay respectively. The aggregator operates on the day-ahead spot market: for this reason all the operations described in this paper, i.e., flexibility generation, aggregation and optimization, are performed more than 12 hours before the beginning of the day when flexibility is actually executed, and flexibility will be modeled for all the 24 hours of the following day. Also, spot prices are known the day before, while imbalance prices are forecast.

Our experiment for batteries works as follows. B starts with the settings of the *switching* example, and we choose a time horizon T and a probability threshold P_0. Now, we generate a FO for the next T time units, and we optimize it with the objective to maximize the profit function PF, where the constraints on e are defined by the FO. We then check whether the schedules obtained by the optimization violate the constraints of the battery model; if yes, we calculate the imbalance penalties as the minimum possible cost of the difference between the schedule and a feasible one. This cost is calculated as $Cost(e) = Ipr \cdot e$. We then repeat this procedure for the next T time units again and again, until the simulation covers a total of 365 days. The approaches used for the experiments are: the theoretical optimum based on battery constraints (**Theo.Opt.**), **DFOs**, UFOs with different values of P_0 (**UFO-P_0**), an uncertain baseline (**UOnDFO**) and correlation-aware UFOs with two-step optimization (**2UFO-P_0**). The approximation **Theo.Opt.** uses the most accurate representation possible of flexibility, but it is not exact since the representation depends on $SoC(1)$. As we said at the beginning of Section 4.1 an FO is generated at least 12 hours prior to its execution, so the value for $SoC(1)$ has to be predicted and is therefore subject to error, and so is **Theo.Opt.**, which is however the most accurate representation possible of flexibility at that point in time as the constraints are modeled without error. The approximation **UOnDFO** models flexibility by DFO constraints (described in Section 2), whose generation considers uncertainty over the maximum charge value. This uncertainty is modeled with the technique used in [19], by a uniform distribution, with the following procedure. When generating the DFO, we assume that the maximum state of charge is modeled by an uniform distribution between SoC_{max} and SoC_{max} plus the maximum possible amount of energy that may have been dispersed by losses. We have interest in comparing UFOs with this baseline, since they both consider uncertainty, while the other proposed approaches are deterministic. We have run this experiment for $T = 6$ and different values of P_0.

Type	Pre-Imb	Imb.Pen.	Profit	% of TO
High penalties				
Theo.Opt.	14.27	0	14.27	100%
DFO	18.34	6.94	11.40	79.8%
2UFO-0.95	12.98	0.60	12.38	86.8%
UOnDFO-0.95	18.35	6.98	11.36	79.6%
2UFO-0.6	15.16	2.80	12.36	86.6%
UOnDFO-0.6	18.34	6.97	11.37	79.7%
All penalties				
Theo.Opt.	38.10	0	38.10	100%
DFO	55.28	31.48	23.80	62.5%
2UFO-0.95	20.59	0.75	19.84	52.1%
UOnDFO-0.95	55.34	31.60	23.74	62.3%
2UFO-0.6	50.02	24.73	25.29	66.4%
UOnDFO-0.6	55.34	31.57	23.77	62.4%

Table 3: Results for *switching* process ($T = 6$), measured in thousands of euro.

Figure 5 shows the optimization time. As explained at the beginning of Section 4.1, FOs are issued day-ahead, and in a real case they would be generated for the whole duration of the following day. However, all the EU markets are moving in the direction of using 15 minutes long time units instead of hourly: for this reason, in order to be feasible in practice, a flexibility model has to be able to perform optimization for $T = 96$. Time for optimizing the theoretical optimum grows exponentially in T, for $T = 24$ (6 hours) it is higher than 5.6 hours and for $T = 25$ (6 hours 15 minutes) or higher takes more than 29.05 hours, making it infeasible in practice. In comparison, for the time horizon $T = 96$, UFOs can be optimized in 0.277 seconds, DFOs and **UOnDFO** in 1.301 seconds and two-step UFOs in 5.270 seconds. Table 3 shows the results for profit. The columns describes respectively: **Type** of approximation, value of the PF function calculated before imbalance penalties (**Pre-Imb**), imbalance penalties (**Imb.Pen.**), **Profit** calculated after considering imbalance penalties, and percentage of profits compared to the theoretical optimum (**% of TO**).

Table 3 presents the results for the *switching* case. In particular, we have run the experiment for the *switching* case in two scenarios. One is our primary use case, in which the difference between imbalance and spot prices is higher than a certain threshold th (**high penalties**); the other scenario, in which the difference between imbalance and spot prices has no pre-determined lower bound (**All penalties**), is reported for completeness. We ran the experiment for $th = 20, 30, 40, 50$ respectively. For the high penalties case, results show that UFOs behave a lot better compared to DFOs. In particular, Table 3 shows the results for $th = 50$: the approximation made with **UFOs** can capture up to 86.8% of the total profits, against the 79.6% obtained by **UOnDFO** at its best and the 79.7% for **DFOs**. In this case, however, high values of P_0 such as $P_0 = 0.95$ obtain the best results: since the imbalance penalties are high, for lower

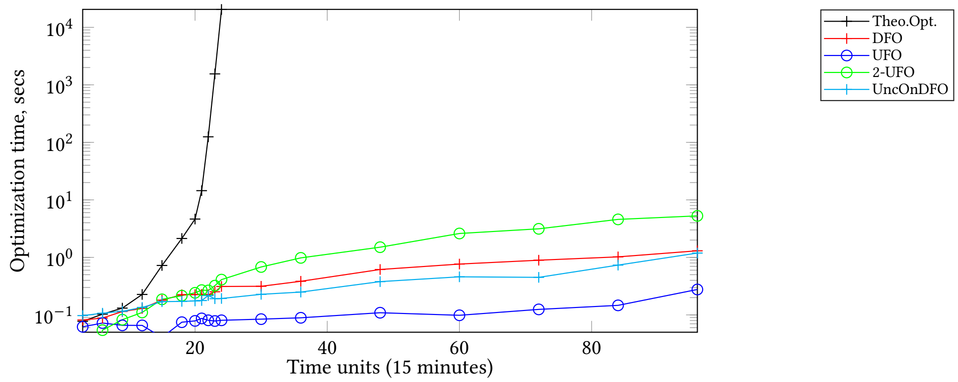

Figure 5: Time for optimization and constraints generation

values of P_0 they become too high compared to the profits that would be obtained by the increased flexibility, although UFOs still outperform the baselines. For the all penalties case, for $P_0 = 0.6$ and lower, **UFOs** perform better than **DFOs** and **UOnDFO**: they are able to capture up to 66.4% of the total profits, against the 62.5% obtained from the DFOs and the 62.4% from **UOnDFO**. In this case, high values of P_0 have lower performance, since outer approximations are not punished for causing imbalance as in the main (high penalties) case. It has to be noted that **Theo.Opt.** is never penalized, since it models the battery constraints exactly and therefore never occurs on violations of those. We chose to show results for one relatively high value and one relatively low value of P_0: we chose $P_0 = 0.95$ because it is a common threshold for statistic tests (95% confidence), and $P_0 = 0.6$ because with lower values the increase in profit from UFOs is minimal, as the increased amount of flexibility is balanced by the increasing penalties. It is important to remark that the *all penalties* case refers to all the 365 days of 2018, while the *high penalties* case only refers to 118 days of the year 2018, when the differences between the tariffs are high enough.

Regarding EVs, our experiment has been designed as follows. We consider the case of a *Tesla model S*, with the following specifics: capacity is 75 kWh, state of charge has to stay between 20% and 80%, charging power is 7 kW, charging efficiency is 84%. We are considering the EV *charging only* example described in Section 3.2. The experiment works like the battery experiment described earlier in this section: the FO is optimized in order to minimize costs, and then energy costs are calculated like in the battery case. As said at the beginning of Section 4.1, imbalance penalties are calculated depending on how the actions of the prosumer impact the imbalance penalties that the aggregator has to pay. For example, if the EV is not plugged in when the aggregator predicts it to be, the aggregator's prediction of consumption will be higher than the amount of energy actually consumed, generating imbalances. In a similar way, overestimating or underestimating the amount of energy needed to charge the EV will result in the aggregator respectively overestimating or underestimating the amount of energy consumption, causing imbalances. When the FO is aware of time uncertainty, a reduced amount of flexibility is bid, proportional to the value P_t from Section 3.2, since we assume that the aggregator bids that amount of flexibility, following the strategy described in Section 3.4. For this experiment we have compared: the theoretical optimum (**Theo.Opt.**), **DFO**s, **2-UFO**s and an uncertain baseline built on the same principle as **UOnDFO**s for the battery case. In this case, the theoretical optimum is the same as the battery case, and also assumes that we know whether the EV will be plugged in or not, as well as plug-in and plug-out times, exactly and in advance. Obviously this is not realistic and such a baseline cannot exist in real life, and is only used to evaluate the performance of the other baselines.

Type	Pre-Imb	Imb.Pen.	Cost	% of TO
High penalties				
Theo.Opt.	73.17	0	73.17	100%
DFO	63.31	55.01	118.32	61.8%
2UFO-0.95	53.05	39.41	92.46	79.1%
UOnDFO-0.95	51.87	46.50	98.37	74.4%
2UFO-0.6	52.83	30.77	83.60	87.5%
UOnDFO-0.6	60.80	39.72	100.52	72.8%
All penalties				
Theo.Opt.	340.76	0	340.76	0%
DFO	318.42	122.23	440.65	77.3%
2UFO-0.95	275.11	96.11	371.22	91.9%
UOnDFO-0.95	260.79	110.93	371.72	91.7%
2UFO-0.6	306.06	63.22	369.28	92.0%
UOnDFO-0.6	296.34	82.14	378.48	90.0%

Table 4: Costs for charging the EV in the EV *charging only* example, measured in thousands of euros.

4.2 Aggregation

We have performed experiments for measuring the effectiveness of UFOs aggregation. We compared it against four approaches: **UOnDFO**, **Minkowski**, an approach based on an approximated Minkowski sum [2] (**AppMink**), and an exact baseline called LTI Aggregation [20] (**LTIAgg**). **Minkowski** aggregates flexibility polygons by computing their Minkowski sum, **AppMink** computes an approximate Minkowski sum that is faster to calculate, **UOnDFO** is DFO aggregation over the DFOs on which uncertainty is then modeled (which is also aggregated as an uniform distribution), and

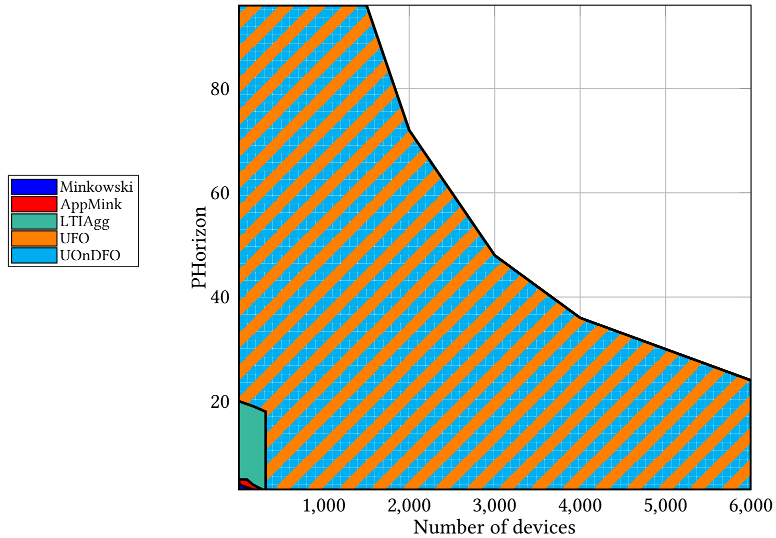

Figure 6: Feasibility for time horizons and devices

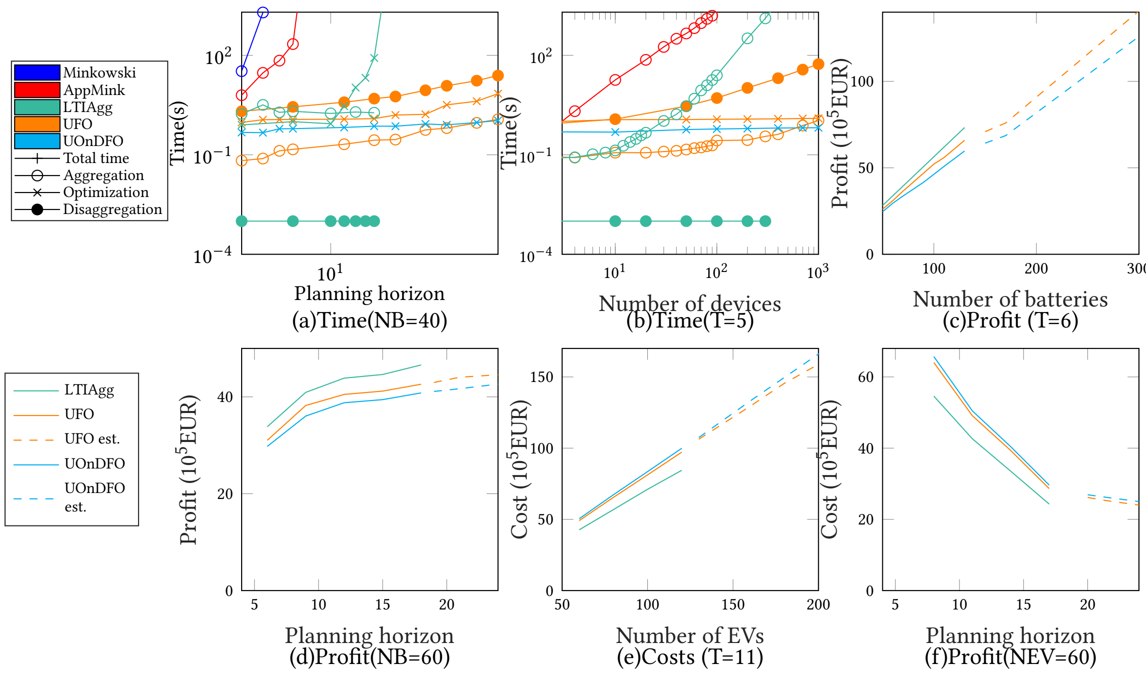

Figure 7: Results for aggregation time (a-b), profit for batteries (c-d) and costs for EVs (e-f).

LTIAgg combines many LTI models into one single LTI model. An LTI model [3, 15] is a model that describes the evolution over time of a physical system, by a set of linear recurrence relations; in our case, the recurrence relations of the model of a single battery are the ones in Eq. 1. We measure the amount of retained flexibility by economic revenue. Like the previous subsection, we have performed two types of experiments: one for batteries, and one for EVs. The purpose of the first experiment is to aggregate NB batteries, which are $\frac{NB}{2}$ copies of B, and NB copies of a Pika Energy Harbor battery, which specifics are: $SoC_{\max} = 20kWh, SoC_{\max} = -SoC_{\min} = 6.7kWh$, round-trip efficiency 96.5%. First, we choose a time horizon T and a probability threshold P_0 for generating FOs. Now, for each battery, we instantiate an LTI model, a DFO and a UFO. After this, we perform aggregation on those models by using the respective approaches, and we perform optimization for each of the resulting aggregated models, with the objective of maximizing PF. After this, we calculate imbalance costs as described in the experiment in Section 4.1, and finally we perform disaggregation on the aggregated models. This procedure is then repeated again and again.

In the EV experiment, we simulate the behavior of a number NEV of prosumers: half of the considered EVs is the same as described in the EV *charging only* example, the other half are *Renault Megane*s (for our purposes the specs are the same as Tesla model S, except the capacity is 60kWh). Their probability to be plugged in during weekdays and weekends is like in the EV *charging only* example, and their plug-in time follows the pattern described in [1], although different users may have different mean plug-in time, for which we followed the distribution in [30]. We generate the FOs, aggregate them, perform cost optimization and then disaggregate them. After this, we check the amount of flexibility that has effectively been delivered, and calculate imbalance penalties accordingly. As described in Section 3.4, the aggregator bids only a part of its available flexibility which is proportional to time uncertainty. The procedure is then repeated again and again.

Since bids for flexibility need to be done at most one hour before the deadline, in order to reduce errors [23], and flexibility has to be modeled before it can be aggregated, we chose to consider 30 minutes as the time limit within which an approach is considered *feasible*. Figure 6 shows results for feasibility. It is possible to aggregate 6000 UFOs for a time horizon of $T = 24$, and 750 UFOs for $T = 96$, and the same is true for **UOnDFO**, which in the graph is overlapped by UFOs, as their aggregation process works similarly to DFOs. In comparison, **LTIAgg** fails for $T > 21$ or 330 devices, **Minkowski** and **AppMink** are infeasible for $T > 6$, and the first fails for 180 devices, and the second for 290 devices. Figure 7(a) and (b) shows results for the time needed for each single process . As it can be seen, aggregation time grows exponentially with respect to T for **Minkowski** and **AppMink**, and grows linearly for **LTIAgg**, **UOnDFO** and **UFOs**. Aggregation time with respect to number of devices grows linearly for **UFOs** and **UOnDFO**, and grows superlinearly for **Minkowski**, **AppMink** and **LTIAgg**. Optimization time grows exponentially in T for **Minkowski**, **AppMink** and **LTIAgg**, which is the reason why **LTIAgg** fails for $T > 21$. Disaggregation time grows linearly for every approach except **LTIAgg**, for which it is always 0.001s. Figure 7(c) and (d) shows results for profit for batteries in the high penalties case (results for $NB > 130$ are estimated). It can be seen that for $T = 6$, UFOs can retain up to 92.5% of the profits, while **UOnDFO** can only retain around 82% of the profits. Scalability with respect to time is also good: for $NB = 60$, UFOs can capture 90.5% of the profits even for $T = 18$, against the 88% obtained from **UOnDFO**. Figure 7(e) and (f) shows results for costs for charging EVs in the high penalties case (results for $NEV > 120$ are estimated). Similar to Section 4.1, we measure retained flexibility as the ratio between the costs in the **LTIAgg** case and the costs obtained by each baseline. With this premise, the graphs show that in general **UFOs** are capable of retaining up to 86.8% of the total amount of flexibility even for long time horizons and many devices, while the **UOnDFO** approach can capture up to 84.5% of the total flexibility.

In summary, the experiments show that: i) theoretical optimum baselines fail for many loads and/or long time horizons, ii) UFOs scale well with both number of loads and time horizon, and iii) UFOs can retain most of the profits after aggregation, outperforming the baselines.

5 CONCLUSIONS AND FUTURE WORK

The purpose of this work is to propose a model for flexibility which i) captures flexibility from different devices in an unified format; ii) is scalable for long time horizons and aggregation of many loads; iii) can capture most of the available flexibility; iv) takes into account uncertainty that affects flexibility when modeled for many time units.In this paper, we propose Uncertain FlexOffers (UFOs), which capture uncertainty while fulfilling i)-iii) like the previously existing FlexOffer (FO) model. We show how UFOs can be generated, optimized and aggregated, and how UFOs can consider dependency between energy consumption at different times. The model is device-independent, which accomplishes property i). We then compared UFOs to exact baselines and FOs, and showed how the inclusion of uncertainty allows UFOs to outperform FOs, while remaining still scalable. Flexibility operations (generation, optimization and aggregation) are performed day-ahead: as energy markets are adopting 15 minutes long time units, those operations have to be performed for time horizons of 96 time units. Our results show that UFOs can perform optimization in 5.27 seconds for 96 time units (6 hours 15 minutes) ahead, while exact models need more than 29.05 hours for optimizing 25 time units in advance, which makes them infeasible in a practical case: this achieves property ii). For a charging battery, UFOs are able to model flexibility without losses; when penalties are high, UFOs can capture up to 86.8% of the total profits when penalties are high, compared to the 79.6% retained by the uncertain baseline. For EVs, UFOs are able to retain up to 87.5% of the total flexibility in the high penalties case, while the uncertain baseline can only capture 74.4% of it. Furthermore, UFOs permit to aggregate up to 6000 loads or for 96 time units, while exact models are infeasible above 330 loads or 21 time units. When penalties are high, aggregated UFOs retain more than 90.5% of the available flexibility for batteries, against the 82% from the uncertain baseline. For EVs, UFOs can retain 86.8% of the total amount of flexibility, while the uncertain baseline can capture 84.5% of it. The fact that UFOs can retain most of the available flexibility satisfies iii), and uncertainty modelization makes them perform well in our primary use case, i.e., when inaccurate approximations are penalized, which accomplishes iv). Future work will focus on improving accuracy of UFOs in capturing flexibility, and creating better analytic generation methods for further improving their scalability.

ACKNOWLEDGMENTS

This work was supported by the DomOS and FEVER projects, funded under the Horizon 2020 programme, GAs 864537 and 894240 respectively, and Flexible Energy Denmark, funded by Innovation Fund Denmark (Case No. 8090-00069B). We thank Hamdi Ben Hamadou for proofreading.

REFERENCES

[1] Andres Arias et al. 2017. Optimal probabilistic charging of electric vehicles in distribution systems. *IET Electrical Systems in Transportation* (2017).
[2] Suhail Barot et al. 2016. An outer approximation of the Minkowski sum of convex conic sets with application to demand response. *IEEE CDC* (2016).
[3] Francesco Borrelli, Alberto Bemporad, and Manfred Morari. 2017. *Predictive Control for Linear and Hybrid Systems.* Cambridge University Press.

[4] Lu Chen et al. 2021. Optimal energy management of smart building for peak shaving considering multi-energy flexibility measures. *Energy and Buildings* (2021).
[5] Yongbao Chen et al. 2018. Measures to improve energy demand flexibility in buildings for demand response (DR): A review. *Energy and Buildings* (2018).
[6] Edoardo Corsetti et al. 2021. Modelling and deploying multi-energy flexibility: The energy lattice framework. *Advances in Applied Energy* (2021).
[7] Cherrelle Eid et al. 2015. Aggregation of Demand Side flexibility in a Smart Grid: A review for European Market Design. *EEM* (2015).
[8] R. Fonteijn et al. 2018. Flexibility for congestion management: A demonstration of a multi-mechanism approach. (2018).
[9] Kyriaki Foteinaki et al. 2020. Evaluation of energy flexibility of low-energy residential buildings connected to district heating. *Energy and Buildings* (2020).
[10] Davide Frazzetto et al. 2018. Adaptive User-Oriented Direct Load-Control of Residential Flexible Devices. *e-Energy* (2018).
[11] Irina Harris et al. 2014. A hybrid multi-objective approach to capacitated facility location with flexible store allocation for green logistics modeling. *Transportation Research Part E: Logistics and Transportation Review* (2014).
[12] Maomao Hu and Fu Xiao. 2020. Quantifying uncertainty in the aggregate energy flexibility of high-rise residential building clusters considering stochastic occupancy and occupant behavior. *Energy* (2020).
[13] Rune Grønborg Junker et al. 2018. Characterizing the energy flexibility of buildings and districts. *Applied Energy* (2018).
[14] Rune Grønborg Junker et al. 2020. Stochastic nonlinear modelling and application of price-based energy flexibility. *Applied Energy* (2020).
[15] M. Koller et al. 2013. Defining a degradation cost function for optimal control of a battery energy storage system. *IEEE Grenoble Conf.* (2013).
[16] Soumya Kundu, Karanjit Kalsi, and Scott Backhaus. 2018. Approximating Flexibility in Distributed Energy Resources: A Geometric Approach. *CoRR* (2018).
[17] Ki-Beom Lee et al. 2020. Deep Reinforcement Learning Based Optimal Route and Charging Station Selection. *Energies* (2020).
[18] Han Li et al. 2021. Energy flexibility of residential buildings: A systematic review of characterization and quantification methods and applications. *Advances in Applied Energy* (2021).
[19] Fabio Lilliu et al. 2019. An uncertainty-aware optimization approach for flexible loads of smart grid prosumers: A use case on the Cardiff energy grid. *Sustainable Energy, Grids and Networks* (2019).
[20] Fabio Lilliu, Torben Bach Pedersen, and Laurynas Siksnys. 2021. Capturing Battery Flexibility in a General and Scalable Way Using the FlexOffer Model. *SmartGridComm* (2021).
[21] Fabio Lilliu, Torben Bach Pedersen, Laurynas Siksnys, and Bijay Neupane. 2022. Uncertain flexoffers, a scalable, uncertainty-aware model for energy flexibility. (2022).
[22] Rui Amaral Lopes et al. 2016. A Literature Review of Methodologies Used to Assess the Energy Flexibility of Buildings. *Energy Procedia* (2016).
[23] Meiqin Mao et al. 2019. Schedulable capacity forecasting for electric vehicles based on big data analysis. *J. Mod. Power Syst. Clean Energy* (2019).
[24] Ilaria Marotta et al. 2021. Investigation of design strategies and quantification of energy flexibility in buildings: A case-study in southern Italy. *Journal of Building Engineering* (2021).
[25] Jinhao Meng et al. 2019. A Simplified Model-Based State-of-Charge Estimation Approach for Lithium-Ion Battery With Dynamic Linear Model. *TIE* (2019).
[26] Alice Mugnini et al. 2021. Energy Flexibility as Additional Energy Source in Multi-Energy Systems with District Cooling. *Energies* (2021).
[27] Fabian L. Müller et al. 2019. Aggregation and Disaggregation of Energetic Flexibility From Distributed Energy Resources. *IEEE Trans. Smart Grid* (2019).
[28] Bijay Neupane, Laurynas Siksnys, and Torben Bach Pedersen. 2017. Generation and Evaluation of Flex-Offers from Flexible Electrical Devices. *e-Energy* (2017).
[29] Torben Bach Pedersen, Laurynas Siksnys, and Bijay Neupane. 2018. Modeling and Managing Energy Flexibility Using FlexOffers. *SmartGridComm* (2018).
[30] Jairo Quirós-Tortós et al. 2015. A statistical analysis of EV charging behavior in the UK. In *ISGT LATAM*.
[31] Fabian Scheller et al. 2020. Competition between simultaneous demand-side flexibility options: The case of community electricity storage systems. *CoRR* (2020).
[32] Paul Schott et al. 2019. A Generic Data Model for Describing Flexibility in Power Markets. *Energies* (2019).
[33] Laurynas Siksnys and Torben Bach Pedersen. 2016. Dependency-based FlexOffers: scalable management of flexible loads with dependencies. *e-Energy* (2016).
[34] Hanne Sæle et al. 2020. Assessment of flexibility in different ancillary services for the power system. https://doi.org/10.1109/EEM49802.2020.9221996
[35] Mari Tveit et al. 2022. Behind-the-meter residential electric vehicle smart charging strategies: Danish cases.
[36] Emmanouil Valsomatzis, Katja Hose, Torben Bach Pedersen, and Laurynas Siksnys. 2015. Measuring and Comparing Energy Flexibilities. *EDBT/ICDT* (2015).
[37] Ziyang Wang et al. 2021. A new interactive real-time pricing mechanism of demand response based on an evaluation model. *Applied Energy* (2021).