D4.6 - Initial Cooperative EV charging station control algorithms

Deliverable ID	**D4.6**
Deliverable Title	**Initial Cooperative EV charging station control algorithms**
Work Package	**WP4**
Dissemination Level	**PUBLIC**
Version	**1.0**
Date	**14/08/2018**
Status	**final**
Type	**Prototype**
Lead Editor	**Fraunhofer FIT**
Main Contributors	**FIT**

Published by the Storage4Grid Consortium

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 731155.

Document History

Version	Date	Author(s)	Description
0.1	2018-06-25	Fraunhofer	First TOC
0.2	2018-08-01	Fraunhofer	First draft
0.3	2018-08-07	Fraunhofer	Comments of UNINOVA were included
1.0	2018-08-14	Fraunhofer	Final version to be submitted. Comments of LIBAL were included

Internal Review History

Review Date	Reviewer	Summary of Comments
2018-08-04 (v0.2)	Vasco Delgado-Gomes (UNINOVA)	Approved: • Minor corrections • Acronyms addition
2016-08-13 (v0.3)	Yini Xu (LiBal)	Approved: • Minor corrections • Few comments on section 2 and 3

Legal Notice

The research work leading to these results has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 731155 - Storage4Grid project. The information in this document is subject to change without notice. The Members of the Storage4Grid Consortium make no warranty of any kind with regard to this document, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. The Members of the Storage4Grid Consortium shall not be held liable for errors contained herein or direct, indirect, special, incidental or consequential damages in connection with the furnishing, performance, or use of this material. The European Union and the Innovation and Networks Executive Agency (INEA) are not responsible for any use that may be made of the information contained therein.

Table of Contents

Executive Summary

This deliverable presents the first developed algorithms implementing optimal control of electric vehicles (EVs) charging in a car park including uncertainties. The algorithms are implemented into a software tool named Professional Realtime Optimization Framework for Electric Vehicles (PROFEV) which is being developed within Storage4Grid.

The PROFEV is a software component running stochastic optimization algorithms for the control of the charging of EVs and the charging/discharging of energy storage systems (ESS). It receives in quasi real time all available information from local devices (ESS systems, PV energy meters, load-side energy meters, EV chargers, etc.). In S4G PROFEV works also together with GESSCon, a global ESS controller service, linking the charging/discharging profiles for the next 24h sent by GESSCon with the internal optimal control model of PROFEV. In some applications (e.g. related to distributed energy management) it can be configured to act as the main local EMS.

For clarification PROFEV differs from PROFESS presented in D4.2 – "Updated User-side ESS control system", in the fact that it runs stochastic optimization models for including uncertainty in the optimal control mechanism.

In the following deliverable (D4.7 – "Final Cooperative EV charging station control algorithms"), the extensions of this algorithm will be implemented in PROFEV.

1 Introduction

D4.6 describes the "Initial Cooperative EV charging station control algorithms" prototype, developed within the Storage4Grid project. The optimal control algorithms use stochastic dynamic programming for optimizing the charging of electric vehicles (EVs) together and the charging and discharging of energy storage systems (ESS). The uncertainty generated by EVs is modelled using Markov chains. The PROFEV tool is responsible for implementing the algorithms in a real-time optimal control system.

1.1 Scope

This prototype deliverable has been developed by Task T4.3 – "Cooperative EV charging station control". Further update of this prototype is expected to be released as D4.7 (M33).

1.2 Related documents

ID	Title	Reference	Version	Date
[D2.1]	Initial Storage Scenarios and Use Cases	D2.1	1.1	2017-06-08
[D2.2]	Final Storage Scenarios and Use Cases	D2.2	1.0	2018-07-31
[D3.2]	Updated S4G Components, Interfaces and Architecture Specification	D3.2	1.0	2018-08-31
[D4.2]	Updated User-side ESS control system	D4.2	1.0	2018-06-14
[D4.8]	Initial USM Extensions for Storage Systems	D4.8	1.0	2017-08-31

2 Background

2.1 Description

Cooperative EV Charging Scenario for Storage4Grid project aims at reducing the unbalance due to high power demand that is inserted by the diffusion of EVs. It considers an EV car park use case which hosts an EV fleet and that features slow (7.4 kW) and fast charging (22 kW) with 10 charging points (CP). Storage4Grid will control the installed charging stations via OCPP communication protocol. Therefore, fine-grained monitoring and control of the charging process will be enabled. This infrastructure will allow developing a number of predictive control algorithms to be deployed and tested for the purpose of establishing a cooperative behaviour between the EV charging and storage control processes. A general diagram of the use case is presented in Figure 1 - Commercial EV Charging

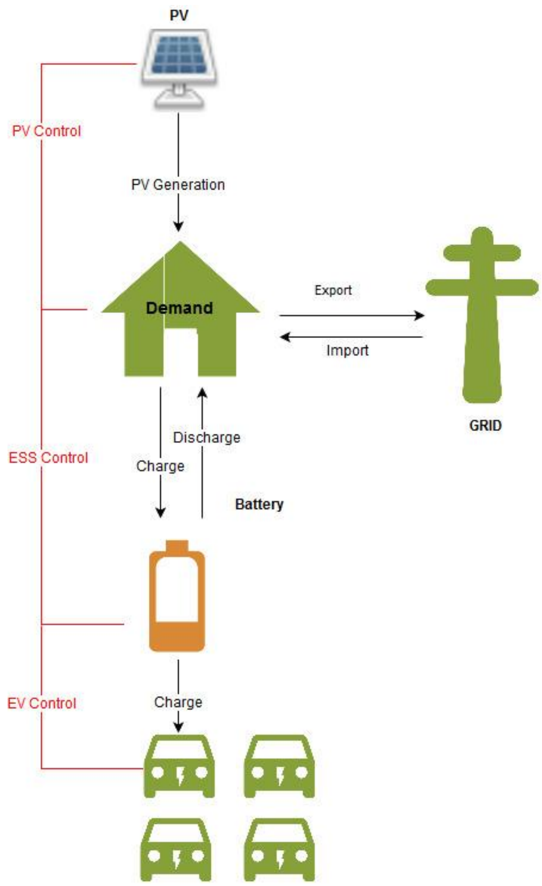

Figure 1 - Commercial EV Charging

2.2 Mathematical optimization

Optimization is "an act, process, or methodology of making something (as a design, system, or decision) as fully perfect, functional, or effective as possible" [1].

Equations 2.1-2.3 show the generic representation of an optimization problem. Main idea of optimization is minimization (maximization) of a cost (benefit) function expressed in terms of control variables and parameters that determine the system to be optimized. This function is usually called objective function (Eq. 2.1), which is the function to be optimized. In addition to the objective functions, optimization problems are defined with variables and constraints. Optimization variables are the inputs of optimization problems. Optimal solution is obtained by finding the best combination of these inputs. However, in many optimization problems, there are some limitations for the values that the optimization variables can take. These limitations are expressed with constraint functions, which are also combination of parameters and variables, see Eq. 2.2-2.3.

$$minimize \quad f(x) \qquad x \in X \tag{2.1}$$

$$s.t \quad g(x) \le 0 \tag{2.2}$$
$$h(x) = 0 \tag{2.3}$$

Where, *x* is the vector of optimization variables. Optimization variables can take values from the set *X*, which is a subset of $\mathbb{R}^n$ .The function *f* is called objective function. *g* and *h* functions determine the feasible region of the optimization problems and they are called inequality and equality constraints respectively.
Points in domain sets might satisfy the conditions that is expressed by the constraints. Points that satisfy all constraints are called feasible. The set of all feasible points in the problem constitutes the feasibility region. Mathematical problems whose feasibility region is empty are infeasible problems.
Decision problems for any physical or hypothetical system can be formulated as mathematical optimization problems whereas complexity and size of the problem defines the computational feasibility. Algorithms to solve optimization problems are designed according to the problem class and are classified according to the following issues:

- Whether the model involves constraints
- Number of the variables
- Type of the variables
- Fixedness of variables i.e. static or dynamic
- Deterministic or stochastic
- Linearity

Optimization problems may contain single or multiple decision variables. These variables may be continuous or discrete. Discrete decision variables can take values only from a specific domain such as integer, Boolean or a finite collection of subsets (combinatorial programs), whereas continuous variables can take any values. Problems that has only integer variables are integer programs (IP). Binary programs are a subclass of integer programs with decision domain in the binary ranges (0-1). If only some of the variables are restricted to be integers and rest of the variables are continuous, this problem is said to be a Mixed Integer Problem (MIP) [2].
Mathematical optimization has a scope that covers different nature of parameters. Some system parameters are deterministic while some parameters include randomness in nature. The problems that have random parameter(s) are called stochastic optimization problems. Modelling techniques for deterministic problems are usually more straightforward. On the contrary, stochastic problems involve distribution functions in constraints and objective functions as our use case of the cooperative EV charging problem.
Finally, linearity is an important classification criterion for optimization. A linear optimization problem comprises of a linear objective and constraint functions. Linear problems (LP) are simpler thanks to analytical solutions. Non-linear optimization problems (NLP) are not suitable for analytical solutions and need algorithms to linearize or numerically solve the problem.

2.3 Stochastic Optimization

There is a wide range of problems in energy systems that introduces uncertainty in different forms such as unit commitment, energy storage, charging of electric vehicles, bidding energy resources, equipment replacement, contract pricing and investment planning. These problems present a wide variation in the types of stochastic optimization problems in terms of the nature of the decisions (discrete or continuous, scalar or vector), the uncertainties (binomial failures, Gaussian noise in weather, loads and generation, heavy-tailed electricity prices) and the dynamics (the models of the storage process can be known, but the models of climate change, commodity prices and the behaviour of competing utilities are unknown). In S4G, stochastic dynamic programming is used for modelling and solving the charging of electric vehicles problem. Therefore, the following subchapters give an introduction to this technique

2.3.1 Dynamic programming

In optimization, the size of the sequential decision problem, i.e. optimization horizon and time resolution, has a direct relationship with the number of control variables. A linear increase in the number of control variables is reflected with exponential increase in computation time. In other words, increase in optimization horizon or time resolution require unreasonably long computation times for calculating the solution of optimization problems.

Dynamic programming is a computer friendly optimization technique for large problems as it partitions optimization problem into smaller, and thus easier to solve parts. There are two levels of this partitioning. First, optimization horizon is divided into a number of stages (*T*) –usually one stage for each time step in horizon. Second, each stage is divided into a number of states (*S*), which characterize the physical system at that stage. There is an optimal decision for any state at each stage. In other words, dynamic program consists of $T \times S$ different optimization problems. Figure 2 depicts division of a complete problem into T=4 stages and S=3 states.

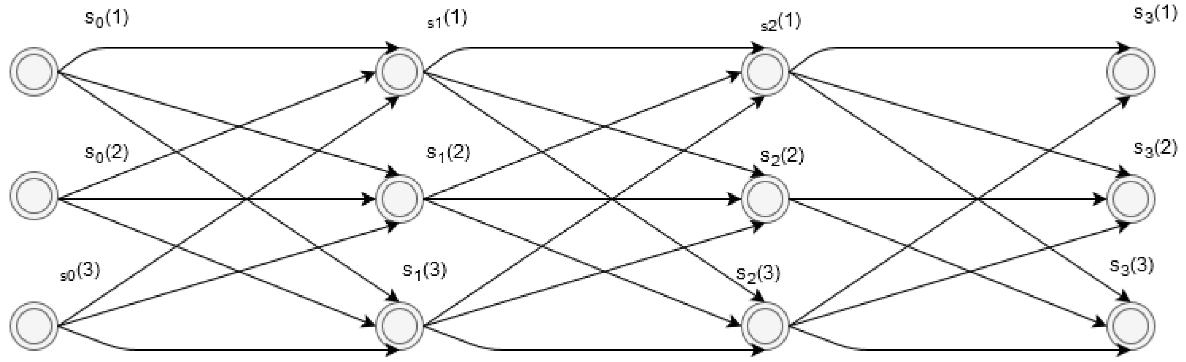

Figure 2: Problem Partitioning for Dynamic Programs

There are rules that couple states (s_t) and decision taken (x_t) at the prior stage with the state (s_{t+1}) at the next stage. This relationship is expressed as transition functions in the formulation, see Eq. 2.4.

$$s_{t+1} = f(x_t, s_t) \qquad 2.4$$

Next important element of dynamic programs is the value function. Value function v_t defines the benefit (cost) of a taken decision x_t when the system is at a particular state s_t at a particular stage t. It consists of two elements: incurred immediate cost (benefit) of taken decision $c_t(x_t)$ and optimal cost (benefit) of the transformed state s_{t+1} as result of taken decision $v_{t+1}(s_{t+1})$.

$$v_t = c_t(x_t) + v_{t+1}(f(x_t, s_t)) \qquad 2.5$$

Dynamic program computes the optimal decision for any state at each stage with the Eq. 2.6 objective function by using backward induction, i.e. starting the calculation from the final stage and proceeding backwards. Thus, dynamic programming stores the future steps' optimal values and optimal decisions in order to facilitate the computation.

$$\min c_t(x_t) + v_{t+1}(S_{t+1}) \qquad 2.6$$

2.3.2 Stochastic dynamic programming

In Section 2.3.1 a definition and formulation of deterministic dynamic programming was presented. Similarly, dynamic programming is a computer friendly solution technique in stochastic problems as it partitions a large optimization problem into small ones. However, the coupling between states of subsequent stages are not deterministic but probabilistic (represented dotted lines in Figure 3. This is suitable for representation of uncertainty factors with probabilistic timely variation models such as Markov Chains.

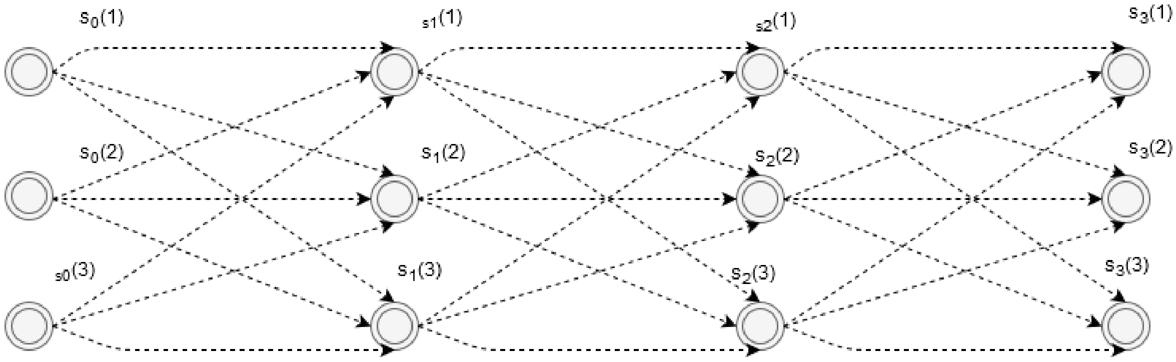

Figure 3: Problem Partitioning for Stochastic Dynamic Programs

As the rules that couple states (s_t) and decision taken (x_t) at the prior stage with the state (s_{t+1}) at the next stage are stochastic, transition function (Eq. 2.7) includes also term for randomness (ξ_t).

$$s_{t+1} = f(x_t, s_t, \xi_t) \quad 2.7$$

The value function (Eq. 2.5) takes also a probabilistic form. It includes an expectation term instead of value function of future state. w represents a scenario and ${\xi_t}^w$is the probability of this scenario to occur. $v_{t+1}(s_{t+1})$ is the optimal value of being at state s_{t+1} in the future stage.

$$v_t = c_t(x_t) + \sum_{w=1}^{W} {\xi_t}^w v_{t+1}(s_{t+1}) \quad 2.8$$

The objective function takes the stochastic form (Eq. 2.9)

$$\min c_t(x_t) + \sum_{w=1}^{W} {\xi_t}^w v_{t+1}(s_{t+1}) \quad 2.9$$

2.4 EV Charging Demand Uncertainty

The increase in EV use, demands a public charging infrastructure that adequately provides the needs of all EV users. For this reason, understanding the charging behaviours of EV users in terms of charging schedule, energy consumption, charging time, and their choice of where to charge are essential [3].

This S4G scenario considers EV as the source of uncertainty for the optimization problems. Optimization models have to cope with the uncertainties that stem from probabilistic driving profiles. The first step for cooperative charging of EVs optimization is therefore integrating a consistent uncertainty modelling technique that reflects the behaviour of EV users. However, it is complicated to quantify driving profiles due to internal factors such as mobility pattern of an EV driver/owner and external factors such as road topology, traffic, driving style, and ambient temperature.

Because understanding the charging behaviours of EV users in terms of when they charge is essential for both designing good charging policies and infrastructure, a myriad of work has been dedicated to develop

uncertainty modelling for driving profiles. Not only when the car is driven, but also the energy consumption of this drive is uncertain. Therefore, cooperative EV charging problem seeks for the optimal energy management strategy, e.g. for minimizing the power bill, while meeting car charging demand, which is uncertain both in profile and amplitude.
Following subchapters provide information on uncertainty modelling techniques for behaviour uncertainty.

2.4.1 Model for Describing Driving Behaviors: Markov Chains

A Markov chain is "a stochastic model describing a sequence of possible events in which the probability of each event depends only on the state attained in the previous event" [4]. Markov property, which emphasizes the serial dependency of adjacent variables, promises good predictions for the future if current state and probabilities of potential transitions of a system is known.
A Markov chain is characterized by the transition probabilities. If the transition probabilities do not depend on time, the process is called a homogeneous Markov chain. If the transition probabilities depend on time, the process is known as an inhomogeneous Markov chain. Equation 2.10 describes the probability of switching from j to k state at time interval t [5]:

$$p_{jk}(t) = P(x_{t+1} = k | x_t = j) \quad 2.10$$

According authors of [6], EV use is described simplest with two states: driving and non-driving. A more detailed version would consist of a larger number of states that include information on parking place, speed of drive or trip type. For the EV charging problem, level of information detail is not complete as needed. On the other hand, collecting and summarizing data for when the car is at home is quite convenient. Therefore, this S4G scenario adopts Inhomogeneous Markov Model that is described with two states: home and away (See Figure 4).

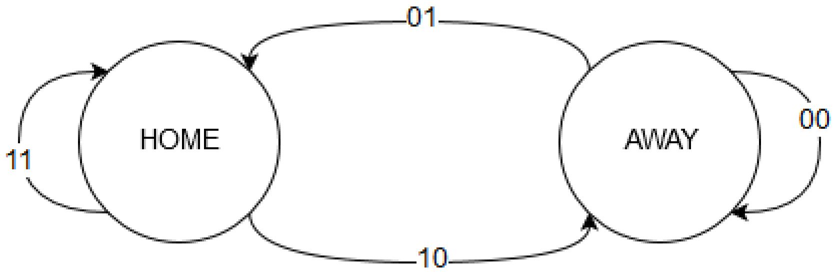

Figure 4: Inhomogeneous Markov Models for Describing Driving Patterns

Probabilities for one-time interval $p(t)$ are expressed as a transition probability matrix, where $p_{01}(t)$ is probability of switching from home to away state (Eq. 2.11).

$$p(t) = \begin{vmatrix} p_{10}(t) & p_{11}(t) \\ p_{00}(t) & p_{01}(t) \end{vmatrix} \quad 2.11$$

The required data to model the transition probability matrices is a number of time series showing the states of a car within one day. These parameters are calculated by dividing the number of observed transitions from home state to away state into the number of total transitions that started from home state:

$$p_{10}(t) = \frac{n_{10}(t)}{n_{10}(t) + n_{10}(t)} \quad 2.12$$

3 Cooperative EV charging station control algorithm

This chapter introduces a control technique for charging EVs at a car park that hosts eight EVs as stated in the use case of Bolzano. The control problem is structured in such a way to respond to the charging demand of the car park EV charging scenario of Storage4Grid. Therefore, the car park serves to a definite number of EVs that a company uses for business purposes.

3.1 Dealing with the uncertainty

EV charging problem that seeks optimal policy for foregoing energy management has to cope with two uncertainty factors: EVs position at a certain time and energy demand after charge (energy consumed by the EV while driving). Chapter 2.4 introduced base approaches for modelling these uncertainties respectively. However, the presence of multiple cars, each of which introduces its own uncertainty, makes the car park charging problem more complicated.

The future energy demand at the car park will depend on how many cars are driving at each time interval. The Markov model describes the transition probabilities between 'the numbers of cars at the car park'.

$$p_{jk}(t) = P(x_{t+1} = k | x_t = j) \tag{3.1}$$

Above, j and k are numbers that show how many cars are at the car park. The probabilities are calculated through Monte-Carlo simulations. The required data to model the transition probability matrices is a number of time series showing the number of the cars at car park (i.e. translated into the number of driving cars) within one day. These parameters are calculated by dividing the number of observed transitions from one state to another into the number of total transitions that started from being at car park:

$$p_{jk}(t) = \frac{n_{jk}(t)}{\sum_{m=1}^{M} n_{jm}(t)} \tag{3.2}$$

Consumption uncertainty is modelled by using the following assumption. Eq. 3.3 calculates the average consumption of an EV car's daily trip E_{daily} according to the collected data. Collected data provides the mean departure t^{dep} and arrival time t^{arr} of the cars that are charged at this station. Also, state-of-charge (SOC) levels of departing and arriving cars are recorded; and mean values (SOC^{dep} and SOC^{arr}) are calculated. Then this number is divided into the average trip period ΔT of a car to calculate the mean unit-time consumption of one car's away state i.e. $E_{unit-time}$. It is important to note that t^{dep} and t^{arr} are plugged into Eq. 3.4 as number of time steps regarding the optimization horizon.

$$E_{daily} = (SOC^{dep} - SOC^{arr}) * E_{capacity} \tag{3.3}$$

$$P_{unit-time} = \frac{E_{daily}/\Delta T}{t^{arr} - t^{dep}} \tag{3.4}$$

3.2 Virtual Aggregated Capacity (VAC)

Car park EV charging scenario of Storage4Grid project is an EV park whose guests have cooperative targets. For this reason, optimal charging policy has to be designed in such a way to combine all hosted EV charging demands. However, addressing individual demand separately overcomplicates the problem. In order to simplify the problem, an aggregation concept is used. According to this concept, EV group that is served by a station is modelled as a Virtual Aggregated Capacity (VAC). VAC is a virtual battery with an energy capacity that equals to the summation of EV battery capacities of EV cars that are charged by the EV car park, Eq. 3.5.

$$E_{VAC} = \sum_{n=1}^{N} E_{EV}^{n} \tag{3.5}$$

According to this concept optimal policy will be defined with respect to virtual aggregated capacity's needs and uncertainties. Optimal policy is calculated with Stochastic Dynamic Programming (SDP) that captures uncertainty of daily trips of served EV group. A single SDP will be structured at the beginning of each interval of Energy Management i.e. a shifting optimization horizon through a day. Calculated optimal policy for the initial time step of the optimization horizon will be implemented into the actual EVs at the car park. Following two subsections present the formulation of SDP and implementation of the optimal policy respectively.

3.3 Stochastic Dynamic Programming (SDP) for VAC Charging

Optimal policy design for VAC instead of a real EV allows decoupling the optimal policy calculation from the implementation of this policy.
According to the VAC concept, optimal policy is calculated as if it was also possible to charge a car even when it is not one there. Therefore, SDP model does not need a state that describe the EV position. The variables that describe the system state are the SoC of ESS (stationary battery) and SoC of VAC so that the system switches between the combinations of these states:

$$\begin{gathered} s_{ESS}^{t} \rightarrow s_{ESS}^{t+1} \\ s_{VAC}^{t} \rightarrow s_{VAC}^{t+1} \end{gathered} \tag{3.6}$$

Transition between ESS SoC states are deterministic, so it has to respect the following constraint, where E_{ESS} stands for the energy capacity of ESS in kWh and x_{ESS} represents the charging power to the ESS:

$$s_{ESS}^{t+1} = s_{ESS}^{t} + x_{ESS} \frac{\Delta T}{E_{ESS}} \tag{3.7}$$

Transition between VAC and SoC states are stochastic and depend on the consumed energy during the time interval $P_{cons}(t)$. x_{EV} represents the charging power to the VAC.

$$s_{VAC}^{t+1} = s_{VAC}^{t} + (x_{EV} - P_{cons}(t)) \frac{\Delta T}{E_{VAC}} \tag{3.8}$$

Consumed energy has two uncertainties: behavioural uncertainty i.e. number of driving cars; and consumption uncertainty i.e. energy consumption of driving cars. Behavioural uncertainty is modelled as probability mass functions that represent the probability of number of cars driving at each time interval. Below $p_n(t)$ stands for 'n number of cars driving' case for time interval *t*:

$$p(t) = \begin{vmatrix} p_0(t) \\ p_1(t) \\ p_n(t) \\ p_N(t) \end{vmatrix} \tag{3.9}$$

Energy consumption uncertainty of VAC is neglected by using the average consumption assumption (see Eq. 3.4). The expected number of driving cars are multiplied with the mean unit-time power consumption of one car's away state to estimate the total consumption of the time interval *t*:

$$P_t^{exp} = P_{unit-time} \sum_{n=0}^{N} n p_n(t) \quad 3.10$$

Since SDP needs discretized state variables, s_{ESS}^t and s_{VAC}^t can take finite number of values within the SOC range. This limits the solution space for optimal action but makes the problem an integer problem. As each state combination is structured and solved as an independent optimization problem, decision variable for each will be a vector of PV power output, power from the ESS, power from the grid import and power to the VAC:

$$x = \begin{vmatrix} x_{PV} \\ x_{ESS} \\ x_{Grid} \\ x_{VAC} \end{vmatrix} \quad 3.11$$

PV output depends on the weather; and is thus constrained by the maximum power point of the time interval, Eq. 3.12. VAC charging power term has no restriction about the cars' position but has to follow the chargers' power limitations $P_{evCh,max}^t$ and can take positive values only when the car is present at home, Eq. 3.13. Eq. 3.14 represents the electric power balance that assures meeting the household demand while charging the VAC:

$$x_{PV} \leq P_{pv,max}^t \quad 3.12$$

$$x_{EV} \leq P_{evCh,max}^t \quad 3.13$$

$$P_{dem}^t + x_{VAC} = x_{PV} + x_{ESS} + x_{Grid} \quad 3.14$$

As explained in chapter 2.3.2, stochastic dynamic programming solves an optimization function that has two terms: incurred immediate cost of taken action and future cost of taken decision. As dynamic programs are solved backwards, the calculation starts from the final stages. Zero is assigned as cost V^T of being at any state in final stage. In following equation s stands for a specific ESS SoC-VAC SoC state combination, and S for all possible combinations:

$$V^T = 0 \quad \forall s \in S \quad 3.15$$

Following equation shows how the optimal decision is calculated if the car park EV charging energy management target is minimizing the import from the grid. $\sum_{w=1}^{W} \xi_t{}^w v_{t+1}(s_{t+1})$ is the expected future cost of taken decision; and it depends on the state that is expected to be reached by taking the decision *x* now.

$$\min W_h x_{Grid} + W_d L(s_{VAC}^{t+1} < 0|x) + \sum_{w=1}^{W} \xi_t{}^w v_{t+1}(s_{t+1}) \quad 3.16$$

Eq. 3.16 import is penalized with a factor W_h. An important remark on the nature of this optimization approach is that this model motivates EV charging through the second term in the objective function. To clarify, immediate cost part ($W_h x_{Grid}$) has no influence on the VAC charging. However, it is also possible to reach some physically infeasible end states of VAC such as following inequality:

$$s_{VAC}^{t+1} < 0 \quad 3.17$$

Dropping below 0% SoC is mathematically possible but physically impossible. In reality, this is compensated by charging car at another station. Hypothetical negative SOC is compensated by adding extra cost to the objective function according to the likelihood of reaching negative end state s_{VAC}^{t+1} with the decision *x*. W_d is the penalty factor for this case.

Optimal cost of starting from this initial state equals to the value of the objective function when the optimal decision is taken.

$$V^t(s_{ESS}^t, s_{VAC}^t) = W_h x_{Grid}^* + W_d L(s_{VAC}^{t+1} < 0|x^*) + \sum_{w=1}^{W} \xi_t{}^w v_{t+1}(s_{t+1}) \quad 3.18$$

3.4 Disaggregation of VAC Charging into Plugged Cars

In fact, control action (charging EVs) is constrained with the actual number of cars at the car park. On the other hand, as the cars' position are not represented by any state variable in SDP, optimal policy is calculated by neglecting potential physical infeasibilities. Furthermore, SDP returns an optimal policy for a virtual battery. This policy has to be distributed into the plugged cars at the car park, so that the control actions could be implemented by the battery chargers.

VAC charging power will be shared into the 'home' cars with respect to their charge level. Firstly, depth-of-discharge (DoD) for each car is calculated. Depth-of-discharge (DoD_n) defines the depleted portion of the battery capacity of car *n*:

$$DoD_n = 1 - SoC_n \quad 3.19$$

Next, total DoD of the home cars are calculated by summing DoDs of all electric vehicles.

$$DoD_{Tot} = \sum_n DoD_n \quad 3.20$$

This approach prioritizes charging for car with the lowest SoC i.e. largest DoD. Therefore, VAC power is disaggregated into the home cars proportionally to their DoD levels as shown in Eq. 3.21:

$$P_{ch,n} = x_{VAC}^0 \frac{DoD_n}{DoD_{Tot}} \quad 3.21$$

Our approach for disaggregation VAC charging power into the plugged cars have two constraints. First one is whether the battery capacities of the EVs allow this energy input. Second constraint is about the chargers' capacity. If calculated VAC charging power is too large to be implemented, then the surplus power is utilized according to the local EM target. For example, if VAC charging power is calculated as 27 kW whereas only 22 kW can be charged to the electric vehicles, then 5 kW surplus will be either charged to ESS or in the worst-case PV generation will be curtailed.

4 PROFEV

PROFEV [7] is a framework that combines data from various sources and offers a flexible optimization setting environment for controlling EVs and ESS together. The architecture includes modules for management and signal processing of sensor data (smart meters in S4G), linking of predictive algorithms to deliver inputs to the optimization model, optimization modelling, linking of a solver, an optimization controller and a post-processer module for formatting the results or creating events. The framework offers an API that describes the insertion of new optimization models, allows the registration of data input and output and presents a set of commands to control the start of the framework.

PROFEV will be used in S4G for the optimal control of EVs. Therefore, PROFEV is able of running stochastic dynamic programming as explained in chapter 2.3.2.

5 Installation/Deployment instructions

PROFEV, the tool running the stochastic dynamic programming algorithms, uses a container architecture which simplifies its deployment. Images of the software are uploaded into an open space of Docker Hub. The user has to download the images and make them run into the final platform (Aggregator). Any change of the settings can be achieved through the PROFEV API.
Another possibility for the deployment is by using the open source Linksmart® repository [7]. The code can be cloned from there. In that case, developers can change the code if necessary, then build it and run into the platform.

6 Software dependencies and requirements

This D4.6 prototype has the software dependencies described in Table 1 – Software Dependencies

Table 1 – Software Dependencies

Dependency	License	Role
Docker and docker-compose for Raspbian	Apache License 2.0	Docker is used to facilitate the PROFEV installation.
Raspbian, version "Jessie with Desktop"	GNU General Public License (GPL) 2.0	The operating system of the Raspberry PI used as SMCore

7 API Reference

7.1 PROFEV

As shown in Figure 5, the PROFEV allows different operations to enable the control of the field-related ESS devices and charging station for electric vehicles. Moreover, it allows the operation of the algorithms, and to insert and remove data.

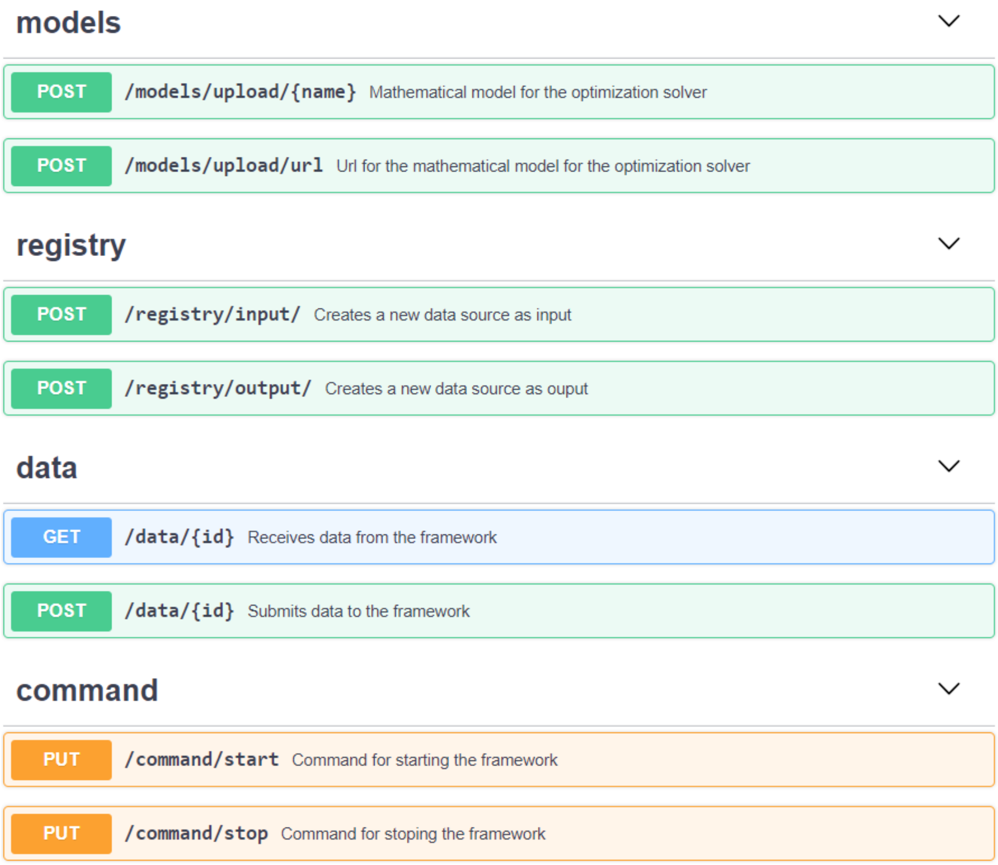

Figure 5: PROFEV API

8 Conclusions

The present deliverable provides all the necessary information regarding D4.6 – "Initial Cooperative EV charging station control algorithms" prototype, developed by the Storage4Grid project.

The first development of the algorithm models a virtual battery of all EVs using the car park. Based on the virtual battery a stochastic optimization problem was modelled and solved using stochastic dynamic programming. This optimization model copes with the driving uncertainty of the vehicles by using Markov chains to represents their behaviour. The implementation of the algorithm in a raspberry is achieved by using the PROFEV framework developed in S4G.

The current algorithm will be extended and updated in order to obtain the best model for the optimal control of EVs in the car park.

Acronyms

Acronym	Explanation
API	Application Program Interface
CP	Charging point
DoD	Depth of discharge
ESS	Energy storage system
EV	Electric vehicle
GESSCon	Grid ESS Controller
GPL	General Public License
LP	Linear problem
MIP	Mixed integer problem
NLP	Non-linear problem
OCPP	Open charge point protocol
PROFESS	Professional Realtime Optimization Framework for Energy Storage Systems
PROFEV	Professional Realtime Optimization Framework for Electric Vehicles
PV	Photovoltaics
SoC	State of charge
S4G	Storage4Grid
SDP	Stochastic dynamic programming
VAC	Virtual aggregated capacity

List of figures

List of tables

References

[1] Merriam-Webster, "Optimization | Definition of optimization by Merriam-Webster," *Merriam-Webster*, 2015. [Online]. Available: https://www.merriam-webster.com/dictionary/optimization. [Accessed: 20-Jun-2018].

[2] J. Keirstead and N. Shah, "Urban Energy Systems: An Integrated Approach," no. June, p. 312, 2012.

[3] P. Morrissey, P. Weldon, and M. O'Mahony, "Future standard and fast charging infrastructure planning: An analysis of electric vehicle charging behaviour," *Energy Policy*, vol. 89, pp. 257–270, 2016.

[4] C.-C. Lin, H. Peng, and J. W. Grizzle, "A stochastic control strategy for hybrid electric vehicles," *Proc. 2004 Am. Control Conf.*, vol. 5, pp. 4710–4715, 2004.

[5] E. B. Iversen, J. K. Moller, J. M. Morales, and H. Madsen, "Inhomogeneous Markov Models for Describing Driving Patterns," *IEEE Trans. Smart Grid*, vol. 8, no. 2, pp. 581–588, 2017.

[6] E. B. Iversen, J. M. Morales, and H. Madsen, "Optimal charging of an electric vehicle using a Markov decision process," *Appl. Energy*, vol. 123, pp. 1–12, Oct. 2014.

[7] Fraunhofer FIT, "PROFEV Repository," 2018. [Online]. Available: https://code.linksmart.eu/projects/SPF/repos/optimization-framework/browse. [Accessed: 01-Aug-2018].