Žilinská Univerzita v Žiline
Fakulta riadenia a informatiky

Diplomová práca

Študijný odbor: **Informačné systémy**
Zameranie: **Spracovanie dát**

Bc. Miroslav Gardlo

Matematické modelovanie a simulácia nabíjania elektrických vozidiel

Vedúci: **doc. Ing. Ľuboš Buzna, PhD.**

Ministerské číslo práce: 28360420182044
Registračné číslo: 44/2017
Žilina 2018

Poďakovanie

Chcel by som sa poďakovať vedúcemu práce doc. Ing. Ľuboš Buznovi, PhD. za veľké množstvo cenných rád a odovzdaných informácií pri vypracovávaní diplomovej práce.

Prehlásenie

Prehlasujem, že som túto prácu napísal samostatne a že som uviedol všetky použité pramene a literatúru, z ktorých som čerpal.

V Žiline, dňa 21.4.2018 Miroslav Gardlo

Abstrakt

Miroslav Gardlo: *Matematické modelovanie a simulácia nabíjania elektrických vozidiel* [Diplomová práca]
Žilinská Univerzita v Žiline, Fakulta riadenia a informatiky, Katedra matematických metód a operačnej analýzy.
Vedúci: doc. Ing. Ľuboš Buzna, PhD.
Stupeň odbornej kvalifikácie: Inžinier.
Žilina: FRI ŽU v Žiline, 2018. 71 strán.

Rastúci počet elektrických vozidiel využívajúcich elektrickú sieť pre nabíjanie môže v budúcnosti spôsobovať zvýšenú záťaž siete, prípadne jej preťaženie. Zvýšené nároky kladené na elektrickú sieť si môžu v budúcnosti vynútiť návrh a implementáciu protokolu, ktorý bude riadiť tok elektrickej energie v elektrickej sieti. Cieľom diplomovej práce je navrhnúť, implementovať a analyzovať používateľské stratégie pre nabíjanie elektrických vozidiel, ktoré dokážu decentralizovaným spôsobom ovplyvňovať množstvo prideľovanej elektrickej energie počas nabíjania, a tým potenciálne predchádzať preťažovaniu siete. Používateľská stratégia pre nabíjanie elektrických vozidiel je softvérový agent reprezentujúci elektrické vozidlo, respektíve používateľa v elektrickej sieti počas nabíjania a na základe stavových informácií o vozidle ako napríklad ostávajúci čas pre nabíjanie, aktuálny stav batérie, ale aj na základe ďalších ukazovateľov, dokáže samostatne modifikovať parameter nazvaný *ochota používateľa platiť*, pomocou ktorého stratégie ovplyvňujú množstvo prideľovanej elektrickej energie.

Kľúčové slová: nabíjanie elektrických vozidiel, konvexná optimalizácia, matematické modelovanie, simulácia

Abstract

Miroslav Gardlo: *Mathematical modelling and simulation of electric vehicle charging* [Master's thesis]
University of Žilina, Faculty of Management Science and Informatics, Department of mathematical methods and operations research.
Tutor: doc. Ing. Ľuboš Buzna, PhD.
Qualification level: Engineer.
Žilina: FRI ŽU in Žilina, 2018. 71 pages.

Increasing number of electric vehicles plugging into electric network may eventually lead to increased load on the electric network, or to congestion. In the future, increased load may lead to a need of designing and implementing a protocol controlling flow of electric power in the network. The main goal of this thesis is to design, implement and analyze electric vehicle charging strategies (also called user strategies), that are capable of influencing assignment of electric power in decentralized fashion and potentially prevent network congestion. Electric vehicle charging strategy is a software agent that represents a specific electric vehicle (or user) in the electric network during the charging process and based on state information of the vehicle including, but not limited to current battery state or time left for charging, modify a parameter called *willingness to pay*, which is used to influence assignment of electric power.

Keywords: electric vehicle charging, convex optimization, mathematical modelling, simulation

Contents

List of Figures

List of Tables

Notation

- $\mathcal{V}$ set of nodes in the electric network,
- $\mathcal{V}^+$ set of nodes in the electric network except for the node that is source of electric power,
- r_{ij} resistance of edge $e_{ij} \in \mathcal{E}$,
- x_{ij} reactance of edge $e_{ij} \in \mathcal{E}$,
- z_{ij} impedance of edge $e_{ij} \in \mathcal{E}$,
- y_{ij} admittance of edge $e_{ij} \in \mathcal{E}$,
- g_{ij} conductance of edge $e_{ij} \in \mathcal{E}$,
- b_{ij} susceptance of edge $e_{ij} \in \mathcal{E}$,
- $\mathcal{G}(\mathcal{V}, \mathcal{E})$ directed graph representing the model of the distribution network,
- t simulation time
- t_c^l length of time for which vehicle l is present in the network (charging time),
- $w_l(t), w_l(t_c^l)$ willingness to pay of the vehicle l at specific time t, t_c^l,
- w_{min}^l minimum willingness to pay of vehicle l,
- w_{max}^l maximum willingness to pay of vehicle l,
- w_{min} minimal willingness to pay (not vehicle-specific),
- $\mathcal{N}(t)$ the set of vehicles with $w_l(t) > 0$,
- $P(t)$ vector of power allocations,
- $P_l(t)$ power allocated to vehicle l,
- $\mathcal{P}(t)$ the set of all feasible power allocations $P(t)$,
- $\overline{P}_i(t)$ power allocated to network node $i \in \mathcal{V}^+$,
- $\overline{w}_i(t)$ willingness to pay associated with node $i \in \mathcal{V}^+$,
- $\mathcal{E}$ set of edges (representing lines of the electric network),
- $e_{ij} \in \mathcal{E}$ edge from node i to node j,

- $\mathcal{E}^+$ set of edges $e_{ij} \in \mathcal{E}$, where $i > j$,
- $v_i(t)$ voltage on the node $i \in \mathcal{V}^+$ (complex number),
- $\underline{v}_i$ lower bound of the voltage amplitude at network node $i \in \mathcal{V}$,
- $\bar{v}_i$ upper bound of the voltage amplitude at network node $i \in \mathcal{V}$,
- $V_{ii}(t), V_{ij}(t)$ substitute voltage variables,
- $i_{ij}(t)$ electric current that flows from node $i \in \mathcal{V}$ to node $j \in \mathcal{V}$,
- $p_{ij}(t)$ real part of electric power that flows from node $i \in \mathcal{V}$ to $j \in \mathcal{V}$,
- $q_{ij}(t)$ reactive part of electric power that flows from node $i \in \mathcal{V}$ to node $j \in \mathcal{V}$,
- $C_i(t)$ set of electric vehicles that are connected to node $i \in \mathcal{V}^+$ at time t, with non-zero willingness to pay,
- $\mathcal{M}^+(t) = \{i \in \mathcal{V}^+ \mid C_i(t) \neq \emptyset\}$ a set of occupied network nodes
- $Re\{x\}$ real part of a complex number x,
- $Im\{x\}$ imaginary part of a complex number x,
- T^l_{max} maximum time t^l_c for which vehicle l can stay in the electric network,
- T_{max} maximum time for which vehicles can stay in the electric network (not vehicle-specific),
- W^l_{max} maximum budget that a vehicle is allowed to spend on charging,
- W_{max} maximum budget that can be spent on charging (not vehicle-specific),
- $W^l_r(t), W^l_r(t^l_c)$ remaining budget of vehicle l at time t, t^l_c,
- B^l_{max} maximum battery capacity of vehicle l,
- B_{max} maximum battery capacity (not vehicle-specific),
- $B^l(t), B^l(t^l_c)$ battery state of vehicle l at time t, t^l_c,
- $B^l(0)$ battery state of vehicle l at time of arrival to the network ($t^l_c = 0$),
- d^l parameter associated with vehicle l using AUT *strategy*,
- d parameter associated with AUT *strategy* (not vehicle-specific),

1. Introduction

1.1 Motivation

Electric vehicles are nowadays more and more popular worldwide. With rising pollution, electric vehicles may be one of the possible solutions for slowing down global warming, which is why many countries intend to replace vehicles with combustion engines by electric vehicles as part of new laws.

Currently, there is no resource sharing protocol implemented in electric networks, which would control flow of electric power, however, with rapid raise in numbers of electric vehicles plugging into electric network, modern control of electric power flow will eventually be required. Aim of this thesis is to present a decentralized charging protocol, that would allow electric vehicles influence assigned electric power during charging process.

Since state of electric network changes in time, owners of electric vehicles would have to supervise charging of their vehicle in the real time and make real time decisions in matter of seconds, which would definitely be impossible. To overcome this situation, electric vehicles will be represented by a software agent called *electric vehicle charging strategy (or user strategy)*. *Electric vehicle charging strategy* is an algorithm that analyzes current situation of the user that it represents and takes action in real time, with intention to influence assigned electric power. While making a decision, an *electric vehicle charging strategy* can take multiple variables into consideration including, but not limited to current battery state or time left for charging. Strategies influence assigned electric power using a non-negative parameter called *willingness to pay*, when electric vehicle charging strategy increases this parameter, it expects to gain more electric energy in the short future and vice versa. Charging strategies can be thought of as algorithms updating willingness to pay parameter with intention to influence electric power assignment. Detailed information about charging strategies is in the section (3).

Replacement of human decision making by software agents is notable in stock markets. According to article [2], only 10 percent of stocks are now traded daily by individuals making active choices about what to buy or sell. The article also states that the issue for markets is how technology and automated, program-driven trading can and is creating powerful anomalies, which in some cases can only be explained by software and passive trading that lead to blind selling and blind buying.

Charging strategies representing electric vehicles in the network are similar to the stock trading software, they replace human decision making by an algorithm, the purpose of which is to influence intake of electric power in order to meet the demands of user that the strategy represents. Since replacement of decision making by software could lead to unexpected behaviour, individual and collective behaviour of the charging strategies will be studied in this thesis.

1.2 Convex optimization

Electric power assigned to electric vehicles will be determined by solving a convex optimization problem. In this section, information from [3] will be summarized to provide basic information about the convex optimization terms used in this thesis.

1.2.1 Important definitions from convex optimization

Definition 1. *A symmetric matrix A is called positive semidefinite if for all $x, x^T Ax \geq 0$. We denote this as $A \succeq 0$. Notation S^n_+ is used to denote the set of positive semidefinite matrices.*

Definition 2. *A set C is convex if the line segment between any two points in C lies in C, i.e., if for any $x_1, x_2 \in C$ and any θ with $0 \leq \theta \leq 1$, we have $\theta x_1 + (1-\theta)x_2 \in C$.*

Definition 3. *A function $f : R_n \rightarrow R$ is convex if $domf$ is a convex set and if for all x, $y \in domf$ and θ with $0 \leq \theta \leq 1$, we have $f(\theta x + (1-\theta)y) \leq \theta f(x) + (1-\theta)f(y)$. Geometrically, this inequality means that the line segment between $(x, f(x))$ and $(y, f(y))$, which is the chord from x to y, lies above the graph of f.*

Definition 4. *The α-sublevel set of a function $f : R_n \rightarrow R$ is defined as $C_\alpha = \{x \in domf \mid f(x) \leq \alpha\}$. Sublevel sets of a convex function are convex, for any value of α.*

Definition 5. *A cone $K \subseteq R^n$ is called a proper cone if it satisfies the following:*

- *K is convex.*
- *K is closed.*
- *K is solid, which means it has nonempty interior.*
- *K is pointed, which means that it contains no line.*

A proper cone K can be used to define a generalized inequality, which is a partial ordering on R^n that has many of the properties of the standard ordering on R. We associate with the proper cone K the partial ordering on R^n defined by $x \preceq_K y \Leftrightarrow y - x \in K$. When $K = R_+$, the partial ordering $\preceq_K K$ is the usual ordering $\leq$ on R.

1.2.2 Classification of convex optimization problems

A convex optimization problem is a problem of form:

$$\begin{aligned} \text{minimize} \quad & f_0(x) && && \text{(1.1a)} \\ \text{subject to} \quad & f_i(x) \leq 0 && i = 1, \dots, m, && \text{(1.1b)} \\ & a_i^T x = b_i && i = 1, \dots, p, && \text{(1.1c)} \end{aligned}$$

where $f_0, \dots, f_m$ are convex functions. The feasible set of a convex optimization problem is convex, since it is the intersection of the domain of the problem $\bigcap_{i=1}^{m} f_i$, which is a convex set, with m convex sublevel sets $\{x \mid f_i(x) \leq 0\}$ and p hyperplanes $\{x \mid a_i^T x = b_i\}$.

Categories of convex optimization problems will be summarized below in increasing order of computational complexity.

Linear optimization problems

When the objective and constraint functions are all affine, the problem is called a linear program. Linear program has the form:

$$\begin{aligned} \text{minimize} \quad & c^T x + d && \text{(1.2a)} \\ \text{subject to} \quad & Gx \preceq h, && \text{(1.2b)} \\ & Ax = b, && \text{(1.2c)} \end{aligned}$$

where $G \in R^{m \times n}$ and $A \in R^{p \times n}$. Constant d can be omitted in the objective function, since it does not affect the optimal (or feasible) set. Affine objective $c^T x + d$ can be maximized, by minimizing $-c^T x - d$, therefore a maximization problem with affine objective and constraint functions is an LP.

Linear-fractional programming

Linear-fractional program takes the form:

$$\begin{aligned} \text{minimize} \quad & f_0(x) && \text{(1.3a)} \\ \text{subject to} \quad & Gx \preceq h, && \text{(1.3b)} \\ & Ax = b, && \text{(1.3c)} \end{aligned}$$

where the objective function is given by $f_0(x) = \frac{c^T x + d}{e^T x + f}$, $dom f_0 = \{x \mid e^T x + f > 0\}$.

Quadratic optimization problems

The convex optimization problem is called a quadratic program (QP) if the objective function is (convex) quadratic, and the constraint functions are affine. A quadratic program can be expressed in the form:

$$\begin{aligned} \text{minimize} \quad & \left(\frac{1}{2}\right) x^T P x + q^T x + r && (1.4a)\\ \text{subject to} \quad & Gx \preceq h, && (1.4b)\\ & Ax = b, && (1.4c) \end{aligned}$$

where $P \in S^n_+, G \in R^{m\times n}, A \in R^{p\times n}$.

Quadratically constrained quadratic programming

Quadratic program with quadratic inequality constraint functions is called quadratically constrained quadratic program (QCQP) and takes the form

$$\begin{aligned} \text{minimize} \quad & \left(\frac{1}{2}\right) x^T P_0 x + q_0^T x + r_0 && (1.5a)\\ \text{subject to} \quad & \left(\frac{1}{2}\right) x^T P_i x + q_i^T x + r_i \leq 0 \qquad i = 1, \dots m, && (1.5b)\\ & Ax = b, && (1.5c) \end{aligned}$$

where $P_i \in S^n_+, i = 0, \dots, m$. In QCQP, we minimize a convex quadratic function over a feasible region that is the intersection of ellipsoids.

Second-order cone programming

Second-order cone programs take the form

$$\begin{aligned} \text{minimize} \quad & f^T x && (1.6a)\\ \text{subject to} \quad & \|A_i x + b_i\|_2 \leq c_i^T x + d_i \quad i = 1, \dots, m, && (1.6b)\\ & Fx = g, && (1.6c) \end{aligned}$$

where $x \in R^n$ is the optimization variable, $A_i \in R^{n_i \times n}$, and $F \in R^{p\times n}$. Constraint of the form $\|Ax + b\|_2 \leq c^T x + d$, where $A \in R^{k\times n}$, is called a second-order cone constraint. Notation $\|v\|_2$ represents an euclidian norm of vector v.

Semidefinite programming

Semidefinite programs take the form

$$\begin{aligned} \text{minimize} \quad & c^T x && (1.7a)\\ \text{subject to} \quad & x_1 F_1 + \ldots + x_n F_n + G \preceq 0, && (1.7b)\\ & Ax = b, && (1.7c) \end{aligned}$$

where $G, F_1, ..., F_n \in S^k$, and $A \in R^{p\times n}$. Set S^k denotes a set of symmetric matrices.

1.3 Proportional fairness

To allocate electric power to vehicles, an objective function called *weighted proportional fairness* will be used. According to [4], proportional fairness is a well-known fairness notion that has been motivated by communication network applications. Proportional fairness takes the form

$$\sum_{d \in D} \log(x_d), \tag{1.8}$$

where D represents a set of demands $D = \{1,2,\ldots,n\}$. It can be shown that optimal solution x^{pf} satisfies

$$\sum_{d \in D} \frac{x_d - x_d^{pf}}{x_d^{pf}} \leq 0, \tag{1.9}$$

where $x_d, d \in D$ is any feasible solution. Solution x^{pf} is called proportionally fair when the aggregate of proportional changes with respect to any other feasible solution is zero or negative. Concept of proportional fairness was extended by adding positive weights to each term in the objective function, this is called a *weighted proportional fairness*. Logarithmic function in the objective ensures that there is no zero allocation to any of the demands, since $\log(0) = -\infty$. Proportional fairness is included in a family of performance functions called the $\alpha - fair$ utility functions

$$w_\alpha(x_d) = \frac{x_d^{1-\alpha}}{1-\alpha} \qquad \text{for } \alpha \geq 0, \alpha \neq 1, \text{and } d \in D,$$
$$w_\alpha(x_d) = \log(x_d) \qquad \text{for } \alpha = 1, \text{and } d \in D,$$

where the objective is to maximize the sum of the utility functions over all demands. This utility function captures a whole family of criteria by selecting different values for α. When $\alpha = 1$, the optimal solution is proportionally fair. Proportional fairness has been shown to be effective for rate control in communication networks. When $\alpha = 0$, the utility function is linear. When $\alpha \to \infty$, the function approximates the lexicographic maximin objective [4].

1.3.1 Convexity of proportional fairness

Since our intention is to use weighted proportional fairness as an objective function of a convex optimization problem, it has to be convex. Weighted proportional fairness takes the form $\sum_{d \in D} w_d \log(x_d)$, where $w_d \geq 0$ represents a positive weight associated with decision variable x_d modelling a demand. Logarithm used in the objective is a strictly concave function, because its second derivative is always negative for any point x in its domain. According to [3], function f strictly concave if $-f$ is strictly convex, which means that $-\log(x)$ is a strictly convex function. In this thesis we want to maximize the objective function $\sum_{d \in D} w_d \log(x_d)$, but since formulation of convex problem requires a minimization objective, we have to minimize $\sum_{d \in D} - w_d \log(x_d)$, which has to be a convex function. By substituting $f(x_d)$ for each $-log(x_d)$, objective takes the form $\sum_{d \in D} w_d f(x_d)$, where $f(x_d)$ is a strictly convex function. According to [3], non-negative weighted sum of convex functions is a convex function, which is the case for the substituted objective, therefore maximization of weighted proportional fairness can be used in a convex optimization problem, it only has to be transformed into minimization form in order to conform to the standard form of convex optimization problems.

1.4 Simulation

Behaviour of electric vehicle charging strategies will be analyzed using simulation, therefore basic information about simulation from [5] and [6] will be presented. According to [6] *Simulation is a research technique, which replaces a dynamic system that we intend to study by a simulation model, that is experimented with in order to obtain information about the original dynamic system.* Simulation is particularly useful when dynamic system that we intend to study does not exist at all, or exists, but cannot be experimented with directly because of limitations such as time, safety or even from financial and ethical reasons. Smart charging scheme where electric vehicles are represented by software agents controlling flow of real power does not currently exist, therefore a simulation model will be created in order to study this type of system [5].

In order to create a simulation model needed for studying electric vehicle charging strategies, activities present in the system have to be identified. *Simulation activity is a basic action unit of simulation that represents an activity in the system that we simulate. Each activity has a specific length. Activities can change state of the system. Simulation program executes individual simulation activities in the same order as the corresponding activities occur in the real system* [5].

According to [5] [6], activities can be divided into two categories:

Discrete activities: this type of activity does not model entire duration of the corresponding activity in the modelled system, therefore discrete activities can change the system state solely at the end of activity. Discrete activities are characterized by an occurrence time t.

Continuous activity: this type of activity is used to model real activities with their duration. Continuous activities can be for example used, when modelling an activity which depends on its previous state. Continuous activities are characterized by an interval of real numbers $\langle t_1, t_2 \rangle$.

Based on types of activities present in the simulation model, simulation could be categorized into three areas [5], [6]:

Discrete simulation - simulation contains only discrete activities.

Continuous simulation - simulation contains only continuous activities.

Combined simulation - simulation contains both discrete and continuous activities.

In this thesis a combined simulation will be used. Discrete activities will be processed using *Event planning method*. In *event planning method*, events are planned in advance and are stored in a calendar (time line). Calendar is ordered non-decreasingly based on the occurrence time of the planned events. Simulation processes events in this order [6].

Method based on *Activity scanning* will be used for processing continuous activities, but this method is also suitable for discrete activities. *Activity scanning method* gradually increases simulation time by a time step Δt, after each increase of simulation time, all running activities are evaluated to check whether the activity should end (discrete activity) or there was an important change in the observed dynamic attributes resulting in execution of specific action (continuous activity) [6] . Detailed information about the created simulation model and its implementation is in section (4.2).

1.5 Goals of the thesis

Main goal of this thesis is to simulate and analyze a non existing dynamic system, in which *electric vehicle charging strategies* or (*user strategies)* representing a real world electric vehicles in the electric network influence the amount of assigned electric power in decentralized fashion. Each electric vehicle charging strategy sets and updates a non-negative real parameter called *willingness to pay*, purpose of which is to influence intake of electric power during charging process. This is the only value that user strategies communicate to the electric network. In this thesis, multiple electric vehicle charging strategies will be designed, implemented and experimented with, each of which will approach updating *willingness to pay* parameter from different perspective and with different motivation.

Willingness to pay of each vehicle will be used in an objective function called *weighted proportional fairness*. Allocation of electric power will be done by solving a convex optimization problem, which means that a solver capable of handling convex optimization problems has to be found and the optimization problem has to be implemented in solver-specific form. Another goal of this thesis is to implement a simulation program, which would allow us to study implemented charging strategies and gain information about behaviour of both, individual vehicles represented by charging strategies in various situations and group of vehicles. Studying charging strategies can reveal information about the possible performance of the non-existing dynamic system, in which software agents representing electric vehicles control and influence assignment of electric power. Summary of the goals of this thesis is below:

1. Implementation of mathematical model of optimal power flow.
   (a) Search for a solver capable of solving convex optimization problems.
   (b) Implementation of mathematical model in solver-specific form.
2. Design and implementation of electric vehicle charging strategies.
3. Design and implementation of simulation tool supporting various experiments with implemented electric vehicle charging strategies.
4. Simulation experiments with created simulation tool.
5. Evaluation of performed simulation experiments.

2. Mathematical model

In this section, mathematical model that forms the core of the simulation model will be presented. Optimization model is used to determine the vector of electric power $P(t) = (P_l(t) : l \in \mathcal{N}(t))$ that is allocated to electric vehicles, where $\mathcal{N}(t)$ represents a set of electric vehicles connected to electric network at time t, that are willing to pay for electric energy ($w_l(t) > 0$). The feasible set of power allocations is denoted as $\mathcal{P}(t)$. Symbolic representation of the model takes the form

$$\underset{P(t)}{\text{maximize}} \quad \sum_{l \in \mathcal{N}(t)} w_l(t) \log(P_l(t)) \tag{2.1a}$$

$$\text{subject to} \quad P(t) \in \mathcal{P}(t), \tag{2.1b}$$

where $w_l(t)$ are nonnegative weights called willingness to pay. Variable $P_l(t)$ is only created for vehicles with $w_l(t) > 0$, vehicles with $w_l(t) = 0$ are not willing to pay at time t, therefore $P_l(t) = 0$.

2.1 The set of feasible solutions

The mathematical description of the set of feasible solutions $\mathcal{P}(t)$, is derived from the optimal power flow (OPF) problem [7]. The electric network is modelled by a directed tree graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is a set of nodes and $\mathcal{E}$ is set of edges. Symbol $\mathcal{V}^+$ denotes the set of all nodes excluding the source of electric energy. Voltage drops are significant in distribution networks, and to large extent determine the network capacity, which is why a model of power flow specific to distribution networks [8] is used. The complex variable representing the voltage level at network node $i \in \mathcal{V}$ is denoted as $v_i(t)$. The set of constraints

$$\underline{v}_i(t) \leq |v_i(t)| \leq \bar{v}_i(t) \qquad i \in \mathcal{V}, \tag{2.2}$$

ensures that voltage amplitude at all network nodes is maintained within lover $\underline{v}_i$ and upper $\bar{v}_i$ bounds. In the objective function 2.1a, variables representing electric power allocated to individual vehicles are used, therefore to be able to apply constraints 2.2, constraints linking voltages and electric power by considering line properties of the distribution network have to be included. Edge $e_{ij} \in \mathcal{E}$ that connects nodes $i \in \mathcal{V}$ and $j \in \mathcal{V}$ is characterized by the impedance $z_{ij} = r_{ij} + ix_{ij}$, where r_{ij} is edge resistance and x_{ij} is edge reactance. In mathematical expressions, it is more practical to use admittance, which is the reciprocal of the impedance

$$y_{ij} = \frac{1}{r_{ij} + ix_{ij}} = \frac{r_{ij}}{r_{ij}^2 + x_{ij}^2} - i\frac{x_{ij}}{r_{ij}^2 + x_{ij}^2} = g_{ij} - ib_{ij}. \tag{2.3}$$

The real part is called conductance and the imaginary part susceptance. By Ohm's law, the electric current through the line connecting nodes i and j is given by

$$i_{ij}(t) = (v_i(t) - v_j(t))y_{ij}. \tag{2.4}$$

The power through the line $e_{ij} \in \mathcal{E}$ is obtained by multiplying the conjugate of the current by the voltage rise between the ground and the node i or, alternatively, the voltage drop between node i and the ground on the other side of the load at node j. This yields

$$p_{ij}(t) + iq_{ij}(t) = v_i(t)(v_i^*(t) - v_j^*(t))y_{ij}^*, \tag{2.5}$$

where superscript * denotes the complex conjugate. Note that this is the power that leaves the node i, which is different than the power that reaches node j because of the power losses dissipated on the power line. The power departing node j is then

$$p_{ji}(t) + iq_{ji}(t) = |v_i(t) - v_j(t)|^2 y_{ij}^* - p_{ij}(t) - iq_{ij}(t) = v_j(t)(v_j^*(t) - v_i^*(t))y_{ij}^*. \tag{2.6}$$

$$\underset{v(t),P(t),p(t),q(t)}{\text{maximize}} \quad \sum_{l \in \mathcal{N}(t))} w_l(t) \log(P_l(t)) \tag{2.7a}$$

subject to

$$p_{ij}(t) + iq_{ij}(t) = v_i(t)(v_i^*(t) - v_j^*(t))y_{ij}^* \qquad e_{ij} \in \mathcal{E} \tag{2.7b}$$

$$\sum_{j|e_{ij} \in \mathcal{E}} p_{ij}(t) = \sum_{l \in C_i(t)} P_l(t) \qquad i \in \mathcal{V}^+ \tag{2.7c}$$

$$\sum_{j|e_{ij} \in \mathcal{E}} q_{ij}(t) = 0 \qquad i \in \mathcal{V}^+ \tag{2.7d}$$

$$\underline{v_i} \leq |v_i(t)| \leq \overline{v_i} \qquad i \in \mathcal{V}. \tag{2.7e}$$

Constraints (2.7c) and (2.7d) ensure validity of the Kirchhoff's conservation law for real and reactive power flows, while each electric vehicle consumes only real power $P_l(t)$ and no reactive power. The set of electric vehicles that are connected to node $i \in \mathcal{V}^+$ at time t is denoted as $C_i(t)$. Equality constraint (2.7b) contains a quadratic term, thus the model (2.7a)-(2.7e) is nonconvex in general, which implies that the problem is not directly solvable by polynomial time methods. To overcome this limitation, we follow [9, 10] by substituting voltage variables in (2.7c) and (2.7d) by introducing matrix

$$V(e_{ij},t) = \begin{pmatrix} v_i(t)v_i^*(t) & v_i(t)v_j^*(t) \\ v_j(t)v_i^*(t) & v_j(t)v_j^*(t) \end{pmatrix} = \begin{pmatrix} V_{ii}(t) & V_{ij}(t) \\ V_{ji}(t) & V_{jj}(t) \end{pmatrix} \tag{2.8}$$

for each edge e_{ij} .

The matrices $V(e_{ij},t)$ are rank one and positive semidefinite because they are of the form $v(e_{ij},t)v^*(e_{ij},t)^{\mathrm{T}}$, with $v(e_{ij},t_s) = (v_i(t), v_j(t))^T$. Based on formula (2.3), the variables p_{ij} and q_{ij}, can be replaced by separating the real and imaginary part of the term obtained by multiplying out brackets in condition (2.7c). Substituting $V_{ij}(t)$ for each $v_i(t)v_j(t)^*$ and plugging (2.7b) into (2.7c) results in equivalent representation of the optimization model:

$$\underset{V(t),P(t),p(t),q(t)}{\text{maximize}} \quad \sum_{l\in\mathcal{N}(t)} w_l(t)\log(P_l(t)) \tag{2.9a}$$

subject to

$$\sum_{j|e_{ij}\in\mathcal{E}} (g_{ij}(V_{ii}(t) - Re\{V_{ij}(t)\}) + b_{ij}Im\{V_{ij}(t)\}) = \sum_{l\in C_i(t)} P_l(t) \qquad i \in \mathcal{V}^+ \tag{2.9b}$$

$$\sum_{j|e_{ij}\in\mathcal{E}} (b_{ij}(V_{ii}(t) - Re\{V_{ij}(t)\}) - g_{ij}Im\{V_{ij}(t)\}) = 0 \qquad i \in \mathcal{V}^+ \tag{2.9c}$$

$$\underline{v_i}^2 \le V_{ii}(t) \le \overline{v_i}^2 \qquad i \in \mathcal{V} \tag{2.9d}$$

$$V(e_{ij},t) \succeq 0 \qquad e_{ij} \in \mathcal{E} \tag{2.9e}$$

$$\text{rank}(V(e_{ij},t)) = 1 \qquad e_{ij} \in \mathcal{E}. \tag{2.9f}$$

Now, the only set of nonconvex constraints is the set (2.9f). SDP relaxation is done by removing these nonconvex constraints. Models (2.9a)-(2.9f) and (2.7a)-(2.9e) have the same optimal solution, if the distribution network is radial [9, 10]. Thus, the optimal solution of the problem (2.9a)-(2.9f) satisfies $\text{rank}(V(e_{ij},t)) = 1$ for all edges, which can be validated numerically.

Problems from SDP category can be challenging to solve at very large scales. SDP constrained problem (2.9a)-(2.9f) can be further relaxed to SOCP constrained problem [7]. SOCP problems are a special class of SDP problems that can be solved by more efficient SOCP solvers [3]. SOCP relaxation is achieved by replacing SDP constraints $V(e_{ij},t) \succeq 0$ by constraints:

$$V_{ij}(t)V_{ij}^*(t) \le V_{ii}(t)V_{jj}(t) \qquad e_{ij} \in \mathcal{E} \tag{2.10}$$

$$V_{ii}(t) \ge 0 \qquad i \in \mathcal{V}^+. \tag{2.11}$$

Constraints (2.10) can be rewritten as:

$$\left\| \begin{pmatrix} 2Re\{V_{ij}(t)\} \\ 2Im\{V_{ij}(t)\} \\ V_{ii}(t) - V_{jj}(t) \end{pmatrix} \right\|_2 \le V_{ii}(t) + V_{jj}(t) \qquad e_{ij} \in \mathcal{E}, \tag{2.12}$$

where $Re\{V_{ij}(t)\}$ is real and $Im\{V_{ij}(t)\}$ is imaginary part of complex voltage $V_{ij}(t)$. Constraints (2.12) are identical for $e_{ij}, e_{ji} \in \mathcal{E}$, therefore it is sufficient to introduce these constraints only for edges $e_{ij} \in \mathcal{E} \mid i > j$. Set of edges $e_{ij} \in \mathcal{E} \mid i > j$ will be denoted as $\mathcal{E}^+$. Validity of constraints (2.11) is ensured by constraints (2.9e). Consequently, SOCP relaxation of SDP constrained problem (2.9a)-(2.9f) yields:

$$\underset{V(t),P(t)}{\text{maximize}} \quad \sum_{l\in\mathcal{N}(t)} w_l(t)\log(P_l(t)) \tag{2.13a}$$

subject to

$$\sum_{j|e_{ij}\in\mathcal{E}} (g_{ij}(V_{ii}(t) - Re\{V_{ij}(t)\}) + b_{ij}Im\{V_{ij}(t)\}) = \sum_{l\in C_i(t)} P_l(t) \qquad i \in \mathcal{V}^+ \tag{2.13b}$$

$$\sum_{j|e_{ij}\in\mathcal{E}} (b_{ij}(V_{ii}(t) - Re\{V_{ij}(t)\}) - g_{ij}Im\{V_{ij}(t)\}) = 0 \qquad i \in \mathcal{V}^+ \tag{2.13c}$$

$$\underline{v_i}^2 \leq V_{ii}(t) \leq \overline{v_i}^2 \qquad i \in \mathcal{V} \tag{2.13d}$$

$$\left\| \begin{pmatrix} 2Re\{V_{ij}(t)\} \\ 2Im\{V_{ij}(t)\} \\ V_{ii}(t) - V_{jj}(t) \end{pmatrix} \right\|_2 \leq V_{ii}(t) + V_{jj}(t) \qquad e_{ij} \in \mathcal{E}^+. \tag{2.13e}$$

Constraint (2.13d), will take the form

$$V^2_{nominal}(1-\alpha)^2 \leq V_{ii} \leq V^2_{nominal}(1+\alpha)^2 \quad i \in \mathcal{V}. \tag{2.14}$$

Purpose of the constraint (2.14) is to guarantee network stability, the nodal voltages can only vary by the fraction α from the nominal voltage [1].

2.2 Aggregation of vehicles into "supervehicles"

Variables $P_l(t)$ associated with vehicles connected to the same node $i \in \mathcal{V}^+$ can be represented by the aggregated nodal variable

$$\overline{P}_i(t) = \begin{cases} \sum\limits_{l\in C_i(t)} P_l(t), & \text{if } i \in \mathcal{M}^+(t) \\ 0, & \text{otherwise} \end{cases} \quad i \in \mathcal{V}^+, \qquad \mathcal{M}^+(t) = \{i \in \mathcal{V}^+ \mid C_i(t) \neq \emptyset\}. \tag{2.15}$$

Then, we can formulate the following optimization problem

$$\underset{V(t),\overline{P}(t)}{\text{maximize}} \quad \sum_{i\in\mathcal{M}^+(t)} \overline{w}_i(t)\log(\overline{P}_i(t)) \tag{2.16a}$$

subject to

$$\sum_{j|e_{ij}\in\mathcal{E}} (g_{ij}(V_{ii}(t) - Re\{V_{ij}(t)\}) + b_{ij}Im\{V_{ij}(t)\}) = \overline{P}_i(t) \qquad i \in \mathcal{V}^+ \tag{2.16b}$$

$$(2.13c) - (2.13e), \tag{2.16c}$$

where $\overline{w}_i(t) = \sum_{j \in C_i(t)} w_j(t)$, and

The following corollaries explore connections between problems (2.13a)-(2.13e) and (2.16a)-(2.16c).

Corollary 2.2.1. *For problems (2.13a)-(2.13e) and (2.16a)-(2.16c) the following relation holds:*

$$P_l(t) = \frac{w_l(t)\overline{P}_i(t)}{\sum_{j \in C_i(t)} w_j(t)} \qquad l \in C_i(t), i \in \mathcal{M}^+(t). \tag{2.17}$$

Proof. This can be shown by minimizing the objective (2.13a) over some variables. Such minimization can be done separately for each network node, this will be shown for an arbitrary node $i \in \mathcal{M}^+(t)$. Node i contributes to the objective function by the expression $\sum_{l \in C_i(t)} w_l(t) \log(P_l(t))$. By using Eq. (2.15), for a selected element $\bar{l} \in C_i(t)$, we express the value

$$P_{\bar{l}}(t) = \overline{P}_i(t) - \sum_{l \in C_i(t) - \{\bar{l}\}} P_l(t), \tag{2.18}$$

and calculate

$$\sup_{P_l(t) | l \in C_i(t) - \{\bar{l}\}} \left\{ \sum_{l \in C_i(t)} w_l(t) \log(P_l(t)) \;\middle|\; P_{\bar{l}(t)} = \overline{P}_i(t) - \sum_{l \in C_i(t) - \{\bar{l}\}} P_l(t) \right\}. \tag{2.19}$$

Solving (2.19) analytically, requires calculation of partial derivatives with respect to variables $P_l(t)$ for $l \in C_i(t) - \{\bar{l}\}$ and making them equal to zero. These steps yield

$$\frac{P_l(t)}{w_l(t)} = \frac{\overline{P}_i(t) - \sum_{j \in C_i(t) - \{\bar{l}\}} P_j(t)}{w_{\bar{l}}(t)} \qquad l \in C_i(t) - \{\bar{l}\}. \tag{2.20}$$

Right hand side of equation (2.20) is identical for each $l \in C_i(t) - \{\bar{l}\}$, therefore all values $\frac{P_l(t)}{w_l(t)}$ are identical as well. Based on this, all values $P_j(t)$ for $j \in C_i(t) - \{\bar{l}\}$ can be expressed in Eq. (2.20) by using $P_l(t)$, thus

$$\frac{P_l(t)}{w_l(t)} = \frac{\overline{P}_i(t) - P_l(t) \sum_{j \in C_i(t) - \{\bar{l}\}} \frac{w_j(t)}{w_l(t)}}{w_{\bar{l}}(t)} \qquad l \in C_i(t) - \{\bar{l}\}. \tag{2.21}$$

Separating $P_l(t)$ results in

$$P_l(t) = \frac{w_l(t)\overline{P}_i(t)}{\sum_{j \in C_i(t)} w_j(t)} \qquad l \in C_i(t) - \{\bar{l}\}. \tag{2.22}$$

Equation (2.22) can also be extended for element $\bar{l} \in C_i(t)$. This can be shown by plugging (2.22) into Eq. (2.18), which yields

$$P_l(t) = \frac{w_l(t)\overline{P}_i(t)}{\sum\limits_{j \in C_i(t)} w_j(t)} \qquad l \in C_i(t). \tag{2.23}$$

These statements are valid for all network nodes $i \in \mathcal{M}^+(t)$. □

Corollary 2.2.2. *Problem (2.13a)-(2.13e) and problem (2.16a)-(2.16c) are equivalent (i.e. from solution of one, solution of the other is found, and vice versa).*

Proof. Using corollary 2.2.1, we can rewrite the objective of the problem (2.13a)-(2.13e) as:

$$\begin{aligned}
&\sum_{l \in \mathcal{N}(t)} w_l(t) \log(P_l(t)) = \\
&= \sum_{i \in \mathcal{M}^+(t)} \sum_{l \in C_i(t)} w_l(t) \log \left(\frac{w_l(t)\overline{P}_i(t)}{\sum\limits_{j \in C_i(t)} w_j(t)} \right) \\
&= \sum_{i \in \mathcal{M}^+(t)} \sum_{l \in C_i(t)} \left(w_l(t) \log(w_l(t)) - w_l(t) \log \left(\sum_{j \in C_i(t)} w_j(t) \right) + w_l \log(\overline{P}_i(t)) \right) \\
&= \sum_{i \in \mathcal{M}^+(t)} \sum_{l \in C_i(t)} w_l(t) \left(\log(w_l(t)) - \log \left(\sum_{j \in C_i(t)} w_j(t) \right) \right) + \sum_{i \in \mathcal{M}^+(t)} \overline{w}_i(t) \log(\overline{P}_i(t)).
\end{aligned}$$

The first term is a constant, which means it can be removed from the objective, the second term corresponds to the objective of problem (2.16a)-(2.16c). Thus, after substituting variables following Eq.(2.15) in problem (2.13a)-(2.13e), the problem (2.16a)-(2.16c) is obtained. □

Corollaries 2.2.1 and 2.2.2 have interesting practical implications for numerical experiments. They show that the problem (2.16a)-(2.16c) with smaller number of decision variables can be solved and then using Eq. (2.17), a solution of usually much larger problem (2.13a)-(2.13e) is obtained.

3. Electric vehicle charging strategies

As presented in (1), electric vehicle charging strategy is a software agent representing a real vehicle in the electric network. Its purpose is to influence the charging process by updating a non-negative vehicle specific (or strategy instance specific) parameter called *willingness to pay*. Formula updating a willingness to pay parameter might use a time denoted as t_c^l. Time t_c^l is not the current (simulation) time t, it represents the length of stay of a specific vehicle l in the network. This means that $t_c^l = 0$ when vehicle l arrives to the electric network. Time t_c^l can be computed from the time t by $t_c^l = t - t_{arrival}^l$, where $t_{arrival}^l$ denotes arrival time of vehicle l.

This section is divided into two parts. First part contains existing charging strategies, the second part focuses on design and presentation of the strategies created in this thesis.

3.1 Already existing strategies

In [1], electric vehicle charging strategies named *static user, inflexible user, flexible user* were presented. Analysis of these strategies gave us useful information, which were used when designing new electric vehicle charging strategies.

3.1.1 Static user

Static user is a simple charging strategy that does not change its willingness to pay during the charging process. Willingness to pay of all static users is set to 1.

3.1.2 Inflexible user

Inflexible user has limited charging time T_{max}^l, in which it aims to charge its battery fully. Vehicle represented by this strategy leaves the network when T_{max}^l elapses or its battery is charged fully. Intention of this strategy is to follow a linear charging function defined by points $B(0) = 0, B(T_{max}^l) = B_{max}^l$. Willingness to pay is set by formula:

$$w_l^{infl}(t) = \max\left\{ w_l^{infl}(t - \Delta t) - \kappa \Delta t \left(B^l(t) - \frac{t_c^l}{T_{max}^l} B_{max}^l \right), 0 \right\}. \tag{3.1}$$

Equation (3.1) shows that when battery state $B(t)$ is lower than the expected battery state $\frac{t_c^l}{T_{max}^l} B_{max}^l$, inflexible user strategy increases its willingness to pay, otherwise willingness to pay is decreased. In some cases willingness to pay can be set to zero, which means that this user does not want to receive any electric power.

3.1.3 Flexible user

Flexible users aim to charge their battery fully in time T^l_{max}. They leave the electric network when their battery is fully charged or when T^l_{max} elapses. Unlike inflexible users, flexible users do not follow the linear charging function strictly, their willingness to pay is updated by formula:

$$w_l^{min}(t) = \frac{1 + \frac{B^l_{max} - B^l(t)}{B^l_{max}}}{T^l_{max} - t^l_c} \tag{3.2}$$

$$w_l^{max}(t) = 1000 w_{min}(t) \tag{3.3}$$

$$w_l^{flex}(t) = \begin{cases} w_l^{min}(t), & \text{if } w_l^{infl}(t) \leq w_l^{min}(t) \\ w_l^{infl}(t), & \text{if } w_l^{min}(t) < w_l^{infl}(t) \leq w_l^{max}(t) \\ w_l^{max}(t), & \text{if } w_l^{max}(t) < w_l^{infl}(t) \end{cases}. \tag{3.4}$$

3.2 Strategies with budget

Strategies presented in [1] do not interpret *willingness to pay* parameter as the price per unit of time, that the users pay during the charging process, it is only used as a parameter influencing the charging process. In the strategies designed in this thesis, *willingness to pay* will represent a price that users pay per unit of time. All strategies have a limited spending budget W^l_{max} that the vehicle l can use during the charging process and a limited time to stay in the network T^l_{max}. Remaining budget at time t denoted as $W^l_r(t)$, is updated by formula

$$W^l_r(t) = W^l_r(t - \Delta t) - w_l(t - \Delta t)\Delta t. \tag{3.5}$$

Vehicles represented by strategies with limited budget disconnect at time t in three cases: their battery was charged fully ($B^l(t) = B^l_{max}$), the budget was spent entirely $W^l_r(t) = 0$, or the maximum charging time elapsed $t^l_c = T^l_{max}$. Vehicles are not allowed to spend more than W^l_{max}, if $w_l(t) > W^l_r(t)\Delta t$, then $w_l(t) = W^l_r(t)\Delta t$.

Intention behind introducing a limited spending budget W^l_{max} to all designed charging strategies is to prevent unlimited growth of willingness to pay in situations when the electric network is congested and cannot allocate more electric power, in a more natural way than in [1]. Using remaining budget $W^l_r(t)$ while making a decision about the willingness to pay could lead to some new interesting behavioural patterns.

3.2.1 Uniform spending in time strategy (UT strategy)

Aim of this user strategy is to spend a fixed amount of money per unit of time. Willingness to pay of the uniform spending strategy is computed from formula

$$w_l^{UT}(t) = \frac{W_{max}^l}{T_{max}^l} \tag{3.6}$$

and it does not change during the charging process. Even though this strategy is an extension of *Static user* from [1], parameters T_{max}^l and W_{max}^l create more possibilities for experimentation than the static user. For example, experiments could focus on varying W_{max}^l and T_{max}^l of vehicles entering the network. Homogeneous population of vehicles represented by UT strategy with the same maximum spending budget W_{max}^l and the same T_{max}^l would behave exactly like a homogeneous population of static users from [1], only with limited charging time.

3.2.2 Uniform charging in time strategy (UC strategy)

Uniform charging in time strategy extends inflexible user strategy presented in [1] by introducing a limited budget W_{max}^l. Willingness to pay is updated using the formula (3.1). This charging strategy is very similar to the inflexible user, on the other hand limited spending budget W_{max}^l may result in some alterations of its behaviour. For example, when the strategy is not able to follow the linear charging function, the vehicle l may spent its budget W_{max}^l before reaching maximum battery capacity and leave the network before T_{max}^l.

3.2.3 Affordable price spending strategy (AP strategy)

Affordable price spending strategy was inspired by article [11], specifically by the *File-transfer* user. Affordable price spending strategy updates its willingness to pay by comparing the current price per unit of energy that it is paying $\frac{w_l(t-\Delta t)}{P_l(t)}$, with the price per unit of energy that this user could afford to pay $\frac{W_r^l(t)}{B_{max}^l - B^l(t)}$. If the paid price per unit of energy at time t is lower than the price that the strategy is capable of paying, the willingness to pay $w_l(t)$ is increased and vice versa. Willingness to pay is updated using formula:

$$w_l^{AP}(t) = \max\left\{w_l^{AP}(t-\Delta t) + \kappa\Delta t\left(\frac{W_r^l(t)}{B_{max}^l - B(t)} - \frac{w_l^{AP}(t-\Delta t)}{P_l(t)}\right), w_{min}^l\right\} \tag{3.7}$$

Parameter $w_{min}^l > 0$ forces the strategy representing vehicle l to pay at least minimum amount of money when the price per unit of energy is perceived as too high.

3.2.4 Combination of affordable spending and uniform per time spending strategy (AUT strategy)

AUT strategy is an extension of affordable price spending strategy. Goal of this strategy is maximization of chances to gain as much electric energy as possible, without trying to save budget. Unlike the affordable price spending strategy, AUT strategy uses information about remaining time in the network $T^l_{max} - t^l_c$ when updating willingness to pay. When vehicle l is present in the network longer than $t^l_c = d^l T^l_{max}$, AUT strategy starts considering paying a fraction of price per unit of remaining time $\frac{W^l_r(t)}{T^l_{max}-t^l_c}$, parameter $d^l \in \langle 0, 1)$ represents a fraction of T^l_{max}. Parameter d^l is used to compute fraction $\alpha(t^l_c) \leq 1$ to multiple price $\frac{W^l_r(t)}{T^l_{max}-t^l_c}$ by, $\alpha(t^l_c)$ increases linearly. When $t^l_c = T^l_{max}, \alpha(t^l_c) = 1$, when $t^l_c = d^l T^l_{max}, \alpha(t^l_c) = 0$. Value $\alpha(t^l_c)$ is computed from linear function in the form $\alpha(t^l_c) = at^l_c + b$. Line properties a, b are equal to:

$$a = \frac{1}{T^l_{max}(1 - d^l)} \tag{3.8}$$

$$b = -\frac{d}{1 - d^l}. \tag{3.9}$$

Formula used to change willingness to pay is then equal to:

$$w^{AUT}_l(t) = \max\left\{w^{AP}_l(t), \alpha(t^l_c)\frac{W^l_r(t)}{T^l_{max} - t^l_c}\right\}. \tag{3.10}$$

Example evolution of $\alpha(t^l_c)$ during charging time, where $d^l = 0.5, T^l_{max} = 500$ is in the figure (3.1).

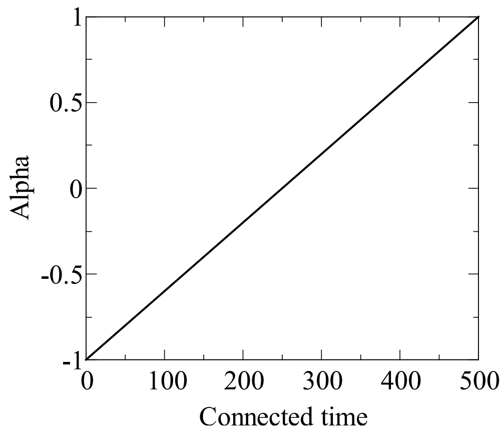


Figure 3.1: Computation of alpha

4. Implementation

4.1 Mathematical model

4.1.1 Limiting sensitivity of model to changes in occupancy of electric network

From the implementation point of view, formulation of model (2.16a)-(2.16c) requires creation of a variable $\overline{P}_i(t)$ in case when a vehicle connects to a node $i \in \mathcal{V}^+ \mid C_i(t) = \emptyset$ and removal of a variable $\overline{P}_i(t)$ when vehicle disconnect from a node $i \in \mathcal{V}^+ \mid |C_i(t)| = 1$. Created/removed variable $\overline{P}_i(t)$ has to be also added/removed to/from constraints (2.16b).

To keep the structure of decision variables consistent, regardless of changes in number of vehicles, a further alteration of the model (2.16a)-(2.16c) can be made, by creating variable $\overline{P}_i(t)$ for all $i \in \mathcal{V}^+$ and by adding constraints that enforce assignment of zero electric energy to nodes $i \notin \mathcal{M}^+(t)$. Addition of the new constraints and decision variables results in a model with a constant number of decision variables $4|\mathcal{N}| - 3$, which is solely dependent on the structure of electric network and thus is not sensitive to changes in occupancy in electric network. Resulting model takes the form:

$$\underset{V(t),\overline{P}(t)}{\text{maximize}} \quad \sum_{i \in \mathcal{M}^+(t)} \overline{w}_i(t) \log(\overline{P}_i(t)) \tag{4.1a}$$

subject to

$$\overline{P}_i(t) = 0 \qquad i \notin \mathcal{M}^+(t) \tag{4.1b}$$

$$\sum_{j|e_{ij} \in \mathcal{E}} (g_{ij}(V_{ii}(t) - Re\{V_{ij}(t)\}) + b_{ij} Im\{V_{ij}(t)\}) = \overline{P}_i(t) \qquad i \in \mathcal{V}^+ \tag{4.1c}$$

$$(2.13c) - (2.13e). \tag{4.1d}$$

The only part of the model (4.1a)-(4.1d) sensitive to arrival and departure of vehicles is the set of constraints (4.1b). It is much more convenient to update the constraints (4.1b), than to handle changes in the structure of decision variables.

4.1.2 Overview of solvers

Before actual implementation, a solver capable of solving problem (4.1a)-(4.1d) had to be chosen. Report [1] that deals with the same mathematical model states, that the package *CVXOPT* for Python programming language was used. The report also states that in some cases, optimization process failed due to numerical problems. Because of these issues, other solvers capable of solving the OPF problem were searched first.

JOptimizer

JOptimizer is an optimization library for JAVA programming language. It is capable of solving problems such as LP, QP, QCQP, SOCP, SDP. Objective function that this solver can deal with has to be twice differentiable real function. In order to add a new type of objective function, Java class representing objective function has to implement interface *ConvexMultivariateRealFunction* [12].

Official website of JOptimizer was basically the only source of information I found about this library, the documentation is not very detailed and there is basically no community supporting developers using this library.

CVXOPT

CVXOPT is a free software package for convex optimization based on the Python programming language. It can be used with the interactive Python interpreter, on the command line by executing Python scripts, or integrated in other software via Python extension modules. Its main purpose is to make the development of software for convex optimization applications straightforward by building on Python's extensive standard library and on the strengths of Python as a high-level programming language [13]. CVXOPT is well documented, it is capable of solving OPF problem and contains many useful examples to learn from.

CVXPY

CVXPY is a Python-embedded modelling language for convex optimization problems. It allows users to express problems in a natural way that follows the math, rather than in the restrictive standard form required by solvers. CVXPY contains solvers such as ECOS, CVXOPT and SCS. Additional solvers can be used, but these have to be installed separately. CVXPY allows users to use object-oriented approach for constructing optimization problems [14].

Problem with this library is the fact that it uses disciplined convex programming (DCP), which makes optimization process slower because of the convexity check of the specified problem.

Gurobi

Gurobi is a professional solver with free academic licence. It supports common problem types such as Linear Programming (LP), Mixed-Integer Linear Programming (MILP), Quadratic Programming (QP), Mixed-Integer Quadratic Programming (MIQP), Quadratically Constrained Programming (QCP), Mixed-Integer Quadratically Constrained Programming (MIQCP). Gurobi website also states that their solution is numerically stable, so the problems with CVXOPT package mentioned in [1] could be avoided [15]. However, official support of Gurobi stated, that this solver only supports problems with linear, piecewise-linear or convex quadratic objective functions, so Gurobi cannot be used for solving (4.1a)-(4.1d).

Mosek

According to information on Mosek website, this solver is capable of solving all common types of problems and it can also deal with nonlinear expressions. Documentation of nonlinear capabilities of this solver contained a warning stating that even though Mosek is capable of dealing with nonlinear expressions, interface providing this functionality is very complicated to use and it is recommended to use this feature only if there is no other option [16].

Chosen solver

Despite numerical problems mentioned in [1] CVXOPT library was chosen. The main reason behind this choice is a very good documentation of the solver and many useful examples. CVXOPT has a support group [17], where the authors of the package answer questions and provide support. Prevention of numerical issues is presented in (5).

Since $CVXOPT$ does not contain a modelling language, the mathematical model (4.1a) -(4.1d) had to be implemented in $CVXOPT$ specific form. In order to represent the mathematical model, optimization matrices A, G and vectors b, h representing constraints of type $Ax = b$, $Gx \preceq h$ had to be created with specific structure. Vector x is a vector of decision variables $V_{ii}(t), Re\{V_{ij}(t)\}, Im\{V_{ij}(t)\}, \overline{P}_i(t)$ from the model (4.1a) -(4.1d). To support nonlinear objective, a function capable of determining a value of the objective function, gradient and hessian for specific point x had to be supplied. Detailed information about the structure and formulation of the nonlinear convex optimization problems can be found in an online user guide [17].

4.2 Simulation

Simulation is performed by two cooperating simulation cores. *Event simulation core* is the main simulation core, it uses event based simulation to model discrete actions. Events are planned in a calendar and are processed in non decreasing order of the occurrence time. When event is processed, simulation time is set to the time of event occurrence and event specific action is performed. Event that starts the entire simulation is called *car arrival*, which connects a new electric vehicle to a node randomly chosen from uniform distribution, and plans another car arrival event. Charging strategy of the arriving vehicle is generated from a car type generator, which uses an empiric distribution for choosing specific charging strategy type. Setup of the car type generator is explained in section (7.4.2). Time between arrival of two vehicles is generated from exponential distribution. Car arrival event also plans an event called *latest car departure event*, purpose of which is to ensure that electric vehicle l specified in the event does not stay in the network longer than T^l_{max}. If $t^l_c = T^l_{max}$ and the vehicle specified in the event is still in the network, event called *car departure event* is planned for this vehicle at exactly the same time as the latest car departure event occurred. When car departure event is processed, specified electric vehicle disconnects from the network.

Continuous simulation core is responsible for simulating continuous charging process of connected electric vehicles. It increases simulation time by a specified time step Δt and then solves the optimization problem (4.1a)-(4.1d), optimal solution is then used to assign electric power to all vehicles connected to electric network at that specific simulation time. When the electric energy is assigned, all connected electric vehicles make a decision about its next willingness to pay based on the charging strategy they are represented by.

Continuous simulation core works in a time frame which is determined by the event simulation core. Length of the time frame is evaluated after each event processing and is equal to the time difference between the occurrence time of the currently processed event and the upcoming event. The time frame can either be used completely, or partially in case the work of continuous simulation core results in event that has to be processed immediately. This situation could for example result from charging a vehicle l fully, before the assigned time frame elapses. When this happens a departure event is planned for vehicle l with the event time equal to the current simulation time, when the event is processed vehicle l leaves the network. Cooperation between the simulation cores is visualized in the figure (4.1).

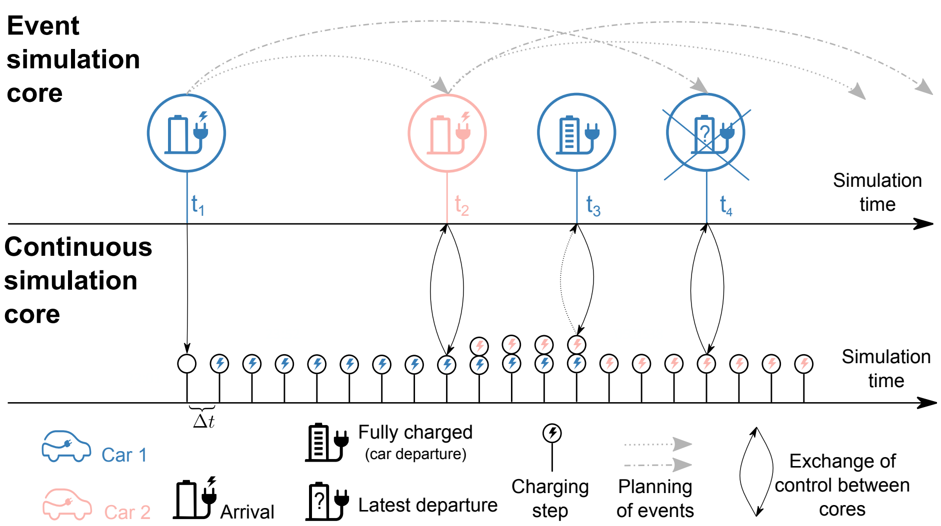


Figure 4.1: Cooperation of simulation cores

4.2.1 Simulation events

Description of currently implemented events with their actions is summarized below:

Simple car arrival: Electric vehicle l specified in the event arrives to the network and connects to a node specified in the event. Car arrival event can also plan latest car departure event in case the arriving vehicle l has a limited charging time T^l_{max}. This event does not plan next car arrival, its purpose is to support creation of simulation scenarios from section (6.3).

Car arrival: This event is an extension of *Simple car arrival event*, it performs the same logic described in the previous event. The only difference is that this event always plans arrival of next vehicle. Event was created to support experiments with random car arrivals.

Latest car departure: This event is planned after arrival of electric vehicle l with limited maximum charging time T^l_{max}. If maximum charging time is reached and the vehicle l is still in the network and not charged fully, car departure event is planned with the occurrence time equal to the occurence time of the latest car departure event.

Car departure: When this event is processed, electric vehicle l specified in the event disconnects from the electric network. This event is either planned by the event simulation core in case the vehicle was not charged fully in specified maximum time T^l_{max} or by the continuous simulation core, in case the electric vehicle l was fully charged or it ran out of resources needed for charging (W^l_{max}).

Visualization of the events is in the figure (4.2), solid directed line from one simulation event to another, means that the event from which the line starts always plans the event to which the line goes to, if the line is dashed, it means that the first event plans the other event conditionally.

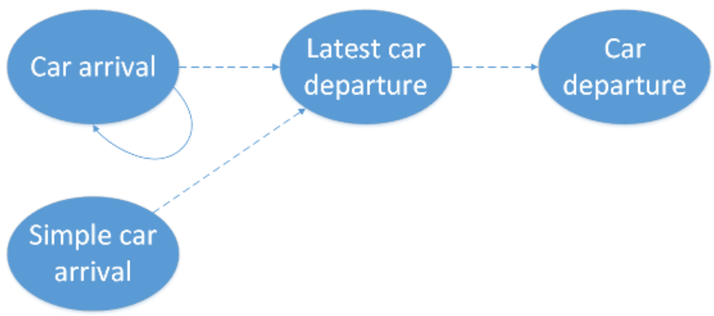


Figure 4.2: Simulation events

Each event is implemented as a separate class, the most general representation of a simulation event is implemented in abstract class called *SimulationEvent*, from which other simulation events inherit attributes. Structure of events is summarized in class diagram (4.3).

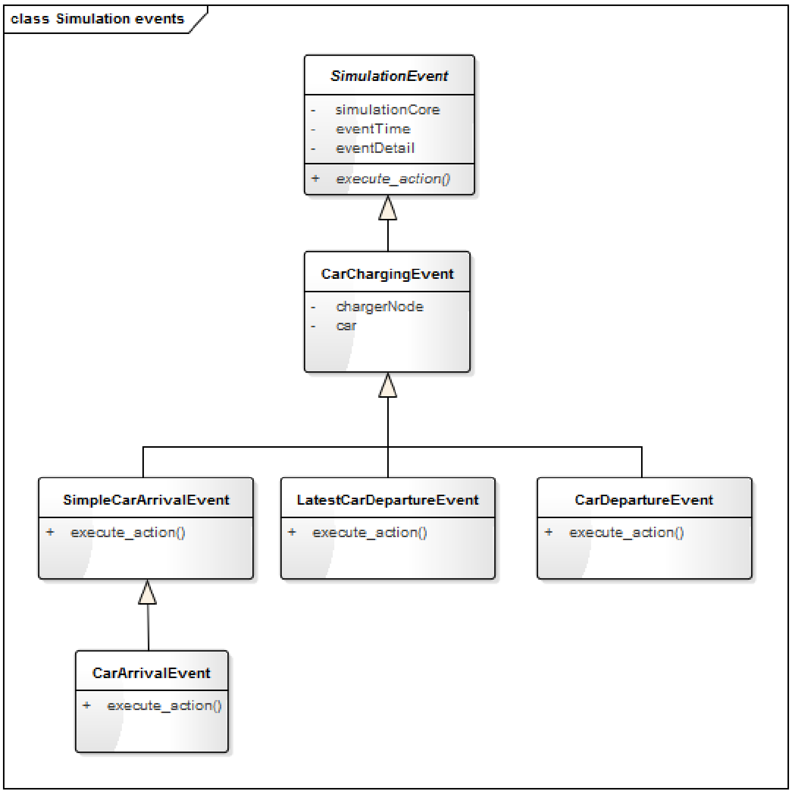


Figure 4.3: Structure of simulation events

4.2.2 Collecting simulation data

During simulation, multiple statistics are collected. Each electric vehicle stores its battery state evolution, willingness to pay evolution, evolution of budget (in case the vehicle is represented by a strategy with budget) and statistic that automatically computes average willingness to pay during the charging process. If the simulation is run with one replication (detailed mode of the simulation), all vehicle related data are saved to separate files when electric vehicle disconnects from the network or the simulation run is finished. The rest of statistics are stored in the event simulation core and are updated while processing specific discrete events. For example, when electric vehicle leaves the network, statistics including, but not limited to, observing average number of vehicles in the system and average battery state are updated. Statistics that are observed during the simulation are found in the section (6.4).

Statistics are implemented as Python classes, which means that the observed data can be easily extended. Extension of statistics only require creating a new instance of statistics class and then populating this instance with observed data.

Simulation is tracked by simulation observer, which is an element that simulation cores notify when replication starts, replication ends, simulation ends, an event occurs or there is a failure in the optimization algorithm. To receive notifications, simulation observer has to register to the simulation core. Multiple simulation observers can be registered to one simulation core. Simulation observers can be used to present state of observed data to the user during simulation experiment. Simulation tool implemented in this thesis does not have a graphical user interface, but the implemented simulation observer saves notifications to simulation log file. Besides tracking the simulation, log file was used to verify that the simulation itself is implemented properly and is very useful for debugging purposes.

Analysis of simulation data

When the simulation is run in the detailed mode, three or four files (depending on the charging strategy) containing data about charging process are created for each individual vehicle. In order to analyze behaviour of individual vehicles separately, graphs depicting behaviour of vehicles present in the simulation experiment had to be created from the simulation files. Creation of graph for one specific vehicle requires combining data from three or four vehicle-specific files, which would be very time consuming for simulation runs with huge number of vehicles. To facilitate analysis of individual vehicles, a tool supporting automatic creation of simulation graphs was created. Created tool is called *Grace graph generator* and uses a program called *QtGrace*. The tool reads a directory containing simulation files and combines vehicle-specific files to create a single graph for each vehicle in the simulation. This enabled us to study behaviour of individual vehicles effectively. The tool first creates *QtGrace* specific graph files with .agr extension and then utilizes command line interface of *QtGrace* to create images in .pdf or .jpg format.

Purpose of the *Grace graph generator* is to make analysis of individual vehicles more efficient, however, for image publishing purposes, output of this tool may need additional processing in the *QtGrace* program. Since *Grace graph generator* creates *Qtgrace* specific files with all vehicle-related data, manual processing of the generated graph in the *QtGrace* program is much faster, than creating the entire graph from scratch.

5. Numerical issues

CVXOPT library that is used for solving the convex optimization problem (4.1a)-(4.1d) may not be able to find a solution in some cases and returns an error message *terminated: singular KKT matrix*. Willingness to pay parameters that each instance of electric vehicle charging strategy sets and updates during the charging process influence $\overline{w}_i(t), i \in \mathcal{V}^+$, according to [18] a few large elements in the model generally destroy accuracy in the solver, therefore huge *willingness to pay* parameters may lead to numerical issues.

5.1 Model scaling strategies

To deal with the numerical issues, a set of model scaling strategies were introduced. Purpose of the model scaling strategies is to alter the model (4.1a)-(4.1d) to make the computation of optimization problem more stable, while preserving the optimal solution of the original non-scaled model. Created scaling strategies are:

Sum scaling: Willingness to pay $\overline{w}_i(t)$ is divided by $\sum_{i \in \mathcal{M}^+(t)} \overline{w}_i(t)$. This scaling strategy was also used in [1].

Max scaling: Willingness to pay $\overline{w}_i(t)$ is divided by $\max\{\overline{w}_i(t),\ i \in \mathcal{M}^+(t)\}$

Scaling equality constraints: CVXOPT user group [18] states that it often helps when optimization matrix A that is part of $Ax = b$ constraints has $\text{rank}(A) < 1$. This strategy divides A, b matrices used in CVXOPT by *max-abs-row-sum norm:* $\max_i \sum_{j=1}^{n} |a_{ij}|$ of matrix A. Sum scaling is used for willingness to pay $\overline{w}_i(t)$.

Scaling nominal voltage : Voltage $V_{nominal}$ in constraint (2.14) is iteratively scaled by factor $\beta > 0$. β starts at specified minimum value β_{min} and increases until either optimal solution is found or specified β_{max} is reached. Sum scaling is used for willingness to pay $\overline{w}_i(t)$. Theoretical background behind this scaling strategy is in the section (5.1.1).

5.1.1 Scaling nominal voltage

Scaling $V_{nominal}$ in constraints (2.14) by factor $\beta > 0$ produces a problem equivalent to the problem (4.1a)-(4.1d), with scaled decision variables $\overline{V_{ii}(t)}, \overline{Re\{V_{ij}(t)\}}$, $\overline{Im\{V_{ij}(t)\}}, \overline{\overline{P}_i(t)}$. Symbolic representation of non-scaled and scaled model, respectively, is below:

$$\begin{array}{ll}
\text{maximize} \sum_{i\in\mathcal{M}^+(t)} \overline{w}_i(t)\overline{P}_i(t) & \text{maximize} \sum_{i\in\mathcal{M}^+(t)} \overline{w}_i(t)\overline{\overline{P}_i(t)} \\
\text{subject to} & \text{subject to} \\
f_k(V_{ii}(t), Re\{V_{ij}(t)\}, Im\{V_{ij}(t)\}) \leq 0 & f_k(\overline{V_{ii}(t)}, \overline{Re\{V_{ij}(t)\}}, \overline{Im\{V_{ij}(t)\}}) \leq 0 \\
h_l(V_{ii}(t), Re\{V_{ij}(t)\}, Im\{V_{ij}(t)\}, \overline{P}_i(t)) = 0 & h_l(\overline{V_{ii}(t)}, \overline{Re\{V_{ij}(t)\}}, \overline{Im\{V_{ij}(t)\}}, \overline{\overline{P}_i(t)}) = 0.
\end{array}$$

To derive the relationships between decision variables of the problems with non-scaled and scaled $V_{nominal}$, two equivalent transformations will be performed on the original non-scaled model (4.1a)-(4.1d).

In the first step, all constraints are multiplied by β^2 and a constant is added to the objective function of the original model. According to [3], multiplying constraints by a non-negative constant does not change the feasible set. A point x is optimal for the original problem if and only if it is optimal for the problem with scaled constraints, so the original and the scaled problem have the same optimal solution. Adding a constant to the objective function does not change the optimal solution, either. Multiplying both equality and inequality constraints by β^2 yields a following optimization problem:

$$\text{maximize} \quad \sum_{i\in\mathcal{M}^+(t)} \Big(\overline{w}_i(t)\log(\overline{P}_i(t)) + \overline{w}_i(t)\log(\beta^2)\Big) \tag{5.1a}$$

subject to

$$\left\| \begin{pmatrix} 2\beta^2 Re\{V_{ij}(t)\} \\ 2\beta^2 Im\{V_{ij}(t)\} \\ \beta^2 V_{ii}(t) - \beta^2 V_{jj}(t) \end{pmatrix} \right\|_2 \leq \beta^2 V_{ii}(t) + \beta^2 V_{jj}(t) \qquad e_{ij} \in \mathcal{E}^+, \tag{5.1b}$$

$$(1+\alpha)^2(\beta V_{nominal})^2 \geq \beta^2 V_{ii}(t) \qquad i \in \mathcal{V}, \tag{5.1c}$$

$$(1-\alpha)^2(\beta V_{nominal})^2 \leq \beta^2 V_{ii}(t) \qquad i \in \mathcal{V}, \tag{5.1d}$$

$$\beta^2 \overline{P}_i(t) = 0 \qquad i \notin \mathcal{M}^+(t), \tag{5.1e}$$

$$\sum_{j|e_{ij}\in\mathcal{E}} (g_{ij}(\beta^2 V_{ii}(t) - \beta^2 Re\{V_{ij}(t)\}) + b_{ij}\beta^2 Im\{V_{ij}(t)\}) = \beta^2\overline{P}_i(t) \qquad i \in \mathcal{V}^+, \tag{5.1f}$$

$$\sum_{j|e_{ij}\in\mathcal{E}} (b_{ij}(\beta^2 V_{ii}(t) - \beta^2 Re\{V_{ij}(t)\}) - g_{ij}\beta^2 Im\{V_{ij}(t)\} = 0 \qquad i \in \mathcal{V}^+. \tag{5.1g}$$

Objective function (5.1a) can be replaced by $\sum_{i\in\mathcal{M}^+(t)} \overline{w}_i(t)\log(\beta^2\overline{P}_i(t))$.

In the next step, all decision variables multiplied by β^2 are substituted. This transformation of the optimization problem is defined in [3] as a change of variables and it results in an equivalent optimization problem. In this case, optimal solution of the problem with substituted variables is not the same as in the original model, however the optimal solution of the original model can be computed based on the performed substitution. Substitution of decision variables results in the following model:

$$\text{maximize} \quad \sum_{i \in \mathcal{M}^+(t)} \overline{w}_i(t) \log(\overline{\overline{P}_i(t)}) \tag{5.2a}$$

subject to

$$\left\| \begin{pmatrix} 2\overline{Re\{V_{ij}(t)\}} \\ 2\overline{Im\{V_{ij}(t)\}} \\ \overline{V_{ii}(t)} - \overline{V_{jj}(t)} \end{pmatrix} \right\|_2 \leq \overline{V_{ii}(t)} + \overline{V_{jj}(t)} \qquad e_{ij} \in \mathcal{E}^+, \tag{5.2b}$$

$$(1+\alpha)^2(\beta V_{nominal})^2 \geq \overline{V_{ii}(t)} \qquad i \in \mathcal{V}, \tag{5.2c}$$

$$(1-\alpha)^2(\beta V_{nominal})^2 \leq \overline{V_{ii}(t)} \qquad i \in \mathcal{V}, \tag{5.2d}$$

$$\overline{\overline{P}_i(t)} = 0 \qquad i \notin M^+, \tag{5.2e}$$

$$\sum_{j|e_{ij} \in \mathcal{E}} (g_{ij}(\overline{V_{ii}(t)} - \overline{Re\{V_{ij}(t)\}}) + b_{ij}\overline{Im\{V_{ij}(t)\}}) = \overline{\overline{P}_i(t)} \quad i \in \mathcal{V}^+, \tag{5.2f}$$

$$\sum_{j|e_{ij} \in \mathcal{E}} (b_{ij}(\overline{V_{ii}(t)} - 2\overline{Re\{V_{ij}(t)\}}) - g_{ij}\overline{Im\{V_{ij}(t)\}}) = 0 \qquad i \in \mathcal{V}^+. \tag{5.2g}$$

To get optimal solution of the original problem (4.1a)-(4.1d), the problem with scaled $V_{nominal}$ is solved and optimal values of decision variables $\overline{V_{ii}(t)}, \overline{Re\{V_{ij}(t)\}}$, $\overline{Im\{V_{ij}(t)\}}, \overline{\overline{P}_i(t)}$ divided by β^2 are returned.

5.2 Scaling tests

In order to evaluate performance of model scaling strategies, two types of tests with three test cases in each type were created. In the first type of scaling tests, all presented model scaling strategies were tested separately, in the second type of tests, a composite scaling strategy containing all individual scaling strategies was analyzed.

Scaling tests were run for specified number of replications and specified number of optimization calls in each replication. In the beginning of each replication, exactly one vehicle was connected to each node $i \in \mathcal{V}^+$. Initial *willingness to pay* of vehicles was generated either from uniform distribution $\langle 0, 1)$ with probability 0.5 or from uniform distribution $\langle 1000, 5000)$ with the same probability. The initial setup was chosen based on our anticipation that huge differences in willingness to pay may be the source of numerical problems.

After initial setup, a specified number of optimization calls were performed. After each optimization call all of the connected cars would change their willingness to pay by adding a pseudorandom number generated from a uniform distribution to their current willingness to pay. In the first test case the willingness to pay change was generated from the interval $\langle -1, 1)$, in the second test case from $\langle -10, 10)$, in the third test case from interval $\langle -100, 100)$.

5.2.1 Tests of separate scaling strategies

Performance of all four model scaling strategies was evaluated on randomly generated optimization problems described in (5.2). 100 replications with 500 optimization calls were set for each test case, which means that each model scaling strategy was tested 50000 times. Failure rates of individual model scaling strategies are shown in the table (5.1).

Test case	Sum	Max	Nominal voltage	Equalities
1	0.0%	0.002%	0.0%	0.0%
2	0.0%	0.0%	0.0%	0.0%
3	0.0%	0.0%	0.0%	0.0%

Table 5.1: Performance of separate model scaling strategies

The only recorded failure of model scaling strategy was found in the first test case, where the max scaling strategy had an average error rate of 0.002%, which means that in average, max scaling strategy fails in 0.002% of times. Even though no other errors were recorded, there could still be cases where model scaling strategies may fail.

5.2.2 Scaling algorithm

The scaling algorithm (or a composite scaling strategy) is a chain of separate scaling strategies presented in section (5). Design of this algorithm allows to change the order of individual scaling strategies with only minor code changes. Additional model scaling strategy can be introduced with no impact on classes that use this algorithm. Algorithm is summarized in the activity diagram (5.1).

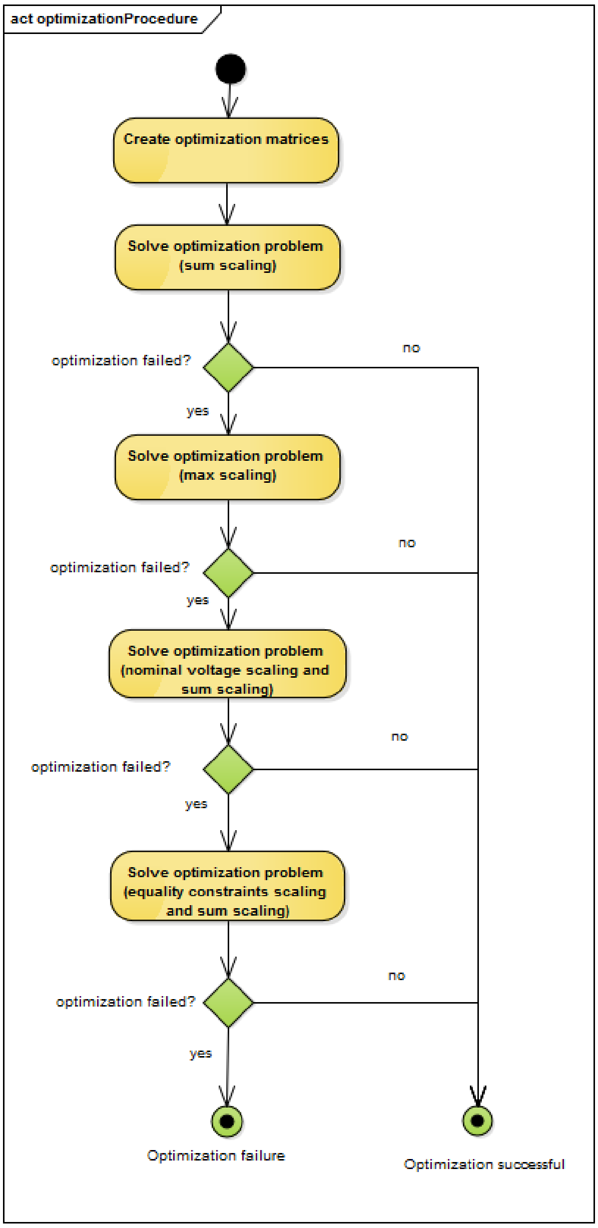


Figure 5.1: Optimization procedure

Optimization procedure starts with sum scaling. If this approach fails, then maximum scaling is used. Even though maximum scaling had a small error rate in the previous tests, this strategy is used because of its simplicity. In case maximum scaling fails as well, then the nominal voltage scaling is used. In case optimization fails again, then the equality constraints scaling strategy with sum scaling is used. If all four strategies fail, then the optimization failed as well.

Just like individual scaling strategy, the composite scaling strategy was tested with 3 different test cases described in section (5.2). All tests used 1000 replications with 100 optimization calls in each of the replication with following results:

Test case	Composite scaling strategy
1	0.0%
2	0.0%
3	0.0%

Table 5.2: Performance of composite scaling strategy

The composite scaling strategy had an error rate of 0% in all test cases, this algorithm was also used for simulation experiments with no failures. Definition of failure in the simulation experiments is very strict - model scaling strategy fails when CVXOPT returns status other than *optimal*, or when rank 1 check of relaxed condition (2.9f) fails.

From the implementation point of view, Composite scaling strategy uses a *Composite* design pattern, which makes the algorithm easy to change in case the order of scaling strategies has to change or new scaling strategy has to be introduced. Structure of scaling strategies is summarized in the class diagram (5.2).

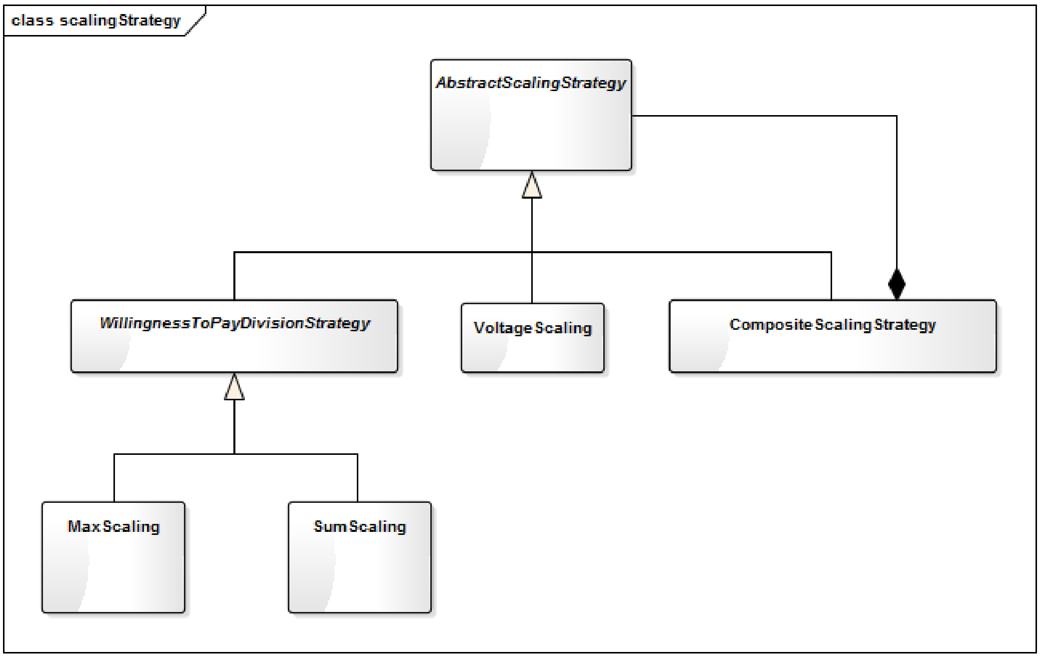


Figure 5.2: Conceptual class diagram of scaling strategies

6. Experiments

Experiments performed with the implemented simulation tool are divided into three sections:

- **Behaviour of individual users** - in this category, we focus on studying behaviour of individual vehicles represented by specific charging strategies.
- **Simulation scenarios** - in this category, a predefined situation called scenario will be created in order to study behaviour of the user strategies with budget in the created situations. Purpose of these experiments is to study a group behaviour of user strategies in smaller scale.
- **Performance of charging strategies** - in this category, we focus on performance of charging strategies from the system viewpoint.

6.1 Electric power assignment

Electric vehicle charging strategies representing a real vehicle in the electric network alter their willingness to pay in order to influence the charging process. However, willingness to pay is not the only factor influencing amount of assigned electric power. Another important aspect is a distance of vehicle from the source of electric power, because the model (4.1a)-(4.1d) considers power losses, when transmitting electric power through the lines. Generally, the further the vehicle connects from the source of electric power the less electric power it receives during the charging process. Illustration of impact of distance from the source of electric power on charging process is in the figure (6.1).

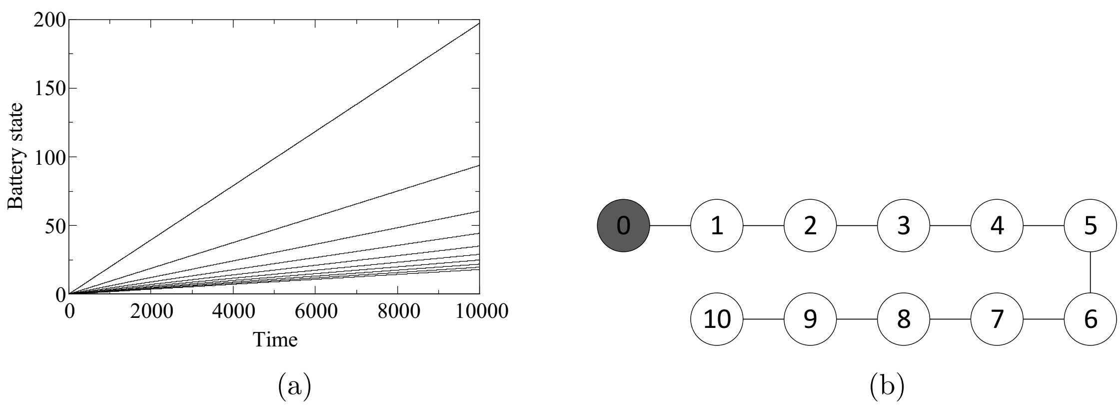


Figure 6.1: Network effect. Figure (6.1a) shows charging process of all vehicles, figure (6.1b) shows structure of the electric network.

Figure (6.1a) illustrates charging process of 10 different vehicles represented by the same charging strategy with the same constant willingness to pay. Vehicle $l = 1, 2, \ldots, 10$ is connected to node $i = l$, structure of the electric network is in the figure (6.1b), node 0 is the source of electric power. Figure (6.1a) shows that amount of electric power assigned to electric vehicles is decreasing with increasing distance from the source of electric power, the most electric power was consumed by vehicle 1 connected to node 1, which is adjacent to the source of electric power.

6.2 Behaviour of individual users

This section focuses on analysis of individual vehicles and their typical behaviour. Data about behaviour of the vehicles were collected by running one replication with initially empty electric network. In each experiment, only one type of user strategy was present. Two examples of individual vehicles present in the simulation experiment will be shown, one leaving the network with partially charged battery, the second one departing the network with full battery capacity. All graphs present examples taken from simulation experiments with average inter-arrival time set to 16.6. All vehicles arrive to electric network with empty battery $B(0) = 0$, their maximum battery capacity $B_{max} = 20$ and their maximum charging time $T_{max} = 500$. Initial willingness to pay is set to 1 (except for UT strategy), the maximum spending budget W_{max} is set to 1000. Electric network used in all experiments had line resistance $r_{ij} = 0.1$ and line reactance $x_{ij} = 0.6$, nominal voltage $V_{nominal} = 1$ and parameter α in set of constraints (2.14) is set to 0.1. Node denoted as 0 is the source of electric power. Structure of the electric network is in the figure (6.2). Simulation step $\Delta t = 1$. Dashed green line in the presented graphs shows a linear charging function defined by points $B(0) = 0$ and $B(T_{max}) = B_{max}$, which inflexible, flexible and UT strategies use when making a decision about willingness to pay.

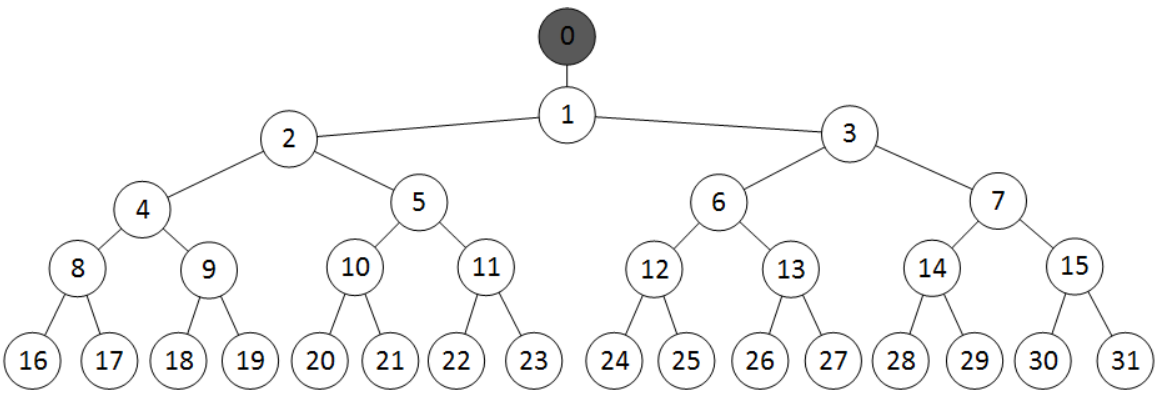


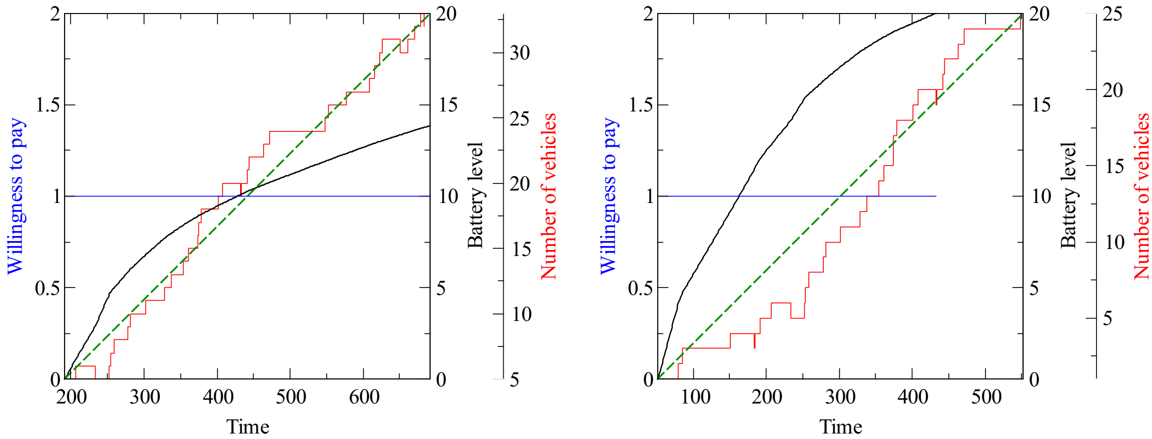
Figure 6.2: Electric network used in experiments

6.2.1 Existing strategies

Simulation experiments started with strategies presented in [1], to test whether behaviour described in the report can be achieved by the simulation program that was developed in this thesis.

Static user

To be able to study partially charged vehicles, this strategy was extended by introducing a limited charging time T^l_{max}. Since static users do not change their willingness to pay during the charging process, assigned electric power will be solely dependent on position of the vehicle in the network and number of vehicles in the network. Generally, with increasing distance from the source of electric power and with increasing number of vehicles in the network, assigned electric power decreases.



(a) (b)

Figure 6.3: Behaviour of static user strategy. Figure (6.3a) shows a partially charged vehicle connected to node 6. Figure (6.3b) shows a fully charged vehicle connected to node 24.

Inflexible user

Vehicles represented by inflexible user strategy are either able to adjust their willingness to pay and follow the linear charging function, or the state of electric network does not allow to follow linear charging function strictly, which results in gradual increase of willingness to pay until either state of the network changes allowing the vehicle l to get back on the linear charging function, or the vehicle l leaves network because of T^l_{max} limitation. Both cases can be seen in the figures (6.4). Inflexible users presented in this section had $\kappa = 1$.

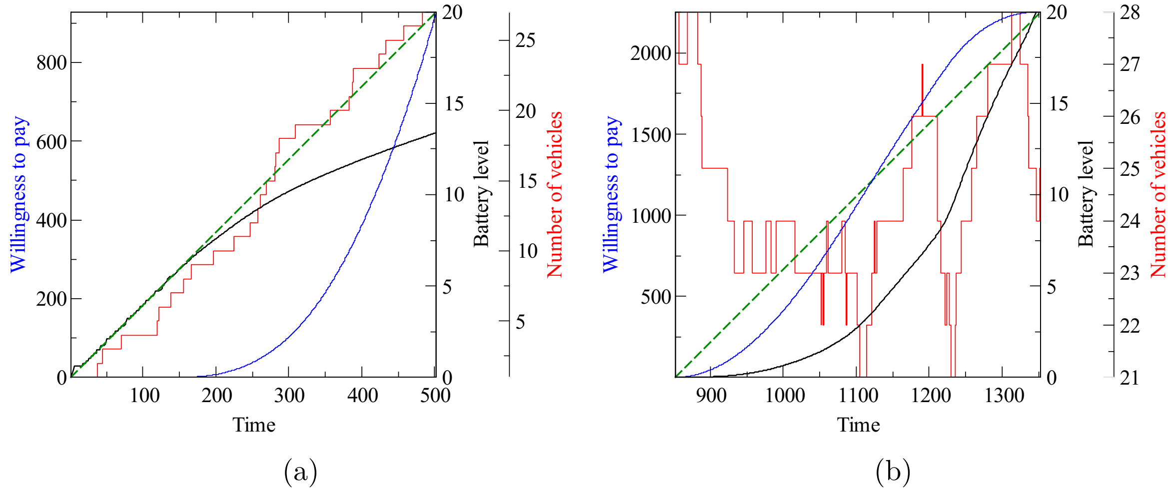


Figure 6.4: Behaviour of inflexible user strategy. Figure (6.4a) shows a partially charged vehicle connected to node 20. Figure (6.4b) shows a fully charged vehicle connected to node 1.

Figure (6.4a) depicting a charging process of the first vehicle arriving to the electric network shows disadvantage of the inflexible user strategy. After connecting to electric network, the vehicle gains more energy than expected, therefore after short time it decreases its willingness to pay to 0. After arrival of additional vehicles the vehicle failed to follow the linear charging function, which resulted in gradual increase of willingness to pay and later in departure with only partially charged battery. Figure (6.4b) shows charging process of a vehicle that managed to reach a full battery state, disadvantage of this behaviour is that the strategy pays too much money in situation, when electric network does not have enough resources to ensure that the vehicle can follow linear charging function. At the end of the charging time, state of electric network changed allowing the vehicle to gain full battery capacity. Inflexible user strategy cannot take advantage of situations when electric network has enough resources to charge the vehicle l fully long before T^l_{max} elapses and stays in the network longer than necessary.

Flexible user

Figure (6.5) shows that flexible users do not follow the linear charging function strictly. This is caused by introducing additional variables $w^l_{min}(t)$ and $w^l_{max}(t)$, but when $w^l_{min}(t) \leq w^l_{lin}(t) \leq w^l_{max}(t)$, behaviour of the flexible strategy is the same as the behaviour of inflexible strategy. Figure (6.5a) shows the situation when the charging time t^l_c approaches T^l_{max} and $B(t)$ is below linear charging function. This situation results in very aggressive increase in willingness to pay, because $w^l_{min}(t)$ is very large. This is justified by formula (3.2), which shows that willingness to pay could diverge to ∞. Just like inflexible user strategy, flexible user strategy does not take advantage of situations when electric network is capable of allocating more electric energy than expected. Parameter κ of flexible user strategy was set to 1000.

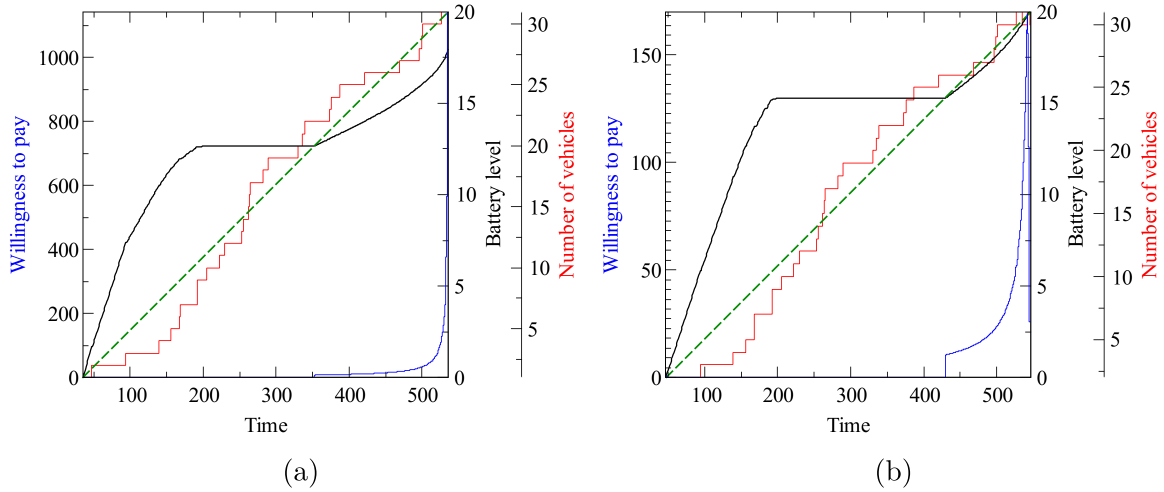


Figure 6.5: Behaviour of flexible user strategy. Figure (6.5a) shows a partially charged vehicle connected to node 21. Figure (6.5b) shows a fully charged vehicle connected to node 31.

6.2.2 Strategies with budget

Uniform spending in time strategy (UT strategy)

Behaviour of Uniform spending in time strategy is generally very simple. As seen in the figure (6.6), UT strategy always spends the same amount of money regardless of remaining battery to charge, time left in the network or the remaining spending budget. UT strategy representing vehicle l either spends its entire budget at $t_c^l = T_{max}^l$, or it reaches full battery state before T_{max}^l elapses, which means that the remaining budget at the time of departure is greater than zero.

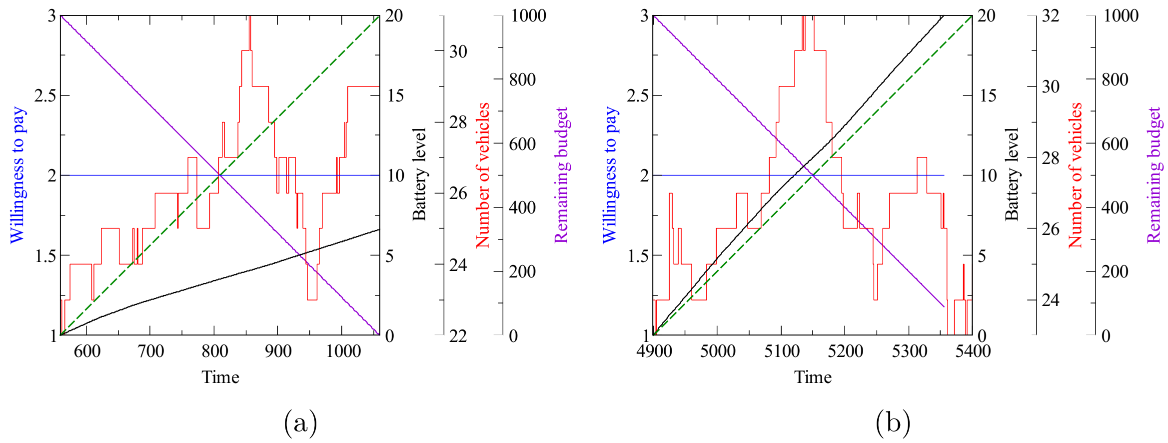


Figure 6.6: Behaviour of UT strategy. Figure (6.6a) shows partially charged vehicle connected to node 15. Figure (6.6b) shows fully charged vehicle connected to node 1.

Uniform charging in time strategy (UC strategy)

Uniform charging in time strategy spends its budget in order to follow a linear charging function defined by points $B(0) = 0, B(T^l_{max}) = B^l_{max}$. If the network has enough resources, the UC strategy is capable of following the linear charging function. On the other hand, if the electric network cannot fulfill this demand, this strategy may start spending too much money, which may result in departure from electric network because the budget was spent entirely. This behaviour can be seen in figure (6.7), parameter κ was set to 1.

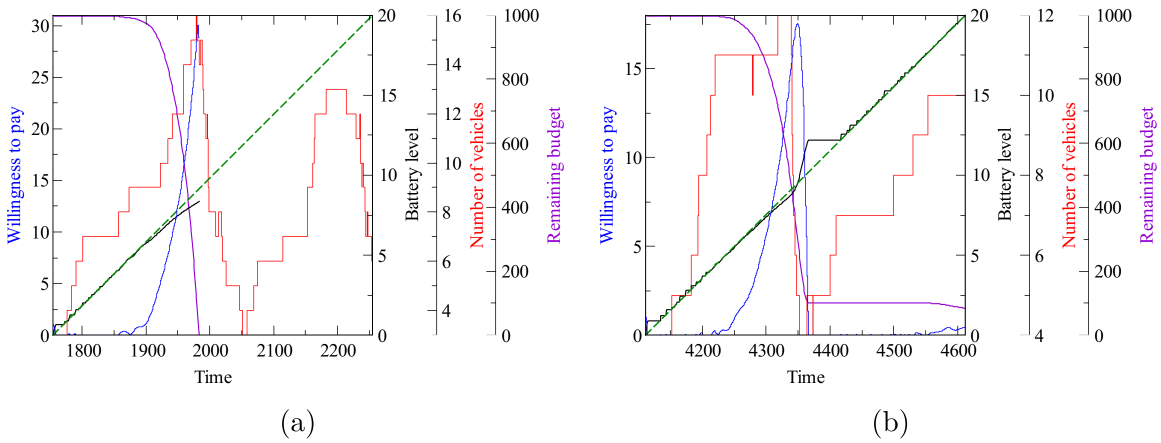
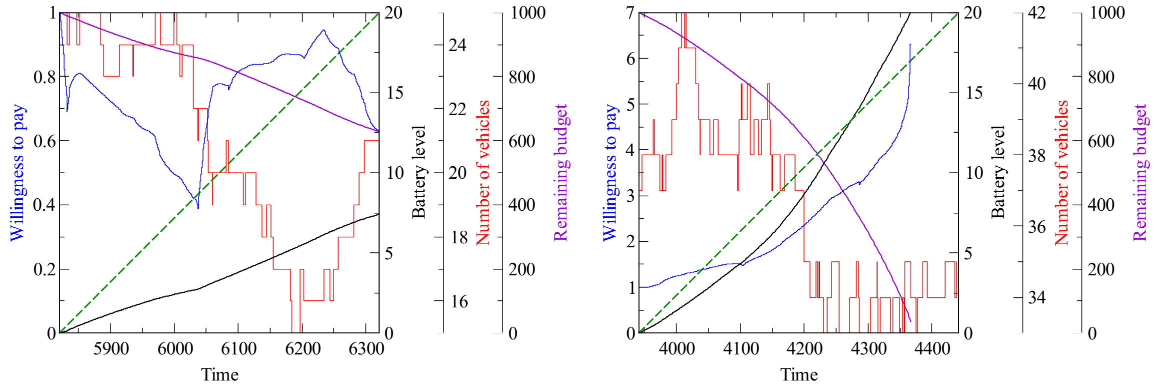


Figure 6.7: Behaviour of UC strategy. Figure (6.7a) shows a partially charged vehicle connected to node 11. Figure (6.7b) shows a fully charged vehicle connected to node 3.

Affordable price spending strategy (AP strategy)

Very interesting behaviour was observed when analyzing individual vehicles represented by affordable price spending strategy. End of charging process of vast majority of vehicles reaching full battery capacity was marked by a gradual increase of willingness to pay, example of a specific vehicle is in the figure (6.8b). If a vehicle l represented by AP strategy has a relatively high remaining budget $W^l_r(t)$ and relatively small amount $B^l_{max} - B^l(t)$ left to charge, it can increase its willingness to pay more aggressively, which may result in increasing price per unit of energy for other users. If the price per unit of remaining energy to charge is too expensive, AP strategy lowers the price it pays. Price per unit of energy is determined by a specific instance l of AP strategy, which means that the price at time t is not the same for all vehicles connected to the electric network. One specific instance of AP strategy $l = 1$ might consider its current price per energy too expensive, while the other instance $l = 2$ considers its price cheap. Described behaviour is presented in the figure (6.8), parameter $\kappa = 0.001$, $w_{min} = 0.01$.

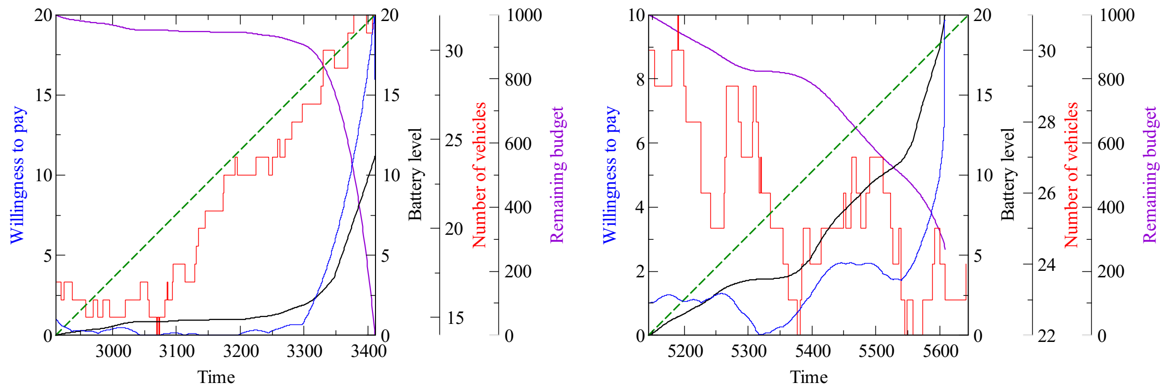


(a) (b)

Figure 6.8: Behaviour of AP strategy. Figure (6.8a) shows a partially charged vehicle connected to node 29. Figure (6.8b) shows a fully charged vehicle connected to node 6.

Combination of affordable spending and uniform per time spending strategy (AUT strategy)

AUT strategy behaves similarly to the AP strategy, however from specific time t, when $t_c^l = dT_{max}^l$, it might start paying a fraction of price per unit of remaining time $\frac{W_r^l(t)}{T-t_c^l}$ (in case it is bigger than the price they pay at time t), this means that from specific time, AUT strategy will try to spend more money in order to gain more electric energy (figure (6.9a)). Two vehicles represented by AUT strategy are in the figure (6.9), parameter d was set to $0.75, \kappa = 0.001$ and $w_{min} = 0.01$.



(a) (b)

Figure 6.9: Behaviour of AUT strategy. Figure (6.9a) shows a partially charged vehicle connected to node 12. Figure (6.9b) show a fully charged vehicle connected to node 1.

6.3 Simulation scenarios

In each of the following experiments, network with structure shown in the figure (6.10) was used. Each line e_{ij} has resistance $r_{ij} = 0.1$ and reactance $x_{ij} = 0.6$, nominal voltage $V_{nominal} = 1$ and parameter α in set of constraints (2.14) is set to 0.1. Node denoted as 0 is the source of electric power.

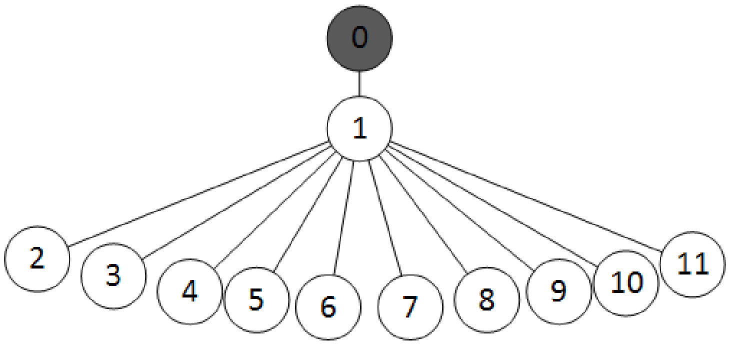


Figure 6.10: Electric network used in scenarios

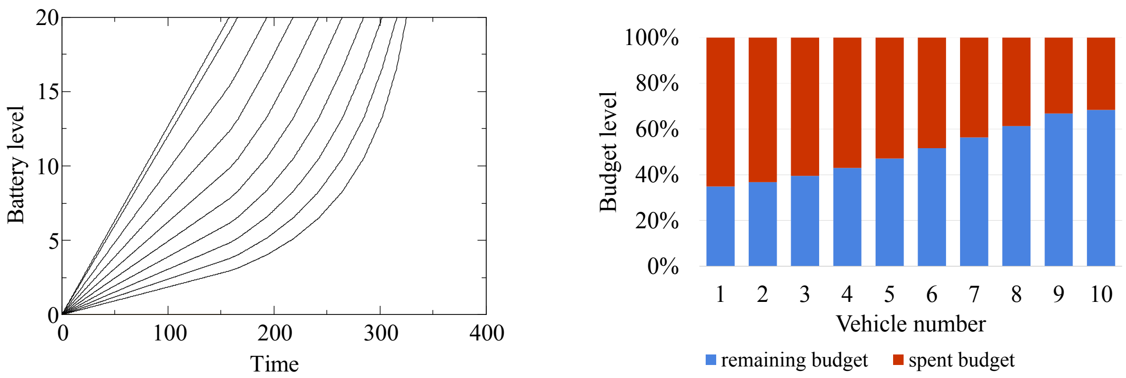
Structure of the electric network was chosen in order to suppress network effect described in section (6.1). In all scenarios, vehicles were only connected to nodes $2, \ldots, 11$ and were only represented by strategies with budget. To obtain more detailed data about behaviour of the strategies, simulation step $\Delta t = 0.5$ was used.

6.3.1 Budget variation scenario

In the budget variation scenario, exactly one vehicle was connected to nodes $2, \ldots, 11$. All vehicles had the same $B_{max} = 20, T_{max} = 500, B(0) = 0$ and arrival time 0, but their maximum spending budgets were different. In this scenario, vehicle l was connected to node $i = l + 1$, for $l = 1, \ldots$,10, $W^l_{max} = 1.3^{l-1}500$, for $l = 1, \ldots 10$. This specific setting was chosen in order to maintain the same relative differences between budgets of individual vehicles, while creating reasonable differences between the maximum and the minimum spending budgets.

Uniform spending in time strategy (UT Strategy)

Vehicles represented by UT strategy were charged in the non-increasing order of the maximum spending budget, meaning that the first vehicle reaching a full battery capacity was the vehicle with the maximum budget. Since UT strategy does not update its willingness to pay during the charging process, the only factor having impact on the speed of charging process is the number of vehicles in the network and their willingness to pay. Figure (6.11a) shows that charging process of vehicles with lower budget speeds up when vehicles with higher budget leave the network.

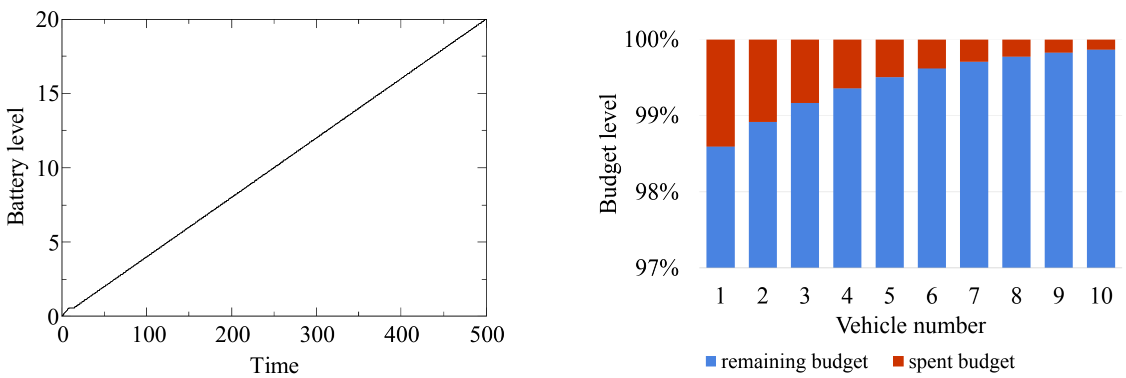


(a) (b)

Figure 6.11: UT strategy - performance in budget variation scenario. Figure (6.11a) shows charging process of all vehicles. Figure (6.11b) shows budget of individual vehicles.

Uniform charging in time strategy (UC strategy)

In this scenario, all vehicles represented by UC strategy reached full battery capacity. In the figure (6.12a), it is not possible to distinguish individual vehicles, because decision about updating willingness to pay is made regardless of the remaining spending budget $W_r^l(t)$, UC strategy only considers battery state $B^l(t)$ and remaining charging time $T_{max}^l - t_c^l$. Since all vehicles started with the same T_{max}^l, the charging process of all connected vehicles had to be the same. The only difference that could have occurred would be that some users could have left the network before reaching full battery capacity because of spending the entire budget. However, in this specific scenario, electric network had enough resources to charge all vehicles including those with lower budgets. In this experiment, UC strategy parameter κ was set to 1.

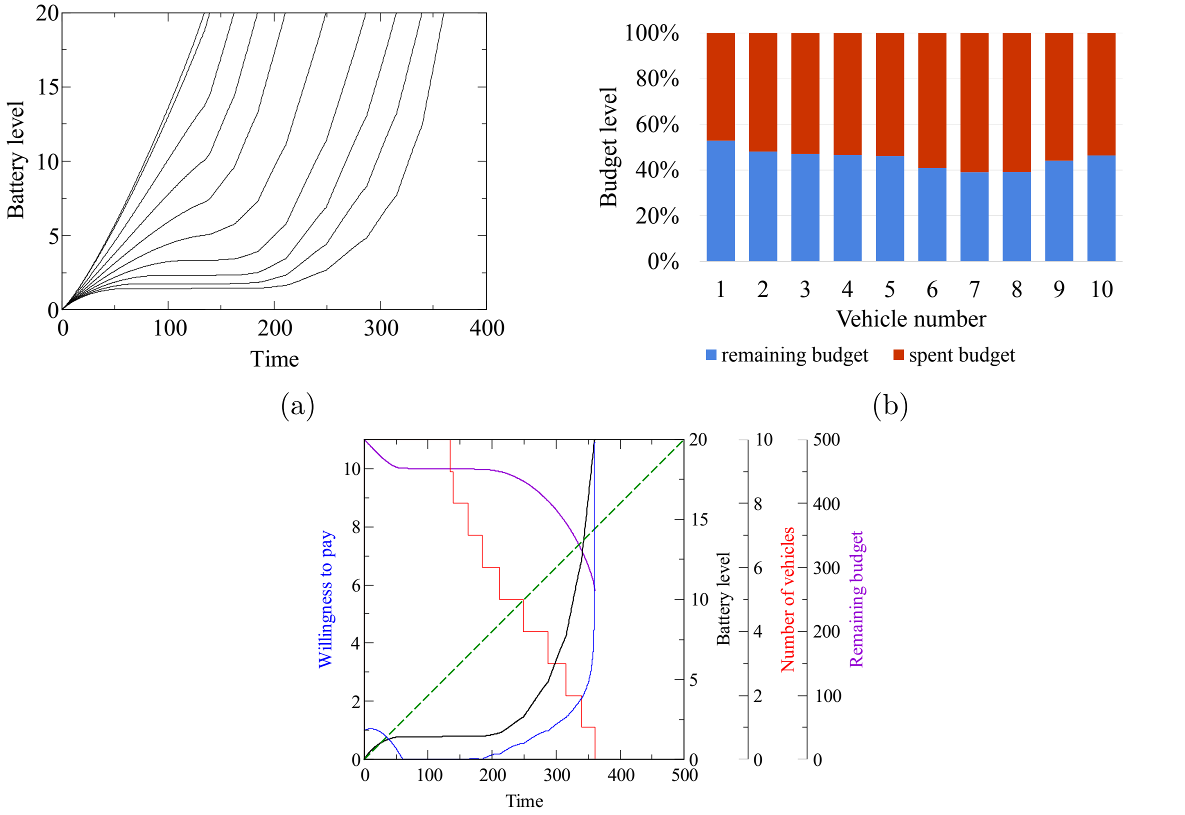


(a) (b)

Figure 6.12: UC strategy - performance in budget variation scenario. Figure (6.12a) shows charging process of all vehicles. Figure (6.12b) shows budget of individual vehicles.

AP strategy and AUT strategy

Behaviour of vehicles represented by AP and AUT strategies was identical, therefore all statements in this section apply to both strategies. Vehicles represented by AP and AUT strategy were charged in the same order as in the case of uniform spending in time strategy. Even though the order is the same, there is one distinct behavioural difference between the AP and AUT strategies and uniform spending in time strategy. Charging process of vehicles with lower budgets represented by AP and AUT strategies is initially slower than in the case of the corresponding vehicles represented by UT strategy. Charging process is initially slower because vehicles with lower budgets perceive price per unit of energy as high, thus they gradually lower their willingness to pay to w_{min} set to 0.01 and wait for the price per unit for energy to decrease. When vehicles with higher budgets leave the network, price decreases, which causes that some vehicles previously paying w_{min} start spending more money. This kind of behaviour can be interpreted as a scheduling effect, which was not created by a central authority controlling order in which vehicles charge their batteries, but was rather caused by a group of vehicles following the same charging algorithm. Charging process of vehicles in this experiment showed, that behaviour of individual vehicles may lead to interesting behavioural patterns, which are not enforced explicitly. Charging process of all connected vehicles is shown in the figure (6.13a), figure (6.13c) shows behaviour of the user with the lowest budget. Parameter $\kappa = 0.001$ and delay parameter d of the *AUT strategy* was set to 0.5.



(a) (b)

(c)

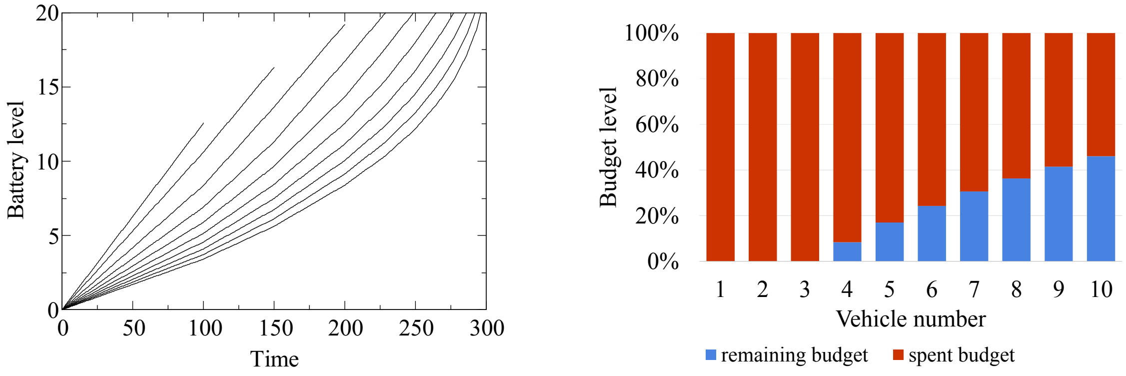
Figure 6.13: AP and AUT strategies - performance in budget variation scenario. Figure (6.13a) shows charging process of all vehicles. Figure (6.13b) shows budget of individual vehicles. Figure (6.13c) shows a charging process of a vehicle with the lowest budget.

6.3.2 Maximum charging time variation experiment

Purpose of this simulation scenario is to show behaviour of strategies when vehicles with various maximum charging times T^l_{max} are present in the network. All vehicles have the same initial budget $W_{max} = 1000$, their $B_{max} = 20$, exactly one vehicle is connected to nodes $i = 2, \dots 11$, vehicle l is connected to node $i = l+1$, for $l = 1, \dots, 10$. Maximum charging time $T^l_{max} = (l-1)50 + 100$, for $l = 1, \dots, 10$.

Uniform spending in time strategy (UT strategy)

Charging process of vehicles with lower T^l_{max} was faster than charging process of vehicles with higher T^l_{max}. Since all vehicles pay $\frac{1000}{T^l_{max}}$, vehicles with lower T^l_{max} can afford to pay more money per unit of time. Maximum charging time T^l_{max}, for $l = 1, 2, 3$ is too short, thus vehicles $1, 2, 3$ leave the network with battery state $B(T^l_{max}) < B^l_{max}$ and with zero remaining budget. This behaviour can be seen in the figure (6.14).

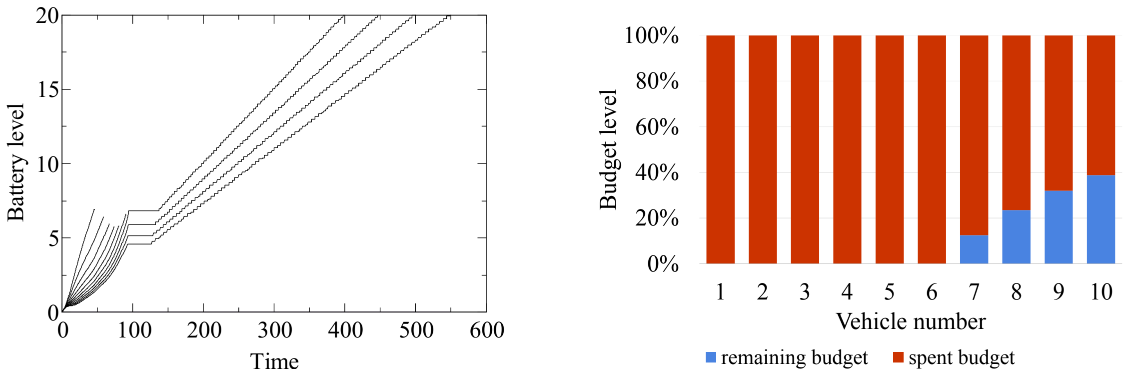


(a) (b)

Figure 6.14: UT strategy - performance in T^l_{max} variation scenario. Figure (6.14a) shows charging process of all vehicles. Figure (6.14b) shows budget of individual vehicles.

Uniform charging in time strategy (UC strategy)

Users $l = 1, \ldots, 10$ represented by UC strategy were all trying to ensure that their battery state $B^l(t)$ after being connected for t^l_c is equal to $\frac{t^l_c}{T^l_{max}} B_{max}$. Since all vehicles were using the same strategy, they all started to compete against each other, which caused that 6 vehicles had to disconnect from the network because their budget was insufficient for competing against the rest of vehicles. Vehicles 7,8,9 and 10 reached maximum battery capacity. Described behaviour can be seen in the figure (6.15), parameter κ of the UC strategy was set to 1.

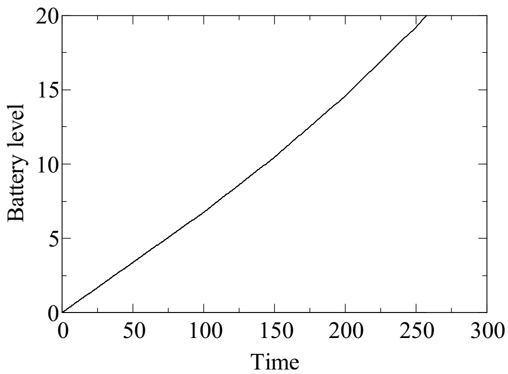


(a) (b)

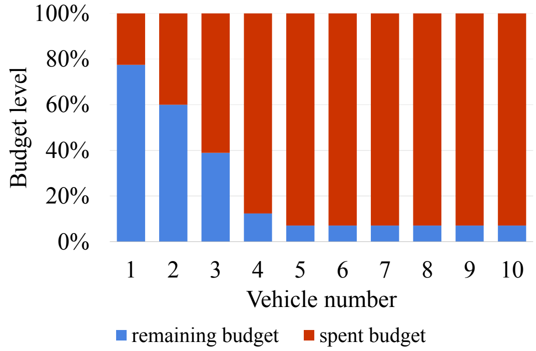
Figure 6.15: UC strategy - performance in T^l_{max} variation scenario. Figure (6.15a) shows charging process of all vehicles. Figure (6.15b) shows budget of individual vehicles.

Affordable price spending strategy (AP strategy)

When making a decision about updating willingness to pay $w_l(t)$, affordable price spending strategy does not use information about the time left in the network $T^l_{max} - t^l_c$. Since all vehicles $l = 1, \ldots, 10$ had the same initial budget, the charging process was the same for all connected vehicles and therefore it is not possible to distinguish individual users from each other in the figure (6.16a). Figure (6.16a) shows that vehicles reached full battery state after time 250, therefore vehicles l = 1,2,3,4 with $T^l_{max} \leq 250$ had to disconnect before reaching 100% battery capacity, vehicles $l = 5,\ldots,10$ had the same charging process, thus left the network with the same remaining budget $W^l_r(t)$. Parameter $\kappa = 0.001$ and $w_{min} = 0.01$.



(a)

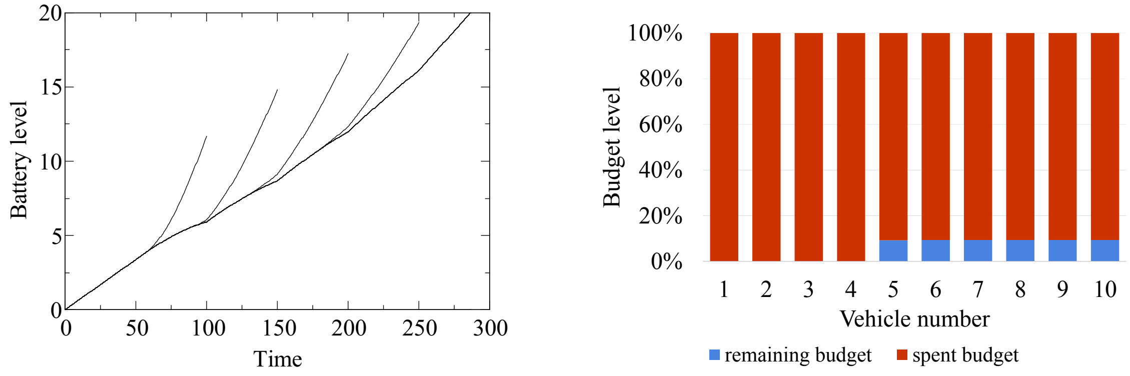


(b)

Figure 6.16: AP strategy - performance in T^l_{max} variation scenario. Figure (6.16a) shows charging process of all vehicles. Figure (6.16b) shows budget of individual vehicles.

Combination of affordable spending and uniform per time spending strategy (AUT strategy)

In this simulation scenario, there is a notable difference between AP and the AUT strategy, because AUT strategy uses information about remaining time in the network $T^l_{max} - t^l_c$ when making a decision about updating willingness to pay. All vehicles $l = 1,\ldots,10$ had a parameter $d^l = 0.5$, which means that when $t^l_c > 0.5T^l_{max}$, AUT strategy starts considering paying more money in order to gain more electric energy. T_{max} awareness of this strategy allowed vehicles l = 1,2,3,4 to gain more electric power, than corresponding vehicles represented by AP strategy. In this scenario, users represented by AP strategy reached an average battery state of only 17.1002, while average battery state of AUT users was 18.3027. Parameter $\kappa = 0.001$ and $w_{min} = 0.01$.



(a) (b)

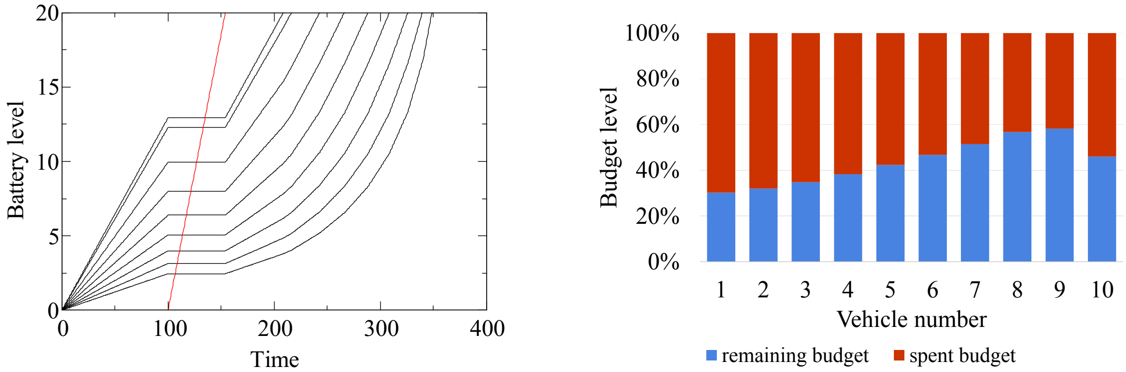
Figure 6.17: AUT strategy - performance in T^l_{max} variation scenario. Figure (6.17a) shows charging process of all vehicles. Figure (6.17b) shows budget of individual vehicles.

6.3.3 Arrival of "aggressive spender"

In this simulation scenario, exactly one vehicle was connected to nodes $2, \ldots, 11$, with $B_{max} = 20, T_{max} = 500, B(0) = 0$, but with different budgets. Vehicle l was connected to node $i = l + 1$, for $l = 1, \ldots, 10$, $W^l_{max} = 1.3^{l-1}500$, for $l = 1, \ldots 9$. Arrival time of vehicles $1, \ldots, 9$ is 0. Vehicle 10 is not in the network in the start of the scenario, it arrives at time $t = 100$, and is represented by a constant spending in time strategy with spending budget equal to 10000000 and $T^{10}_{max} = 100$. Purpose of this experiment is to show how individual strategies with ongoing charging processes behave after arrival of a vehicle that spends huge amount of money. Since the "aggressive spender" will consume majority of the resources of the electric network, this experiment also models a situation, when electric network goes suddenly into congested state, in which demand of individual vehicles cannot be fulfilled. Battery state evolution of the "aggressive spender" will be shown in red color.

Uniform spending in time strategy (UT strategy)

Vehicles represented by UT strategy were all able to reach full battery capacity. After arrival of the "aggressive spender", charging process of initially connected vehicles slowed down dramatically. In spite of the slower charging process, electric network provided enough electric energy to charge all vehicles in $T_{max} = 500$.

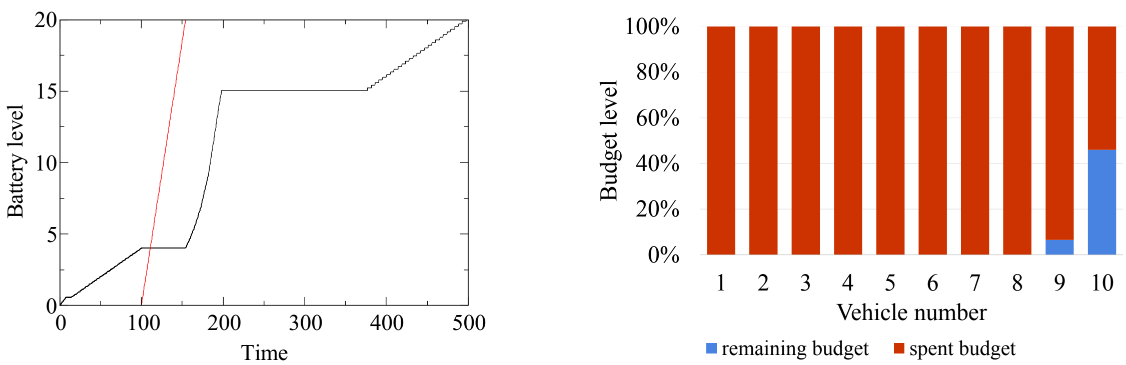


(a) (b)

Figure 6.18: UT strategy - performance in arrival of "aggressive spender" scenario. Figure (6.18a) shows charging process of all vehicles. Figure (6.18b) shows budget of individual vehicles.

Uniform charging in time strategy (UC strategy)

Out of all initially connected vehicles represented by UC strategy, only vehicle 9 reached full battery state. After arrival of "aggressive spender", other vehicles started to compete for electric power by increasing their willingness to pay, because their charging processes did not follow a linear charging function. Competing for electric power among vehicles caused that vehicles $1, \ldots, 8$ spent their budgets W^l_{max} before reaching maximum battery capacity. Since UC strategy does not use remaining budget when making a decision about updating willingness to pay, all vehicles had the same charging process and therefore the figure (6.19a) shows only one battery state curve for initially connected vehicles. Parameter κ of UC strategy was set to 1.

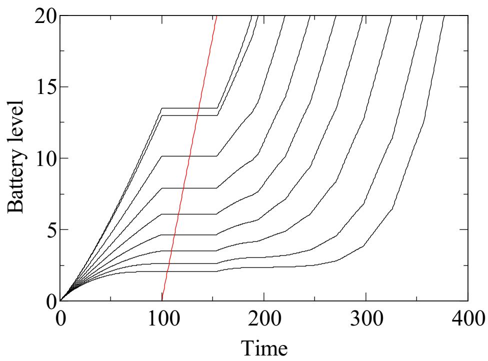


(a) (b)

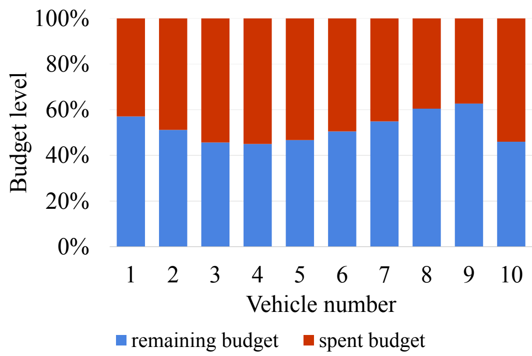
Figure 6.19: UC strategy - performance in arrival of "aggressive spender" scenario. Figure (6.19a) shows charging process of all vehicles. Figure (6.19b) shows budget of individual vehicles.

AP strategy and AUT strategy

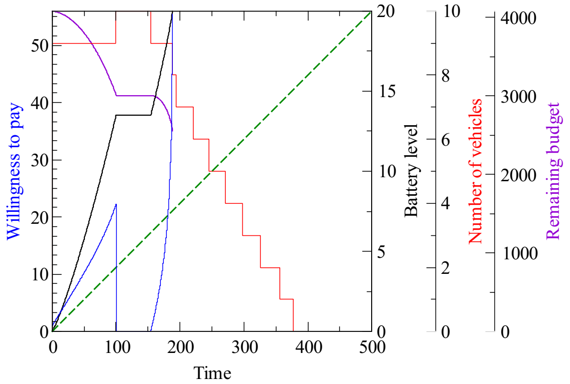
Similarly to the budget variation scenario, behaviour of AP and AUT strategies in arrival of "aggressive spender" scenario was identical. In the experiment $\kappa = 0.001$, $w_{min} = 0.01$, parameter d of AUT strategy was set to 0.5. All vehicles reached full battery state at the end of this scenario. After arrival of the "aggressive spender", all strategy instances detected that the price per unit of energy is too high, which resulted in lowering willingness to pay to the value w_{min}. This can be seen in the figure (6.20c) depicting a charging process of the initially connected vehicle with the highest spending budget. After the "aggressive spender" left the network, the price per unit of energy decreased to the level, which was affordable, therefore the individual strategy instances started increasing their willingness to pay. Figure (6.20b) shows, that in this scenario, vehicles represented by AP and AUT strategies managed to save more money than corresponding vehicles represented by UT and UC strategies.



(a)



(b)



(c)

Figure 6.20: AP and AUT strategies - performance in arrival of "aggressive spender" scenario. Figure (6.20a) shows charging process of all vehicles. Figure (6.20b) shows budget of individual vehicles. Figure (6.20c) shows a detailed charging process of vehicle with the highest budget.

Scenario evaluation

After arrival of "aggressive spender", vehicles represented by AP strategy and AUT strategy lowered the price they were paying to w_{min} and saved some money, in this situation this behaviour was very reasonable since it was basically impossible to compete against the "aggressive spender", which consumed most of the network resources. Users represented by UT strategy did not respond to the arrival of new vehicle at all and UC strategy instance started to compete against the "aggressive spender" by increasing willingness to pay, which is very inefficient in these situations.

System performance could also benefit from lowering willingness to pay to w_{min}, because vehicles do not compete against each other, but rather cooperate with each other (while not having information about the state of network at all), which may result in more efficient performance of the system. Excessive competition could lead to the situation when most of the vehicles cannot be charged fully as shown in section (6.3.3).

6.4 Performance of charging strategies

To study and compare performance of charging strategies, simulation experiments with random arrival of vehicles and multiple replications were run. Purpose of the experiments was to collect important statistics that would give us information about performance of charging strategies as a whole group, not as individuals. Statistics presented in this section were obtained by running 30 replications with maximum simulation time equal to 10000 with three different average inter-arrival times between two vehicles set to $16.6, 25$ and 50, with simulation step $\Delta t = 1$, and empty electric network in the beginning of each replication. Vehicles arrive with empty battery $B(0) = 0$, initial willingness to pay is set to 1 (except for vehicles represented by UT strategy), maximum battery capacity $B_{max} = 20$, limited charging time $T_{max} = 500$ and a limited spending budget $W_{max} = 1000$. Limit T_{max} was also applied to the static user strategy to make it fully comparable with inflexible and flexible user strategies. Electric network used in all experiments is visualized in the figure (6.2), each line e_{ij} has resistance $r_{ij} = 0.1$ and reactance $x_{ij} = 0.6$, nominal voltage $V_{nominal} = 1$ and parameter α in set of constraints (2.14) is set to 0.1. Node denoted as 0 is the source of electric power.

6.4.1 Already existing strategies - performance comparison

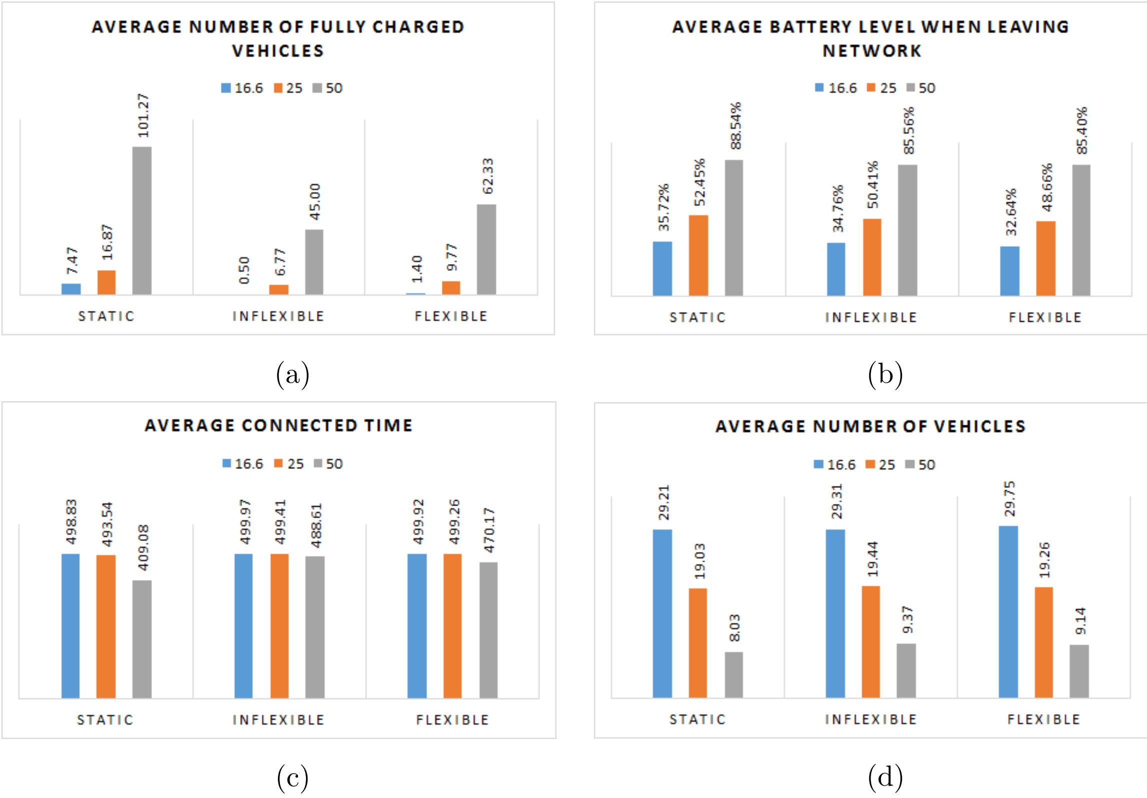
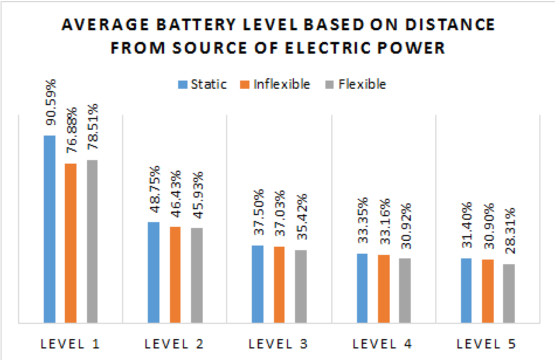


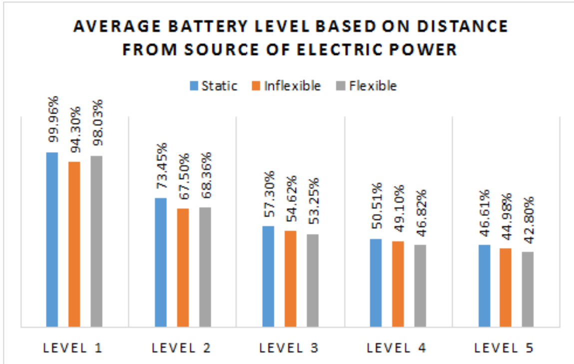
Figure 6.21: Performance comparison of strategies presented in [1] with average inter-arrival times set to 16.6, 25 and 50. Figure (6.21a) shows average number of vehicles, figure (6.21b) average battery state, figure (6.21c) average connected time and figure (6.21d) shows average number of vehicles present in the electric network.

Observed statistics in the figure (6.21) show that the Static user strategy managed to fully charge the most vehicles out of all strategies in all tested settings, it also has the lowest average connected time. Advantage of the *static user* lies in cooperation with the electric network. Static users do not compete against each other when the network is congested and since they keep their willingness to pay constant, they can efficiently take advantage of situation when electric network has enough resources to charge them before T_{max}. With decreasing inter-arrival time, less vehicles represented by *inflexible user strategy* are able to follow linear charging function, which results in lower average battery state when leaving the network and lower average number of fully charged vehicles compared to static users. Since inflexible and flexible users generally reach full battery state close to connected time $t_c^l = T_{max}^l$, they might stay in the network longer than necessary increasing the average number of vehicles present in the system, which can be seen in the figure (6.21d). Increase in number of vehicles in the network and following linear charging trajectory may result in preventing other vehicles from reaching full battery capacity B_{max}^l.

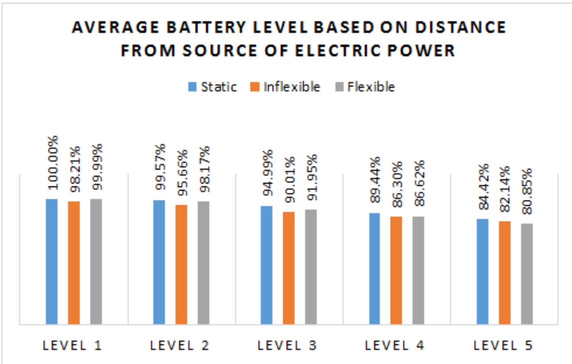
Average battery level comparison based on distance from the source of electric power



(a)



(b)

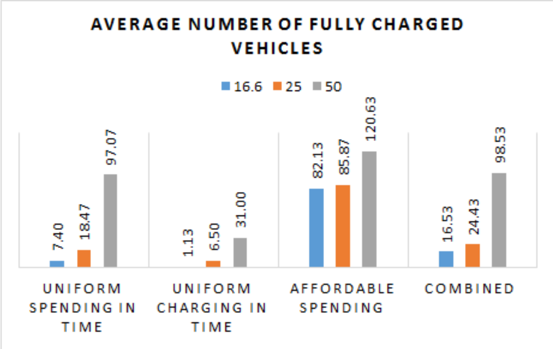


(c)

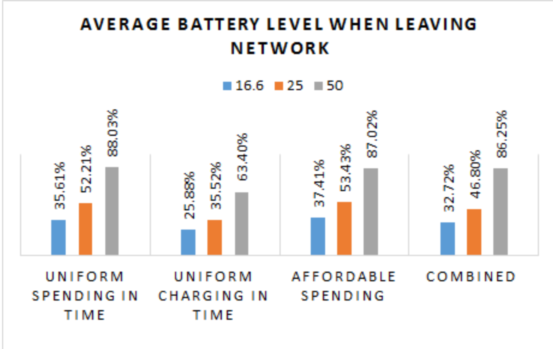
Figure 6.22: Comparison of average battery level on various levels of electric network for strategies from [1] with average inter-arrival time set to 16.6, 25 and 50. Figure (6.22a) shows average battery state for inter-arrival time set to 16.6, (6.22b) for inter-arrival time set to 25, (6.22c) for inter-arrival time set to 50.

Since electric network used in this experiment has multiple levels a network effect documented in section (6.1) is present. This means that average battery state of vehicles also depends on the distance from the source of electric power. Based on the figure (6.22), it can be generally assumed that the average battery state decreases with the increasing distance from the source of electric power and decreasing inter-arrival time. In these specific simulation experiments, *static user strategy* outperformed both *inflexible* and *flexible user strategies.*

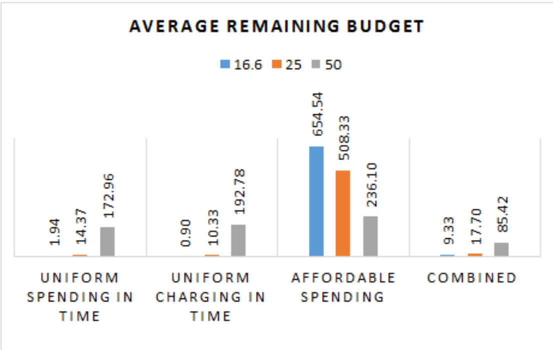
6.4.2 Strategies with budget - performance comparison



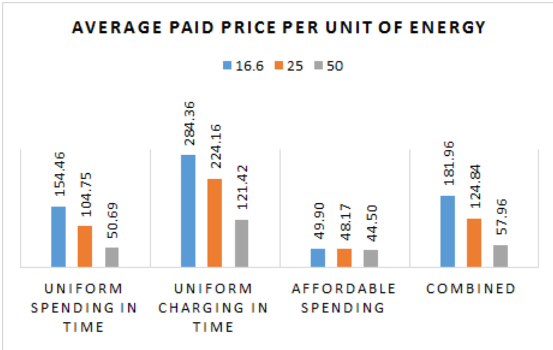
(a)



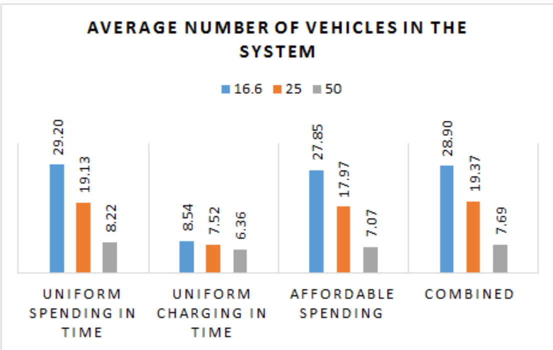
(b)



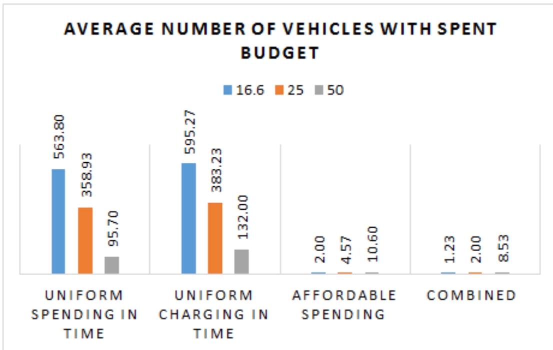
(c)



(d)



(e)



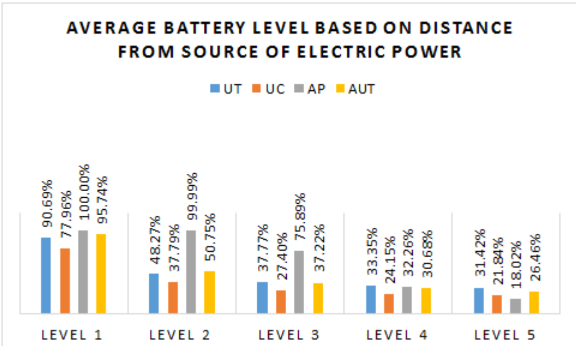
(f)

Figure 6.23: Comparison of observed statistics for budget based strategies with inter-arrival time set to 16.6, 25 and 50. Figure (6.23a) shows average number of fully charged vehicles, figure (6.23b) shows average battery state, figure (6.23c) shows average remaining budget, figure (6.23d) shows average paid price per unit of energy, figure (6.23e) shows average paid price per unit of energy, figure (6.23f) shows average number of vehicles, which spent their budget entirely.

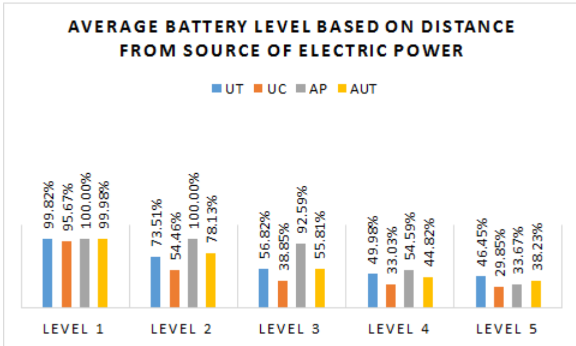
Collected statistics in the figure (6.23) show that, the AP strategy managed to fully charge the biggest number of vehicles in all experiments, this is not surprising because the simulation scenarios with varying budgets and arrival of "aggressive spender" showed that the AP strategy may decrease its willingness to pay to w_{min} (set to 0.01), when the price per unit of power is considered expensive. When vehicles which cannot afford to pay for electric power decrease their willingness to pay to w_{min}, charging process of vehicles which can afford to pay more can be faster. Even though AP strategy managed to charge fully the biggest number of vehicles, average battery level for AP strategy is very close to the UT strategy. UT strategy does not change its willingness to pay during the charging process, therefore amount of electric power assigned to vehicles using UT strategy will be based on the state of the electric network and their position in the electric network. Even though AUT strategy is a T_{max}-aware extension of AP strategy, the time awareness caused that the observed statistics are different.

Very interesting observation is the raise in the average price per unit of energy in the figure (6.23d), where with decreasing inter-arrival time between vehicles the price per unit of energy increases considerably for all strategies, except for AP strategy, where only a minor raise of the price was recorded. This is caused by the fact that vehicles represented by AP strategy choose the price they pay very cautiously based on the price per unit of energy and price they can afford to pay. Another recorded difference among performance of strategies can be seen in the figure (6.23c), where average remaining budget decreases with decreasing inter-arrival time, except for AP strategy where the average remaining budget increased with decreasing inter-arrival time, this is again a result of very sensible consideration of the price to pay. Another interesting fact about behaviour of strategies is shown in the figure (6.23e), which shows average number of vehicles present in the system during simulation. Generally, with decreasing inter-arrival time between vehicles the average number of vehicles increases, however the UC strategy shows only a minor increase, which is caused by the way the strategy spends its budget. If the network is congested UC strategy starts paying too much money, which results in leaving the system because its budget was spent entirely. This is justified by figure (6.23f) showing average number of vehicles, which spent their budget entirely.

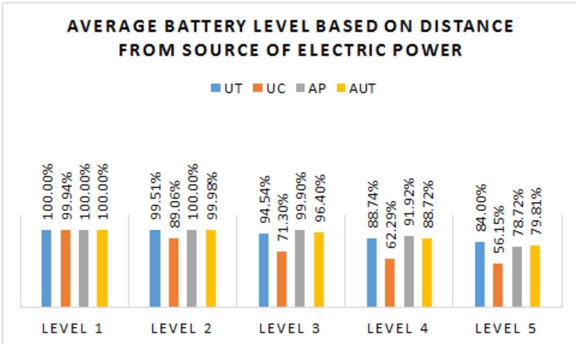
Average battery level comparison based on distance from the source of electric power



(a)



(b)



(c)

Figure 6.24: Comparison of average battery level on various levels of electric network for strategies with budget. Figure (6.22a) shows average battery state on various levels of electric network with inter-arrival time set to 16.6, (6.22b) with inter-arrival time set to 25 and (6.22c) with inter-arrival time set to 50.

Figure (6.24) shows average battery state of vehicles based on the level of the network. Generally, with increasing distance from the source of electric power, and decreasing inter-arrival time between vehicles the average battery level decreases, however separate electric vehicle charging strategies handle distance from the source of electric power differently. In the case of UC strategy, decrease in the average battery state can be seen on all levels of electric network. This is caused by spending too much while trying to follow linear charging function, which is not possible when the network is congested. Decrease in average battery state with decreasing inter-arrival time for AUT strategy is mostly visible on levels $2, \ldots, 5$ and there is only a minor change on the level 1. Very interesting is the case of AP strategy, which prioritizes vehicles closer to the source of electric power, however vehicles connected further from the source gain considerably less electric power. Performance of AP strategy can be considered as a specific case of slower is faster effect from [19]. Vehicles represented by AP strategy that are further from the source of electric power slow their charging processes down when the price per unit of energy is perceived as too high, which leads to increased number of fully charged vehicles and higher average battery state on nodes closer to the source of electric power. Even though AUT strategy is an extension of AP strategy, its performance in terms of average battery state based on the distance from the source of electric power is very different, because AUT strategy might spend more budget towards the end of the charging process. UT strategy does not alter its willingness to pay during the charging process, which basically means that the state of electric network and the position of the vehicle in the network determines the speed of the charging process and cannot be influenced.

7. User guide

To run implemented simulation program, Python version 3, CVXOPT and numpy libraries are required. The program is open source, it is operated from command line and its latest version can be downloaded from `https://bit.ly/2qO9CkX`. Program has three modes, the first mode is used for scaling experiments, which were used to evaluate model scaling strategies. This mode is not aimed at users, it was used for internal experiments, therefore code changes may be required in order to change parameters of scaling experiments. The second mode is used for simulation with random arrivals of electric vehicles. This mode is fully configurable, so no code changes are required in order to change parameters of simulation, only input parameters have to be changed. Third mode is used for running predefined simulation scenarios. Simulation experiments were run on *HPC cluster*. In order to run the program on *HPC cluster*, job starting scripts had to be created. All created scripts are included in the appendix.

7.1 Scaling experiments

To run scaling experiments from command line use command: *python[3] main.py -l number_of_replications -m max_time -r resistance_file -x reactance_file -e experiment_number*. Parameters of the program are:

-l/--replications: represents number of replications,

-m/--max_time: represents maximum simulation time in each replication,

-r/--resistance_file: represents path to the file containing resistance on edges of the network,

-x/--reactance_file: represents path to file containing reactance on edges of the network,

-e/--experiment_number: represents experiment number. Experiments to choose from are listed in section (7.1.1).

7.1.1 Supported scaling experiments

In scaling experiments, exactly one vehicle is connected to nodes $i \in \mathcal{N}^+$. Supported scaling experiments are:

Composite scaling strategy experiment

Experiment evaluates performance of composite scaling strategy. Initial willingness to pay is generated as described in section (5.2). Willingness to pay of each vehicle is updated by adding a pseudorandom number generated from specified interval with uniform distribution (table (7.1)). Composite scaling strategy experiment can be run using experiment number from the table (7.1) .

Experiment number	lower bound	upper bound
0	-1	1
1	-10	10
2	-100	100

Table 7.1: Composite scaling strategy experiment settings

Individual scaling strategies comparison experiment

This experiment evaluates performance of all created scaling strategies. Initial willingness to pay of the connected vehicles and the way how willingness to pay is updated is the same as in the previous experiment. Unlike the previous experiment, which uses composite scaling for solving created optimization problems, this experiment tests each individual scaling strategy on each optimization problem separately.

Experiment number	lower bound	upper bound
3	-1	1
4	-10	10
5	-100	100

Table 7.2: Individual scaling strategies comparison experiment settings

All presented scaling experiments have a console output.

7.2 Simulation experiments

To run the simulation mode from command line use: *python[3] chargingMain.py -s simulation_step -l number_of_replications -m max_time -r resistance_file -x reactance_file -i inter_arrival_time -d simulation_directory -g generator directory.* Parameters of the program are:

-s/–time_step length of the time step used by the continuous simulation core,

-l/–replications: number of replications,

-m/–max_time: maximum simulation time,

-r/–resistance_file: path to the file containing resistance on edges of the network,

-x/–reactance_file: path to file containing reactance on edges of the network,

-i/–inter_arrival_time: average time between arrivals of two electric vehicles,

-d/–simulation_directory: name of the directory to store simulation files,

-g/–generator_directory: path to the file where charging strategy generator settings are located.

Simulation experiments run in two distinct modes: The first mode is called *detailed mode*. The detailed mode is triggered automatically when simulation is run with only one replication. Detailed mode creates a log file with information about all simulation events, file containing car number history and file with all statistics mentioned in (6.4). Detailed mode also creates files containing data of vehicles that were present in the simulation experiment, these information include: basic information about the vehicle, its battery evolution, willingness to pay evolution and budget evolution for strategies with budget. Created files enabled us to study each vehicle individually, the log file was very useful for debugging purposes and verification of the implemented simulation model.

The second mode is a *replication mode*, it is triggered automatically for experiments with more than one replication. In this case simulation offers only statistics that were presented in section (6.4). Reason why this mode does not contain detailed information about vehicles is that with increasing number of replications and decreasing inter-arrival time between vehicles, the simulation directory could consume a huge amount of disk space.

7.3 Simulation scenarios

To run implemented simulation scenarios from command line use: *python[3] non-RandomChargingMain.py -s simulation_step -m max_time -r resistance_file -x reactance_file -d simulation_directory -g generator_directory -n scenario_number*. Parameters of the program are:

-s/–time_step length of the time step used by the continuous simulation core,

-m/–max_time: maximum simulation time,

-r/–resistance_file: path to the file containing resistance on edges of the network,

-x/–reactance_file: path to file containing reactance on edges of the network,

-d/–simulation_directory: name of the directory to store simulation files,

-g/–generator_directory: path to the file where strategy generator settings are located,

-n/–scenario_number: scenario number to perform.

Simulation scenarios use the same simulation core as the simulation experiments with random arrivals of vehicles. Simulation scenarios are always run with only replication, which means that simulation scenarios run in the detailed mode.

7.3.1 Currently implemented scenarios

0: Budget variation experiment: Simulation scenario described in section (6.3.1).

1: T^{l}_{max} variation experiment: Simulation scenario described in section (6.3.2).

2: Arrival of "aggressive spender": Simulation scenario described in section (6.3.3).

7.4 Simulation input files

This section provides information about input files of the program and their structure.

7.4.1 Entering electric network

Electric network that is used in experiments is specified in two separate files: file with resistance and file with reactance on the edges of electric network. Both files are specified in a .csv format. For example, if the user wants to create a simulation experiment with electric network presented in the figure (7.1),

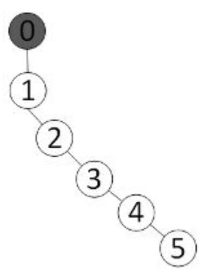


Figure 7.1: Sample structure of electric network

structure of resistance and reactance files, respectively, is:

$$\text{resistance:}\begin{matrix} 0, & r_{01}, & 0, & 0, & 0, & 0 \\ r_{10}, & 0, & r_{12}, & 0, & 0, & 0 \\ 0, & r_{21} & 0, & r_{23}, & 0, & 0 \\ 0, & 0, & r_{32}, & 0, & r_{34}, & 0 \\ 0, & 0, & 0, & r_{43}, & 0, & r_{45} \\ 0, & 0, & 0, & 0, & r_{54}, & 0 \end{matrix} \quad \text{reactance:}\begin{matrix} 0, & x_{01}, & 0, & 0, & 0, & 0 \\ x_{10}, & 0, & x_{12}, & 0, & 0, & 0 \\ 0, & x_{21} & 0, & x_{23}, & 0, & 0 \\ 0, & 0, & x_{32}, & 0, & x_{34}, & 0 \\ 0, & 0, & 0, & x_{43}, & 0, & x_{45} \\ 0, & 0, & 0, & 0, & x_{54}, & 0 \end{matrix}.$$

Value 0 in row i and column j of the resistance or reactance file implies that node i is not adjacent to node j. Value other than zero means, that node i is adjacent to node j (and vice versa), and it specifies resistance/reactance on edge e_{ij}. Program assumes that $r_{ij} = r_{ji}, x_{ij} = x_{ji}$, for $i, j \in \mathcal{V}, i \neq j$. Node 0 is the source of electric power. Structure of the electric network input file was chosen in order to maintain compatibility with already existing Matlab experiments.

7.4.2 Car type generator settings

In the simulation experiments, electric network is populated with instances of predefined electric vehicle charging strategies. Using predefined electric vehicle charging strategies requires minor code changes when adding a new strategy, however this is more user-friendly, because the user does not have to specify all parameters of charging strategies manually. The user only needs to specify the name of a specific instance of charging strategy that should be used in experiments. Settings of the population of electric vehicle strategies used in the experiment are specified in a file .csv file with the following format: *strategy_type_name,probability*. Each line of the file contains strategy name and the probability of generating this strategy. Sum of probabilities in all lines has to be equal to 1. Currently supported strategies to choose from are:

Name	Strategy	B_{max}	T_{max}	Budget	$w(0)$	κ
static_user	Static user	10	X	X	1	X
static_user	Static user	20	X	X	1	X
static_user_limited_time	Static user	20	500	X	1	X
inflexible_user	Inflexible user	10	500	X	1	1
inflexible_user_2	Inflexible user	20	500	X	1	1
flexible_user	Inflexible user	10	500	X	1	1
flexible_user_2	Inflexible user	20	500	X	1	1

Table 7.3: Settings for creating predefined strategies presented in [1]

To create a budget based strategy with $B_{max} = 20, T_{max} = 500$ use:

Name	Strategy	Budget	$w(0)$	w_{min}	κ	d
constant_spender_1	UT	500	1	X	X	X
constant_spender_2	UT	1000	2	X	X	X
constant_spender_3	UT	1500	3	X	X	X
constant_spender_4	UT	2000	4	X	X	X
constant_energy_gainer	UC	1000	1	X	1	X
affordable_price_spender	AP	2000	1	0.01	0.001	X
affordable_price_spender_2	AP	2000	1	0.01	0.001	X
combined	AUT	1000	1	0.01	0.001	0.75
combined_2	AUT	1000	1	0.01	0.001	0.5

Table 7.4: Settings for creating predefined budget based strategies

7.5 Automated creation of graphs from simulation experiments

When simulation is run in the detailed mode, information about electric vehicles including budget evolution, willingness to pay evolution and battery state evolution are stored in files created for each individual vehicle. To simplify analysis of experiments, a tool called *Grace graph generator* was created. Purpose of the tool is to automatically create graphs from collected data for each individual vehicle.

In order to run the *Grace graph generator*, the user has to install a program called *QtGrace* and the *Qtgrace* executable file has to be part of system *Path* variable. *Grace graph generator* is operated from command line. To create PDF graphs from simulation files run the command:
java -jar graceGraphGenerator-1.0-SNAPSHOT-jar-with-dependencies.jar - g "simulationDirectory" - f "PDF" - d "dirName",
for JPG graphs use:
java - jar graceGraphGenerator-1.0-SNAPSHOT-jar-with-dependencies.jar - g "simulationDirectory" - f "JPG" - d "dirName". Parameter *simulationDirectory* is a path where the simulation files are stored and parameter *dirName* is name of the directory in which the created graphs will be stored. When specifying the *dirName* make sure that the directory with the same name does not already exist.

8. Conclusions

In conclusion, I believe that this thesis fulfilled all goals stated in the section (1.5). Behaviour of three already existing electric vehicle charging strategies from [1] was analyzed, based on the analysis, four new electric vehicle charging strategies with a limited spending budget inspired by [11] were introduced. Experiments with strategies were performed using a simulation tool, which was created in this thesis. Core of the simulation tool is a convex optimization model, optimal solution of this model is used for assigning electric power to electric vehicles at specific simulation time t. To limit impact of the changes in occupancy of electric network on the structure of optimization problem, a minor alteration of the convex optimization model was presented.

At first, vehicles represented by charging strategies were analyzed individually to study their typical behaviour, then the analysis focused on behaviour of small groups of vehicles in predefined situations called simulation scenarios. Purpose of the simulation scenarios was to study performance of strategies in various situations and observe whether group of vehicles following the same charging strategy creates interesting behavioural patterns. Finally, charging strategies were tested in simulation experiments with multiple replications in order to evaluate performance of the strategies in the electric network. Analysis of individual instances of electric vehicle charging strategies was facilitated by a tool called *Grace graph generator*, which was developed in this thesis. *Grace graph generator* is able to create QtGrace based graphs depicting behaviour of individual users by aggregating vehicle specific files created in simulation experiments.

8.1 Possible future goals

Strategies and the experiments presented in this thesis form a solid foundation of a research in the area of charging strategies, however this thesis can still be considerably extended. Implemented simulation tool supports analysis of electric charging strategies in more depth than presented here. For instance, only a small portion of all possible parameters of the presented charging strategies was tested, so the created strategies can be be further researched, since their behaviour can be altered using available parameters. Another aspect of this thesis that could be extended, is to perform simulation experiments with multiple electric networks.

Further research of this topic could also include design and implementation of new electric vehicle charging strategies and testing their performance against the strategies presented in this thesis. Apart from creating new strategies, new simulation scenarios and new types of experiments could be proposed in order to test performance of strategies in new situations.

This thesis presented theoretical principles behind charging strategies and their performance in electric network, however simulation time, maximum battery capacity, willingness to pay, maximum budget and other units presented here were not represented by real physical units. In order to support the research in the field of user strategies a relation between presented non-physical units and real world physical units should be found in order to fully apply research presented in this thesis to real world electric networks.

Bibliography

[1] Jonas Knapp. Mathematical Modelling of Electric Vehicle Charging, 2016.

[2] Zachary Karabell. This week's stock market drop was machine-made. the freakout that followed was man-made. - the washington post. `https://wapo.st/2HbJ9FL`. (Cited on 5.4.2018).

[3] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, New York, NY, USA, 2004.

[4] H. Luss. *Equitable resources allocation: Models, Algorithms and Applications*. John Wiley & Sons, 2012.

[5] Peter Márton and Norbert Adamko. *Praktický úvod do modelovania a simulácie*. Žilinská univerzita, 2011.

[6] Antonín Kavička, Valent Klima, and Norbert Adamko. *Agentovo orientovaná simulácia dopravných uzlov*. Žilinská univerzita, 2005.

[7] Joshua Adam Taylor. *Convex Optimization of Power Systems*. Cambridge University Press, 2015.

[8] William H. Kersting. *Distribution System Modeling and Analysis*. CRC Press, 2001.

[9] Steven Low. Convex Relaxation of Optimal Power Flow—Part I: Formulations and Equivalence. *IEEE Transactions on Control of Network Systems*, 1(1):15–27, 2014.

[10] Steven Low. Convex Relaxation of Optimal Power Flow - Part II: Exactness. *IEEE Transactions on Control of Network Systems*, 1(2):177–189, 2014.

[11] R.J. Gibbens and F.P. Kelly. Resource pricing and the evolution of congestion control. *Automatica*, 35(12):1969–1985, 1999.

[12] Joptimizer - java convex optimizer. `http://www.joptimizer.com/`. (Cited on 21.9.2017).

[13] Cvxopt. `http://cvxopt.org/`. (Cited on 21.9.2017).

[14] Cvxpy. `http://www.cvxpy.org/en/latest/#`. (Cited on 21.9.2017).

[15] Gurobi optimization - the state of the art mathematical programming solver. `http://www.gurobi.com/`. (Cited on 21.9.2017).

[16] Mosek. https://www.mosek.com/. (Cited on 21.9.2017).

[17] Nonlinear convex optimization — cvxopt user's guide. http://cvxopt.org/userguide/solvers.html. (Cited on 03.12.2017).

[18] Cvxopt – google group. https://groups.google.com/forum/#!forum/cvxopt. (Cited on 24.11.2017).

[19] Carlos Gershenson and Dirk Helbing. When slower is faster. *Complexity*, 21(2):9–15, 2015.

Appendix

- APPENDIX 1 - DVD with:
  - Source code of the simulation tool
  - Source code of *Grace graph generator*
  - Executable .jar file of *Grace graph generator*
  - Results of all presented experiments
  - HPC cluster scripts
  - Diploma thesis in PDF format
  - Mathematical Modelling of Electric Vehicle Charging report by Jonas Knapp in PDF format