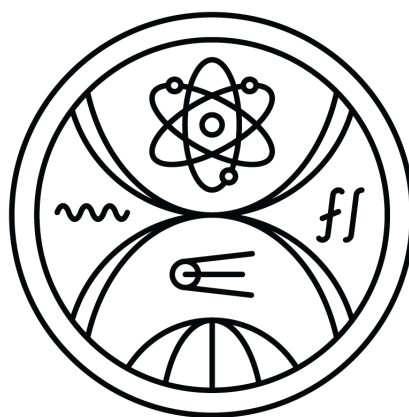


UNIVERZITA KOMENSKÉHO V BRATISLAVE
FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY

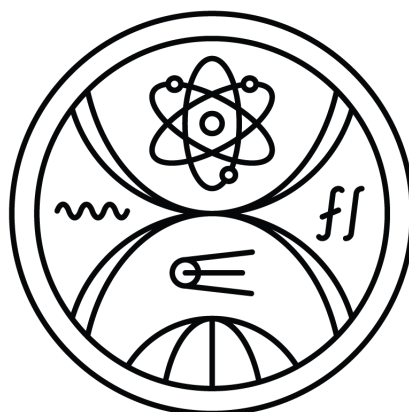


PODPORA INTELIGENTNÉHO RIADENIA
ENERGETICKÝCH SIETÍ
DIPLOMOVÁ PRÁCA

2023

BC. OMAR AL-SHAFE'I

UNIVERZITA KOMENSKÉHO V BRATISLAVE
FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY



PODPORA INTELIGENTNÉHO RIADENIA ENERGETICKÝCH SIETÍ

DIPLOMOVÁ PRÁCA

Študijný program:	Informatika
Študijný odbor:	Aplikovaná informatika
Školiace pracovisko:	Katedra aplikovanej informatiky
Školiteľ:	prof. RNDr. Mária Lucká, PhD.

Bratislava, 2023

Bc. Omar Al-Shafe'i



Univerzita Komenského v Bratislave
Fakulta matematiky, fyziky a informatiky

ZADANIE ZÁVEREČNEJ PRÁCE

Meno a priezvisko študenta: Bc. Omar Al-Shafe'i
Študijný program: aplikovaná informatika (Jednoodborové štúdium, magisterský II. st., denná forma)
Študijný odbor: informatika
Typ záverečnej práce: diplomová
Jazyk záverečnej práce: slovenský
Sekundárny jazyk: anglický

Názov: Podpora inteligentného riadenia energetických sietí
Support for intelligent management of smart energy networks

Anotácia: Stabilita inteligentnej energetickej siete a zabezpečenie dodávok elektrickej energie môže byť potenciálne narušená integráciou obnoviteľných zdrojov energie (napr. fotovoltických panelov), veľkých úložísk energie (batérií), ale aj významným rozvojom elektromobility. Tieto nové prvky sa v súčasnosti v čoraz väčšej miere stávajú súčasťou moderných inteligentných energetických sietí a preto ich integrácia do celej sústavy sa stala mimoriadne dôležitou. Pôvodne jednosmerná sieť, kde sa energia od veľkých výrobcov elektriny dodávala spotrebiteľom, sa postupne mení na obojsmernú sieť, v ktorej sa mnohí odberatelia - vďaka obnoviteľným zdrojom - stávajú súčasne aj drobnými výrobcami – prosumeri. Kvôli garancii stability celej sústavy a zabezpečeniu minimálnej ceny bolo potrebné vytvoriť tzv. agregátorov flexibility, ktorým účastníci siete môžu poskytnúť svoju flexibilitu. Pod flexibilitou pritom rozumieme práva (1) na odber vyrobenej alebo uskladnenej elektrickej energie (napr. z fotovoltiky, batérie alebo elektromobilu) v určitom čase a (2) obmedzenie spotreby (napr. vypnutie kúrenia). Navrhnete a overte model práce agregátora flexibility, ktorý vďaka inteligentným algoritmom a dátovej analýze dokáže optimalizovať tok energie, zabezpečiť stabilné dodávky energie a minimalizovať náklady spotrebiteľov. Svoje riešenie obmedzte pre vybraný typ prosumerov/spotrebiteľov. Použite pritom vhodné optimalizačné metódy, predovšetkým metódy strojového učenia. Svoje riešenie implementujte a porovnajte s existujúcimi riešeniami na dostupných dátach.

Literatúra:

1. Li, T., Sun, B., Chen, Y., Ye, Z., Low, S. H., & Wierman, A. (2020). Real-time Aggregate Flexibility via Reinforcement Learning. 1–17. <http://arxiv.org/abs/2012.11261>
2. Fernando Lezama, Joao Soares, Bruno Canizes, Zita Vale: Flexibility management model of home appliances to support DSO requests in smart grids, Sustainable Cities and Society, Volume 55, 2020, ISSN 2210-6707, <https://doi.org/10.1016/j.scs.2020.102048>
3. Tongxin Li and Bo Sun and Yue Chen and Zixin Ye and Steven H. Low and Adam Wierman: Learning-Based Predictive Control via Real-Time Aggregate Flexibility, IEEE Transactions on Smart Grid, 12 (6), 2021, 97–4913, <https://doi.org/10.1109%2Ftsg.2021.3094719>.



Univerzita Komenského v Bratislave
Fakulta matematiky, fyziky a informatiky

4. Steve Wattam: Artificial intelligence and machine learning approaches to energy demand-side response: A systematic review, Renewable and Sustainable Energy Reviews, Volume 130, 2020, 109899, ISSN 1364-0321, <https://doi.org/10.1016/j.rser.2020.109899>.

Vedúci: prof. RNDr. Mária Lucká, PhD.
Katedra: FMFI.KAI - Katedra aplikovanej informatiky
Vedúci katedry: doc. RNDr. Tatiana Jajcayová, PhD.
Dátum zadania: 13.12.2022

Dátum schválenia: 13.12.2022
prof. RNDr. Roman Ďurikovič, PhD.
garant študijného programu

.....
š student

.....
vedúci práce

Pod'akovanie: Chcel by som pod'akovať mojej školiteľke za cenné rady počas tvorby diplomovej práce.

Abstrakt

Slovenský abstrakt v rozsahu 100-500 slov, jeden odstavec. Abstrakt stručne sumarizuje výsledky práce. Mal by byť pochopiteľný pre bežného informatika. Nemal by teda využívať skratky, termíny alebo označenie zavedené v práci, okrem tých, ktoré sú všeobecne známe.

Kľúčové slová: jedno, druhé, tretie (prípadne štvrté, piate)

Abstract

Abstract in the English language (translation of the abstract in the Slovak language).

Keywords:

Obsah

Úvod	1
1 Východiská	3
1.1 Využité technológie	3
1.1.1 Balík ACN Portal.	3
1.1.2 Ukážka vstupných dát ACN-Data	5
1.1.3 Architektúra simulátora ACN-Sim	5
1.1.4 Balík Adacharge.	6
1.1.5 Balík Acnportal-experiments.	6
1.2 Aplikovateľnosť smart charging algoritmov.	6
1.3 Konfigurácia a obmedzenia nabíjacích sietí	7
1.3.1 Typy nabíjania.	7
1.3.2 Rýchlosť nabíjania batérie.	8
1.3.3 Typy nabíjacích staníc.	9
1.3.4 Obmedzenia pri nabíjaní.	9
1.4 Problém nájdenia optimálneho schedulingu.	9
1.5 Neriadené nabíjanie.	12
1.6 Algoritmus Round Robin.	12
1.7 Triediace scheduling algoritmy	13
1.7.1 Least Laxity First.	13
1.7.2 Smoothed Least Laxity First.	14
1.7.3 Algoritmus Enhanced Least Laxity First.	15
1.7.4 Algoritmus Earliest Deadline First.	15
1.7.5 Group Earliest Deadline First.	15
1.7.6 First-Come First-Served.	15
1.7.7 Last-Come First-Served.	15
1.7.8 Longest Remaining Processing Time.	15
1.7.9 Shortest Job Next.	16
1.8 Kontrolné scheduling algoritmy	16
1.8.1 Typy kontrolných algoritmov	16

1.8.2	Model Predictive Control.	17
1.8.3	Penalised Predictive Control.	17
1.9	Cena nabíjania.	18
2	Návrh riešenia	20
2.1	Least Laxity First	20
2.2	Smoothed Least Laxity First.	20
2.3	Riešenie problému prípustnosti.	20
2.4	Návrh nášeho MPC modelu.	20
3	Implementácia	21
4	Testovanie	22
	Záver	23
	Príloha A	26
	Príloha B	27

Zoznam obrázkov

1.1	Schéma ACN Portal.	4
1.2	Ukážka vstupných dát ACN-Data	5
1.3	Architektúra simulátora ACN-Sim.	5
1.4	Typy nabíjania elektromobilov	8
1.5	Proces schedulingu pre online scheduling algoritmy.	10
1.6	Vplyv vedomostí na výkonnosť scheduling algoritmu.	11
1.7	Príklad Round Robin algoritmu v praxi.	13
1.8	Model systému, využívajúci PPC.	18
1.9	Systém toku energie a ziskov.	19

Zoznam tabuliek

Úvod

S neustále pribúdajúcim množstvom elektrických vozidiel na trhu, pribúda aj množstvo elektrickej energie, ktorú treba dodať týmto vozidlám. Kapacity jednotlivých nabíjajúcich staníc nebudú musieť stačiť pre nabitie každého elektrického vozidla požadovaným množstvom energie. Postupne sa tiež odoberatelia elektrickej energie stávajú výrobcami elektrickej energie pomocou obnoviteľných zdrojov. Tým sa jednosmerná sieť, kde veľký výrobca dodáva energiu spotrebiteľom mení na obojsmernú sieť, kde odoberatelia, ktorí sú výrobcami elektrickej energie môžu poskytovať podporné služby.

Riešením týchto problémov sú inteligentné takzvané smart siete. Tieto siete obsahujú agregátor flexibility, ktorému používatelia nabíjacej siete poskytujú flexibilitu. To znamená, že agregátor flexibility má právo riadiť odber elektrickej energie pre elektrické vozidlá a prispôbovať ho podľa potrieb a podľa podmienok siete. Používatelia, ktorí sú tiež výrobcami elektrickej energie z obnoviteľných zdrojov môžu poskytnúť podporné služby prostredníctvom agregátora flexibility (takýchto používateľov nazývame prosumeri), tým že napríklad agregátor flexibility od nich kúpi energiu. Používatelia v takýchto sieťach môžu nastaviť svoje preferencie, ako napríklad, preferovaný čas nabíjania, množstvo požadovanej energie. Agregátor flexibility následne pomocou inteligentných algoritmov vie optimalizovať tok energie. Využíva pritom optimalizačné metódy, alebo metódy strojového učenia (napr. neurónové siete). Agregátor flexibility tiež minimalizuje cenu elektrickej energie pre spotrebiteľov tým, že nakupuje elektrickú energiu keď je najlacnejšia, a predáva elektrickú energiu keď je najdrahšia.

Cieľom tejto práce je overiť model práce agregátora flexibility, ktorý pomocou inteligentných algoritmov a dátovej analýzy vie optimalizovať tok energie, zabezpečiť stabilné dodávky energie a tiež minimalizovať náklady používateľov elektrických vozidiel. Porovnáme naše riešenia tohoto problému s inými, doterajšími riešeniami. [12]

V prvej kapitole rozoberieme východiská pri tvorbe nášho modelu agregátora flexibility, opíšeme existujúce riešenia a aj obmedzenia infraštruktúry pri nabíjaní elektrických vozidiel. V druhej kapitole predstavujeme model práce nášho agregátora flexibility, spomíname inteligentné algoritmy, ktoré sme v našom modeli použili.

V tretej kapitole overujeme model práce agregátora flexibility, kde porovnávame naše riešenie s existujúcimi riešeniami vzhľadom na pomer dodanej energie, množstvo dodanej energie, cenu dodanej energie a počet výmen vozidiel. Vstupné dáta o elektrických autách

získavame z nabíjajúcich staníc pre elektrické autá: stanica ACN a stanica JPL. Výstup našej implementácie je optimálny scheduling vozidiel v čase. V našej implementácii sa snažíme počet výmen vozidiel minimalizovať, keďže to často vedie k zlému používateľskému zážitku.

1 Východiská

V tejto kapitole popisujeme technologické a teoretické východiská, ktoré využívame v implementácii. Technologické východiská tvoria najmä Balíky, ktoré implementujú existujúce riešenia. Teoretické východiská zostávajú z veľkej časti podobné ako pri existujúcich riešeniach, aby sme vedeli následne efektívne porovnať našu implementáciu s existujúcimi riešeniami.

1.1 Využité technológie

V tejto podkapitole spomíname technológie, na ktorých spočíva naša implementácia. Balíky vysvetlíme hlavne na základe informácií uvedených v ich README.md súboroch a na základe článku [12]. Podrobný návod, ako vytvoriť algoritmy na základe nasledujúcich Balíkov sa nachádza v [12].

Programovací jazyk Python je veľmi používaným jazykom hlavne na účely dátovej vedy a strojového učenia. Vďaka knižniciam Numpy, Pandas, Matplotlib, Scikit-Learn a iným prostriedkom vieme ľahko ukladať dáta, manipulovať s dátami a aj získať dodatočné informácie o dátach. Napríklad pomocou Scikit-Learn vieme generovať štatistické modely na základe dát získaných z nabíjaciach staníc. Tieto štatistické modely potom môžeme použiť vo forme vstupov do nášho algoritmu. Náš problém je možné riešiť aj pomocou učenia posilňovaním, menovite [14], kde vďaka knižniciam Pytorch a Tensorflow vieme vyriešiť SAC algoritmus. [20, 12]

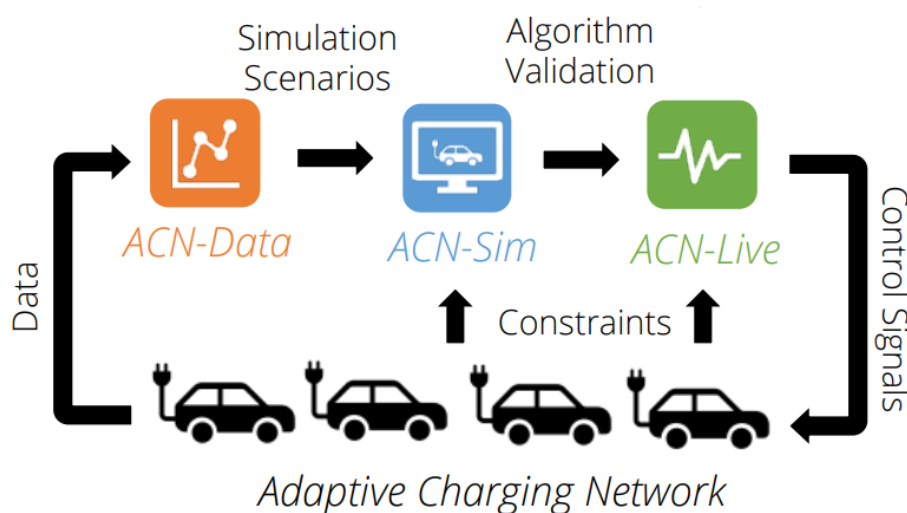
1.1.1 Balík ACN Portal.

Balík ACN Portal obsahuje záznamy nabíjania elektromobilov získane z nabíjaciach staníc Caltech, JPL a Office001. Každý záznam nabíjania elektromobilu obsahuje jeho príchod, odchod a požadovanú energiu. Balík ACN Portal sa nachádza v [7]. Balík ACN Portal pozostáva z viacerých komponentov:

1. **ACN-Data:** Dáta získané z nabíjaciach staníc pre elektromobily, konkrétne zo staníc Caltech, JPL a Office001. Vodiči elektromobilov musia cez mobilnú aplikáciu dať oskenovaný QR kód nabíjačky a potom zadať približný čas odchodu a množstvo požadovanej energie. Ak vodič neuvedie informácie cez mobilnú aplikáciu, nabíjačka bude

nabíjať len 8 ampérov a ak ani po 15 minútach vodič neuvedie informácie, tak nabíjačka prestane nabíjať. Dáta získané od vodičov elektromobilov, ale aj dáta slúžiace ku konfigurácii siete sú uložené v relačnej databáze. Vďaka tejto úložnej vrstve vieme vytvárať vizualizácie pre vodičov a pre sieťových operátorov. Ide hlavne o vizualizáciu stavu systému pre sieťových operátorov a stav nabíjania elektromobilov pre vodičov.

2. **ACN-Sim**: Simulátor používaný na testovanie a overovanie funkcionality algoritmov, systémov. Simulátor zabezpečuje realistické prostredie na overovanie funkcionality algoritmov, hlavne pre výskumníkov, ktorí nemajú prístup k reálnym nabíjacím systémom pre nabíjanie elektromobilov.
3. **ACN-Live**: Hardvér, ktorý bežia algoritmy nabíjania elektromobilov naživo. Keďže má rovnaké rozhranie ako ACN-Sim, tak vieme testovať algoritmy implementované v ACN-Sim, bez žiadnej zmeny kódu. [12, 10]



Obr. 1.1: Schéma ACN Portal. Zdroj obrázka je [12].

Obrázok 1.1 ilustruje interakciu medzi hlavnými componentami ACN Portal. Dáta o nabíjaní získava komponent ACN-Data. O obmedzenia siete a validáciu algoritmov sa stará ACN-Sim. V ACN-Live uvádzame konkrétne algoritmy z ACN-Sim do prevádzky. Na takú operáciu netreba meniť kód, lebo ACN-Sim a ACN-Live fungujú cez rovnaké rozhranie. [12, 13]

1.1.2 Ukážka vstupných dát ACN-Data

```

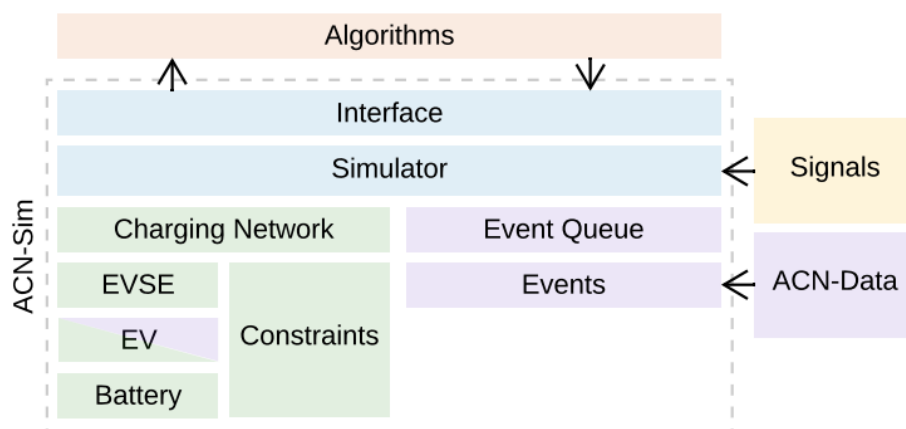
{
  "_id": "5bc90cb9f9af8b0d7fe77cd2",
  "clusterID": "0039",
  "connectionTime": "Wed, 25 Apr 2018 11:08:04 GMT",
  "disconnectTime": "Wed, 25 Apr 2018 13:20:10 GMT",
  "doneChargingTime": "Wed, 25 Apr 2018 13:21:10 GMT",
  "kWhDelivered": 7.932,
  "sessionID": "2_39_78_362_2018-04-25 11:08:04.400812",
  "siteID": "0002",
  "spaceID": "CA-496",
  "stationID": "2-39-78-362",
  "timezone": "America/Los_Angeles",
  "userID": null,
  "userInputs": null
},

```

Obr. 1.2: Vstupné dáta z nabíjania jedného elektromobilu 25. Apríla 2018. Zdroj: [1].

Obrázok vstupných dát 1.2 obsahuje viacero parametrov, ktoré používateľ musí zadať, aby mohol nabíjať svoj elektromobil na jednej staníc Caltech, JPL alebo Office001. Iné nabíjacie stanice používajú odlišné vstupné dáta, úvada sa v [13].

1.1.3 Architektúra simulátora ACN-Sim



Obr. 1.3: Architektúra simulátora ACN-Sim. Obrázok pochádza z článku [12].

Obrázok 1.3 nám pomáha pochopiť štruktúru a interakcie jednotlivých častí systému, aby sme vedeli implementovať vlastné algoritmy.

1.1.4 Balík Adacharge.

Balík Adacharge obsahuje kontrolný algoritmus MPC, ktorý rieši problém pridelenia maximálnej energie elektromobilom pri zachovaní obmedzení infraštruktúry pomocou konvexnej optimalizácie. Podrobnejší popis a zdrojový kód balíka Adacharge sa nachádza v [6].

1.1.5 Balík Acnportal-experiments.

Balík Acnportal-experiments uvádza príklady, ako sa Balík ACN portal dokáže použiť pri jednotlivých experimentoch. Tieto experimenty následne vedú zodpovedať výskumné otázky. Vo vzťahu k implementácii sú prioritné experimenty 1.2, 2.1 a 2.2. Krátky popis experimentov:

1. Experiment 1.2: Cieľom experimentu je porovnať rôzne konfigurácie infraštruktúry (level, kapacita transformera a počet nabíjačiek) a aj scheduling algoritmy. Tento experiment ukazuje výhodu smart sietí, lebo smart sieť je schopná minimalizovať kapitálové náklady na kapacitu transformera a aj minimalizovať operačné náklady.
2. Experiment 2.1: Cieľom experimentu je porovnávať výkonnosť algoritmov Round-Robin, First-Come First-Served, Earliest Deadline First, Least Laxity First. Výsledok experimentu má potom byť koľko percent požadovanej energie každý algoritmus môže dodať.

Podrobnejší popis experimentov a ich výsledky vieme nájsť v [5].

1.2 Aplikovateľnosť smart charging algoritmov.

Mnoho zaujímavých algoritmov z literatúry sa nedá využiť priamo v praxi. Hlavným dôvodom prečo sa nedajú aplikovať v praxi je podľa [13] to, že majú predpoklady, ktoré v praxi nefungujú alebo im chýba schopnosť splniť praktické obmedzenia a aj ciele.

Uvádame list vlastností smart charging algoritmu spomen, ktorý môže byť aplikovateľný v praxi:

1. Algoritmy musia brať do úvahy používateľské vstupy ako napríklad čas odchodu a množstvo požadovanej energie.
2. Algoritmy musia vyriešiť jednoducho problém s viacerými cieľmi.
3. Algoritmy musia vedieť pracovať s obmedzeniami v niekoľkoúrovňovej nevyváženej infraštruktúry.
4. Výstup algoritmov (plán nabíjania) musí spĺňať aspoň jednu z týchto vlastností: plynulé nabíjanie alebo spravodlivé zdieľanie kapacity medzi elektromobilmi.

5. Algoritmy musia byť odolné voči neistote v budúcich príchodoch elektromobilov.
6. Algoritmy musia vedieť narábať s diskretnými bodmi pre pilotové signály.
7. Algoritmy musia mať schopnosť si nárokovať nevyužitú kapacitu v prípade, keď elektromobili nevyužívajú alokovaný pilotový signál.

Algoritmus, ktorý všetky tieto podmienky spĺňa sa nazýva Adaptive Scheduling Algorithm (ASA). Podrobnejší popis, prečo tieto podmienky tento algoritmus spĺňa a aj list podmienok z ktorého sme čerpali sa nachádza v [13].




1.3 Konfigurácia a obmedzenia nabíjacích sietí

V tejto podsekcii uvádzame jednotlivé konfigurácie a obmedzenia nabíjacích sietí, na ktorých budeme navrhovať a testovať model agregátora flexibility. Vysvetlíjeme postupne rôzne typy nabíjania, vplyv batérie, obmedzenia sietí atď.

1.3.1 Typy nabíjania.

Pre naše účely sú potrebné 2 typy nabíjania AC level 1 a AC level 2. Nabíjanie AC level 1 je predovšetkým určené pre vlastníkov elektromobilov, ktorý ich chcú nabíjať celý deň (8 až 10 hodín). Rýchlosť nabíjania pri AC level 1 je 1.4 až do 1.9 kW za hodinu. Kapacita nabíjacieho kábla je od 110 až do 120 voltov.

AC level 2 charging je rýchlejší typ nabíjania než AC level 1. Rýchlosť nabíjania pri AC level 2 je od 2.5 do 19.2 Kw. To znamená, že viacero spotrebiteľov sa môže vystriedať pri nabíjaní počas dňa. Kapacita nabíjacieho kábla je dvojnásobne vyššia než pri AC level 1 nabíjaní. [22]

KNOW YOUR EV CHARGING STATIONS		
<p>AC Level One</p> 	<p>AC Level Two</p> 	<p>DC Fast Charge</p> 
<p>VOLTAGE 120V 1-Phase AC</p>	<p>VOLTAGE 208V or 240V 1-Phase AC</p>	<p>VOLTAGE 208V or 480V 3-Phase AC</p>
<p>AMPS 12–16 Amps</p>	<p>AMPS 12–80 Amps (Typ. 32 Amps)</p>	<p>AMPS >100 Amps</p>
<p>CHARGING LOAD 1.4–1.9 kW</p>	<p>CHARGING LOAD 2.5–19.2 kW (Typ. 6.6 kW)</p>	<p>CHARGING LOAD 50–350 kW</p>
<p>CHARGING TIME 3–5 Miles per Hour</p>	<p>CHARGING TIME 12–60 Miles per Hour</p>	<p>CHARGING TIME 60–80 Miles in 20 Minutes</p>

Obr. 1.4: Všetky dnešné typy nabíjania s príslušnými údajmi o rýchlosti nabíjania a kapacite kábla. Zdroj obrázka: [22].

1.3.2 Rýchlosť nabíjania batérie.

Pomocou triedy Battery, ktorá definuje ideálny model batérie vieme sami definovať vlastné modely batérie. Rýchlosť nabíjania ideálnej batérie je:

$$\hat{r}(t) = \min\{\bar{r}, r(t), \hat{e}(t)\}, \quad (1.1)$$

kde \bar{r} je maximálna rýchlosť nabíjania, $r(t)$ je pilotový signál poslaný do batérie a $\hat{e}(t)$ je nezaplnená energia batérie v čase t v ampéroch.

Existuje aj rozšírenie triedy Battery, ktoré sa nazýva Linear2StageBattery. Linear2StageBattery aproximuje po častiach lineárne nabíjanie elektromobilov s lítiovou batériou. Prvý stav, ktorý nazývame hromadné nabíjanie trvá zvyčajne od 0 do 70 až 80 percent stavu nabíjania. Rýchlosť nabíjania batérie Linear2StageBattery je:

$$\hat{r}(t) = \begin{cases} \min\{\bar{r}, r(t), \hat{e}(t)\} & \text{Ak } SoC \leq th \\ \min\{(1 - SoC) \frac{\bar{r}}{1-th}, r(t)\} & \text{inak} \end{cases} \quad (1.2)$$

kde th je hodnota energie po prechode z hromadného nabíjania do absorpčného nabíjania. Premenná SoC znamená stav nabíjania batérie.

Tento čiastočný lineárny model je dobrá aproximácia správania batérie počas nabíjania. Žiaľ, ani tento model nezachytáva všetky prípady, ktoré môžu správanie batérie zmeniť. [12]

1.3.3 Typy nabíjacích staníc.

Nasledujúce typy nabíjacích staníc pochádzajú z implementácie ACN Portal.

ChargingNetwork

Táto nabíjacia stanica predpokladá, že maximálne jeden elektromobil môže byť napojený na každú nabíjačku.

StochasticNetwork

Táto nabíjacia stanica pridelí nabíjačky elektromobilom náhodne. Implementuje čakací front v prípade ak nie sú voľné nabíjačky. Ak nastavíme parameter early departure na pravdivý, tak dovoľíme výmenu vozidla nabíjajúcim na stanici s vozidlom, ktoré sa nachádza v čakacom fronte. Tento typ nabíjacej siete je vhodný pre aplikovanie v reálnom živote, ale aj v pri generovaní udalostí zo štatistických modelov. [12]

1.3.4 Obmedzenia pri nabíjaní.

Nabíjacie systémy často fungujú na princípe radiálnych sietí. Musíme preto obmedziť množstvo voltov prechádzajúcich cez každé úzke miesto siete. Pomocou Kirchhoffových zákonov vieme definovať obmedzenia pri nabíjaní takto:

$$|I_j(t)| = \left| \sum_{i=1}^N A_{ij} r_i(t) e^{j\phi_i} \right| \leq R_j, \quad (1.3)$$

kde R_j je veľkosť prúdu, $I_j(t)$ je prúd prúdiaci cez úzke miesto siete, N je počet nabíjačiek v nabíjacej stanici, $r_i(t)$ je prúd poskytovaný nabíjačkou i v čase t . Dokopy máme T časových krokov. Parametrom ϕ_i vyjadrujeme fázový uhol pre aktuálny fázor, ktorý závisí na tom ako je nabíjačka i zapojená do siete. [12]

1.4 Problém nájdenia optimálneho schedulingu.

Problém nájdenia optimálneho plánu nabíjania (schedulingu) elektromobilov je optimalizačný problém. Existuje veľa spôsobov riešenia optimalizačného problému nájdenia optimálneho plánu nabíjania vozidiel, lebo môžeme použiť kontrolné algoritmy alebo jednoduché scheduling algoritmy. Kontrolné algoritmy MPC a PPC riešia problém nájdenia optimálneho scheduling tým, že predikujú údaje o budúcich príchode budúcich áut. Algoritmus PPC sa používa v kombinácii s jednoduchým scheduling algoritmom.

Vstupom všetkých scheduling algoritmov je množina elektromobilov a parametre na strane agregátora flexibility. Výstupom všetkých scheduling algoritmov je plán nabíjania (scheduling). [16]

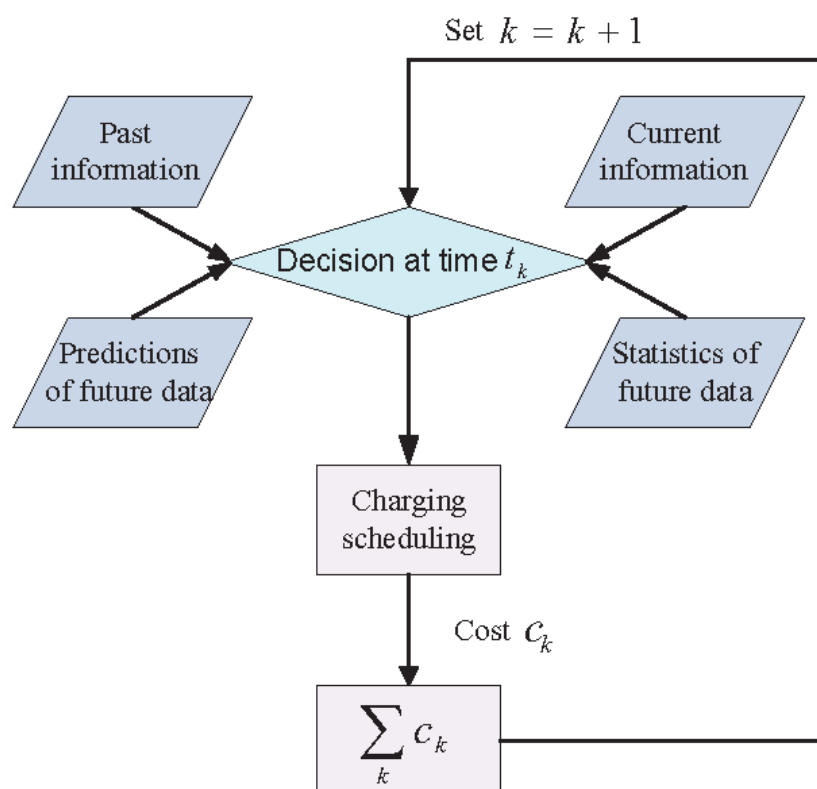
Algoritmy pre nabíjanie elektromobilov spadajú do 2 kategórii. Môžu byť scheduling online algoritmy, alebo offline scheduling algoritmy.

Offline algoritmy

Vyžadujú všetky informácie o všetkých tých elektromobiloch (vrátane parametrov elektromobilov s príchodom v budúcnosti). Pomocou týchto informácií vie agregátor flexibility prideliť vhodné množstvo energie každému elektromobilu a pritom minimalizovať cenu energie pre spotrebiteľov.

Online algoritmy

Online algoritmy využívajú len údaje o elektromobiloch nachádzajúcich sa na nabíjacej stanici. Najväčším problémom, ktorý všetky online algoritmy riešia, je problém predpovedania príchodov nových elektromobilov do nabíjacej stanice. Výkonnosť online algoritmov závisí na tom, ako dobre riešia tento problém. Optimálnu rýchlosť nabíjania v online algoritmoch môžeme zistiť pomocou riešenia problému konvexnej optimalizácie, alebo problému lineárneho programovania. Pre účely minimalizovania časovej a pamäťovej zložitosti metóda bisekcie a triediace algoritmy sa často využívajú. [2]



Obr. 1.5: Proces schedulingu pre online scheduling algoritmy pre nabíjanie elektromobilov. Zdroj obrázka je [18].

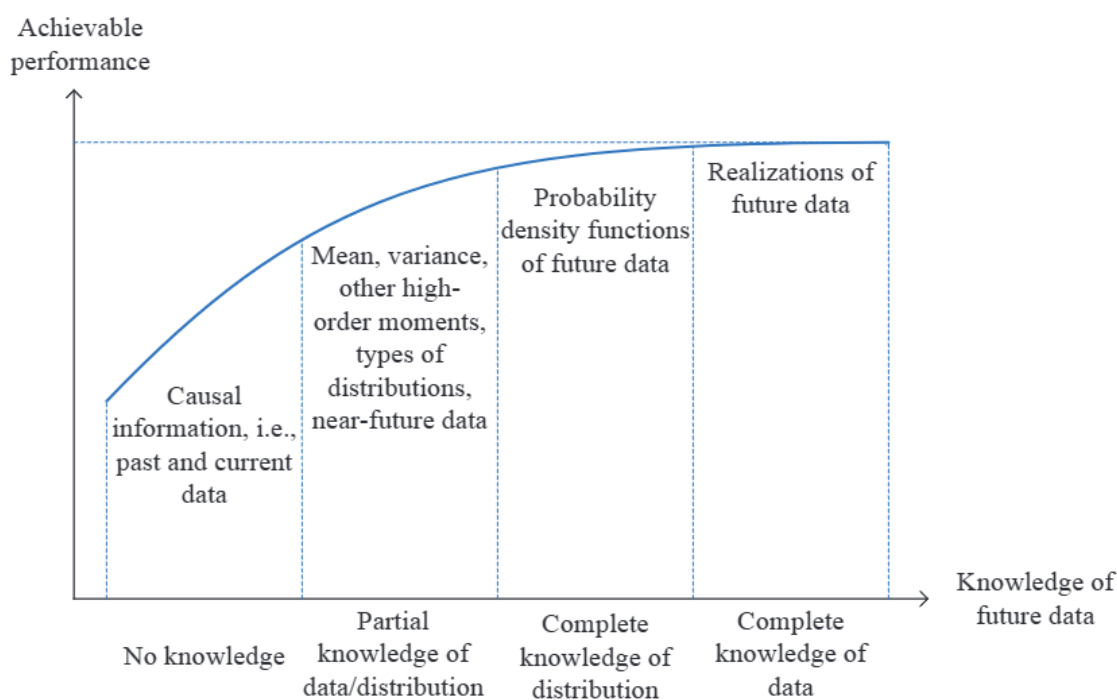
Obrázok 1.5 ilustruje všeobecne proces schedulingu pre online algoritmy. Riešenie opti-

malizačného problému pridelenia maximálnej energie pri zachovaní obmedzení infraštruktúry závisí od:

1. Informácii v minulosti.
2. Informácii v tomto okamihu.
3. Informácii v budúcnosti.
4. Štatistík o budúcich dáta (ak presné dáta o budúcich príchodoch nemáme).

Porovnanie offline a online algoritmov

Keďže je ťažké získať informácie o príchode nových elektromobilov (drahé alebo nemožné), tak sa prevažne využívajú v praxi online algoritmy. Offline algoritmy majú viac informácii o budúcich príchodoch elektromobilov, preto generujú lepší scheduling než online algoritmy.



Obr. 1.6: Graf ilustrujúci vplyv určitých vedomostí o budúcich príchodoch elektromobilov na kvalitu algoritmu. Zdroj grafu je [18].

Náš prístup k riešeniu problému

My riešime problém zabezpečovania optimálneho toku energie pre spotrebiteľov v nabíjacej sieti. Nabíjacia sieť má infraštruktúru a obmedzenia rovnaké ako v článku [12]. Na rie-

šenie tohoto problému používame online scheduling algoritmy. V týchto online algoritmoch riešime optimalizačný problém pridelovania maximálnej energie pri zachovaní obmedzení nabíjacej siete. Problém pridelovania maximálnej energie pri zachovaní obmedzení nabíjacej siete riešime viacerými spôsobmi, pomocou lineárneho vyhľadávania, metódy bisekcie a pomocou metódy zlatého rezu. [2]

Nižšie uvádzame scheduling algoritmy, ktoré my využívame pri implementácii, alebo ich používame už priamo z existujúcich riešení pri porovnávaní algoritmov.

1.5 Neriadené nabíjanie.

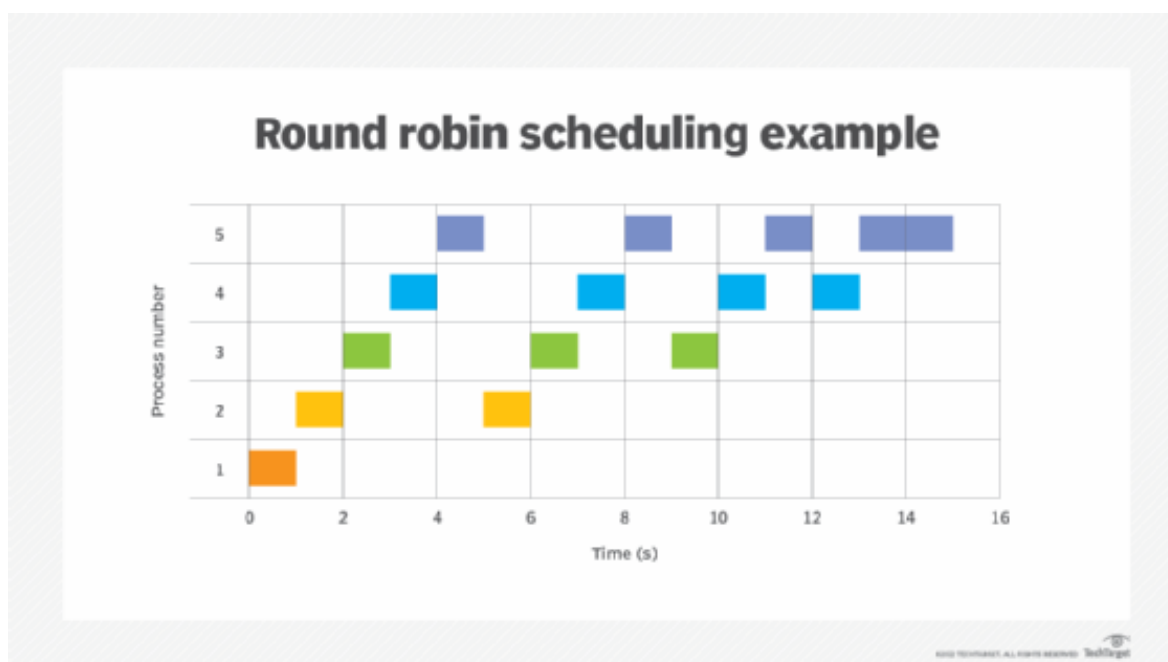
Tento scheduling algoritmus prideluje každému elektrickému vozidlu toľko energie, koľko je možné maximálne prideliť nerátajúc obmedzenia infraštruktúry nabíjacej stanice. Toto je najjednoduchší typ nabíjania, ktorý je stále používaný vo väčšine nabíjacích systémov na svete. [12]

1.6 Algoritmus Round Robin.

Scheduling algoritmus s názvom Round Robin, ktorý nazývame skrátené RR je algoritmus, ktorý je založený na myšlienke zdieľania nabíjacej kapacity medzi všetkými aktívnymi elektromobilmi. Pre každý aktívny elektromobil kontroluje 2 podmienky:

1. Je možné zvýšiť nabíjanie o 1kWh, tak aby sme zachovali obmedzenia nabíjacej siete? Ak áno, tak nabijeme elektromobil so zvýšenou rýchlosťou nabíjania a následne elektromobil pridáme na koniec frontu.
2. Nie je možné zvýšiť nabíjanie o 1kWh, tak aby sme zachovali obmedzenia nabíjacej siete? Ak áno, tak nabijeme elektromobil s fixným nabíjaním. Elektromobil necháme vo fronte na rovnakom mieste.

Elektromobily, ktoré dostali požadovanú energiu sú z fronty odstránené. Algoritmus RR skončí vtedy, keď vo fronte nebude žiaden elektromobil. [12]



Obr. 1.7: Pridel'ovanie času procesom na základe scheduling algoritmu Round Robin. Každý spotrebiteľ má prístup k rovnakým zdrojom pomocou algoritmu Round Robin. Zdroj obrázka: [19].

1.7 Triediace scheduling algoritmy

Algoritmy, ktoré spomínáme v tejto sekcii prideliujú energiu elektromobilom v takom poradí, v akom tie elektromobily utriedili. Triediace scheduling algoritmy používajú rôzne parametre elektromobilov pri ich triedení. Následne na riešenie problému maximálneho pridelenia energie elektromobilu pri zachovaní obmedzení nabíjacej siete triediace scheduling algoritmy využívajú metódu bisekcie alebo iné metód (napr. metóda lineárneho vyhl'adávaní). Tieto algoritmy môžu byť implementované aj ako online aj ako offline algoritmy. V prípade offline verzie triediacich algoritmov vieme lepšie minimalizovať cenu než v online scheduling algoritmoch. [2]

1.7.1 Least Laxity First.

Least Laxity First skrátené nazývame LLF, je scheduling algoritmus, ktorý prideluje elektromobilom energiu v poradí od elektromobilu s najnižšou laxitou po elektromobilu s najvyššou laxitou. Nevýhodou scheduling algoritmu LLF je, že spôsobuje časté výkyvy rýchlosti nabíjania pri dodávaní energie elektromobilom. Dlhodobý, taký spôsob nabíjania môže skrátiť životnosť batérii niektorých vozidiel.[2]

Laxitu počítame na základe vzorca:

$$\text{Laxita} = \text{zostávajúci čas do odchodu} - \frac{\text{zostávajúca požadovaná energia}}{\text{rýchlosť nabíjania}}, \quad (1.4)$$

kde predpokladáme, že rýchlosť nabíjania je vždy konštantná.

Príklad

Každý elektromobil i má definovanú trojicu (a_i, d_i, e_i) , kde a_i je čas pripojenia elektromobilu na nabíjačku, d_i je čas odpojenia vozidla z nabíjačky a e_i je požadovaná energia. Máme 4 vozidlá ktoré majú nasledujúce parametre:

1. (9, 15, 4).
2. (9, 12, 1).
3. (12, 17, 4).
4. (9, 13, 3).

Zároveň máme 4 nabíjačky v nabíjacej stanici, a preto vieme zabezpečiť nabíjanie vozidiel kedykoľvek, bez použitia prioritnej fronty. Predpokladáme, že cena energie je konštantná. Zvoľme rýchlosť nabíjania každej stanice na 1. Obmedzenia nabíjacej siete sú, že môže každú hodinu dodať len $2Kw$ energie.

Naša úloha je nájsť optimálny scheduling, kde našim cieľom je zabezpečiť požadované množstvo energie pre všetky elektromobily.

Náš optimálny scheduling problému je:

	t = 9	t = 10	t = 11	t = 12	t = 14	t = 15	t = 16
Vozidlo 1	1	0	1	1	1	-	-
Vozidlo 2	0	1	0	-	-	-	-
Vozidlo 3	-	-	-	1	1	1	1
Vozidlo 4	1	1	1	0	-	-	-

Ak napríklad vymeníme nabíjanie vozidla 2 o 10:00 s nabíjaním vozidla 1, tak nám vzniká iný scheduling, ktorý je tiež optimálny.

Na druhej strane existuje aj neoptimálny scheduling tohoto príkladu, vypojením vozidla 3 o 12:00. Vozidlo 3 v tom prípade by sa vypojoilo z nabíjacej siete buď o 9:00, 10:00 alebo 11:00 a napojilo o 12:00.

1.7.2 Smoothed Least Laxity First.

Smoothed Least Laxity First (skrátene sLLF) je scheduling algoritmus založený na algoritme LLF. Tento algoritmus lepšie rieši problém pridelovania maximálnej energie pri zachovaní obmedzení infraštruktúry nabíjacej siete lepšie než algoritmus LLF. Algoritmus sLLF tiež neobsahuje veľké výkyvy rýchlosti nabíjania pri pridelovaní energie elektromobilom narozdiel od algoritmu LLF.

Popis algoritmu sLLF sa nachádza v článku [2].

1.7.3 Algoritmus Enhanced Least Laxity First.

Ako reakciu na problém oscilácie pri nabíjaní elektromobilov v algoritme LLF vzniklo mnoho variánt vylepšeného algoritmu LLF. Jedným z nich je Enhanced Least Laxity First (ELLF). Algoritmus ELLF zaradí elektromobily s rovnakou najnižšou laxitou do jednej skupiny. Potom algoritmus ELLF prideli energiu elektromobilom v skupine použitím scheduling algoritmu EDF. Na pridelenie energie zvyšku elektromobilov algoritmus ELLF aplikuje LLF. [15]

1.7.4 Algoritmus Earliest Deadline First.

Early deadline first skrátene nazývame EDF, je scheduling algoritmus. Algoritmus EDF prideli energiu elektromobilom v poradí od elektromobilov s najskorším časom odchodu po elektromobily s najneskorším časom odchodu. [12]

1.7.5 Group Earliest Deadline First.

Group Earliest Deadline First je scheduling algoritmus, ktorého vytvorili nato, aby zlepšil výkonnosť EDF počas preťaženia reálnej multimediálnej aplikácie. Tento scheduling algoritmus sa zakladá na myšlienke skupinového schedulingu. To znamená, že elektromobily, ktoré majú podobné časy príchodu patria do skupiny. Potom sa priradi energiu týmto elektromobilom na základe scheduling algoritmu Shortest Job First. [17]

1.7.6 First-Come First-Served.

Scheduling algoritmus First-Come First-Served (skrátene FCFS) má veľmi podobnú funkcionality ako uncontrolled charging algoritmus. Hlavný rozdiel spočíva v tom, že algoritmus FCFS je časovo oneskorený, lebo obsahuje autá v rade, takže každé elektronické vozidlo si musí počkať pre svoje nabíjanie. [12]

1.7.7 Last-Come First-Served.

Scheduling algoritmus Last-Come First-Served (skrátene LCFS) priradi energiu elektromobilom, prioritne od vozidla s najneskorším časom pripojenia na nabíjačku po vozidlo s najskorším časom pripojenia na nabíjačku. [12]

1.7.8 Longest Remaining Processing Time.

Scheduling algoritmus Longest Remaining Processing Time je algoritmus, ktorý prideľuje energiu elektromobilom takto:

1. Utriedi vstupné pole aktívnych elektromobilov od elektromobilu najdlhším zostávajúcim časom nabíjania po elektromobil s najkratším zostávajúcim časom nabíjania.
2. Pre každé vozidlo v tom poradí algoritmus vyrieši optimalizačný problém pridelovania maximálnej energie pri zachovaní obmedzení infraštruktúry.

1.7.9 Shortest Job Next.

Algoritmus Shortest Job Next, skrátene SJN je scheduling algoritmus. SJN funguje ako Longest Remaining Processing Time až na jeden rozdiel. Rozdiel je, že pole aktívnych elektromobilov triedi od elektromobilu s najkratším zostávajúcim časom nabíjania, po vozidlo s najdlhším zostávajúcim časom nabíjania.

1.8 Kontrolné scheduling algoritmy

V tejto sekcii hovoríme o kontrolných algoritmoch, ktoré riešia problém optimálneho toku energie v nabíjacej sieti. Kontrolné algoritmy sú algoritmy, ktoré poskytujú kontrolné akcie, na základe vstupných dát, aby sme zistili nejaký cieľ. Bez kontrolných algoritmov by dnes nemohli fungovať automatické systémy, ktoré často využívame dnes. Tieto algoritmy musia vedieť fungovať v obmedzenom prostredí ako napríklad pri obmedzeniach uvedených v sekcii 1.3. [3]

1.8.1 Typy kontrolných algoritmov

open-loop algoritmy

Sú najjednoduchším typom kontrolných algoritmov. Vždy vracajú pre rovnaký vstup rovnaký výstup. Používajú sa v systémoch, kde netreba meniť výstupné dáta na základe zmeny vo vstupných dátach.

closed-loop algoritmy

Sú zložitejšie než open-loop algoritmy. Používajú odozvu aby upravili výstupné dáta na základe zmien vo vstupných dátach. Používajú sa v systémoch, kde sa výstup musí upravovať na základe zmien vo vstupe.

feedback-control algoritmy

Sú najzložitejším typom kontrolných algoritmov. Používajú odozvu aby upravili výstupné dáta na základe zmien vo vstupných dátach. Používajú taktiež prediktívne modely

aby predpovedali zmenu vo vstupných dátach. Používajú sa v systémoch, ktoré predpovedajú budúce zmeny vstupov a kde sa výstup musí upravovať na základe zmien vo vstupe. [3]

1.8.2 Model Predictive Control.

Model Predictive Control skrátene MPC je jeden z kontrolných algoritmov. MPC je feedback control algoritmus. MPC využíva prediktívny model, aby zistil kontrolné akcie v budúcnosti, ktoré optimalizujú výkon počas nejakého časového horizontu. Cieľom MPC je zabezpečiť čo najlepší kontrolný výkon, a pritom splňať obmedzenia systému.

Komponenty MPC

MPC sa skladá z 3 komponentov:

1. Matematický model procesu. Tento matematický model zachytáva vzťah medzi vstupom a výstupom a zahŕňa aj dynamiku systému.
2. Optimizátor, ktorý rieši zadaný optimalizačný problém a pritom splňa obmedzenia systému.
3. Funkcia nákladov, ktorá meria kvalitu riešenia optimizátora.

Výhody MPC

Veľká výhoda kontrolného algoritmu MPC v porovnaní s tradičnými kontrolnými algoritmi, je že nemusí mať fixný počet kontrolných parametrov, ktoré sú potom upravované na základe experimentov. Algoritmus MPC využíva dynamický model, aby predikoval správanie v budúcnosti. MPC na rozdiel od tradičných kontrolných algoritmov vie riešiť aj nelineárne kontrolné problémy. Nelineárne kontrolné problémy sú problémy, v ktorých je vzťah medzi vstupom a výstupom je nelineárny. [4]

Existujúce riešenia

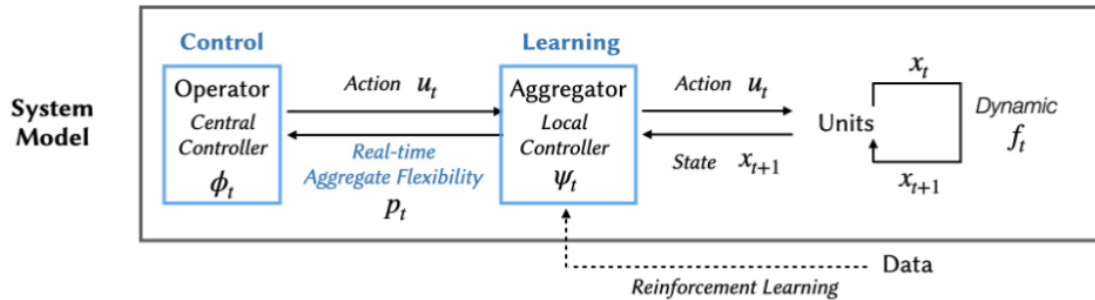
Veľá existujúcich riešení problému hľadania optimálneho schedulingu využíva algoritmus MPC. My sa orientujeme hlavne na algoritmus nazývaný Adaptive Charging Algorithm, ktorý je založený na MPC a rieši pritom problém konvexnej optimalizácie. [12, 13]

1.8.3 Penalised Predictive Control.

Algoritmus Penalised Predictive Control (PPC) je vylepšená verzia algoritmu MPC. Pointa algoritmu spočíva v komunikácii medzi systémovým operátorom a agregátorom. Agregátor poskytne flexibilitu (naučenú na základe RL algoritmu SAC) systémovému operátorovi. Systémový operátor aplikuje algoritmus PPC, ktorého vstupom je flexibilita a výstupom

je energia. Následne systémový operátor pridelí výstupnú energiu algoritmu PPC agregátorovi. Agregátor potom použije jednoduchý scheduling algoritmus (EDF alebo LLF), aby pridelil túto energiu aktívnym elektromobilom.

Narozdiel od MPC tento algoritmus nevyžaduje veľa premenných, ale na druhej strane obsahuje obsiahly algoritmus SAC na učenie flexibility. Výhodou PPC je tiež, že nemusí vedieť o obmedzeniach siete a stavu, pri riešení optimalizačného problému, lebo to je zahrnuté vo flexibilitě generovanej agregátorom. [14]



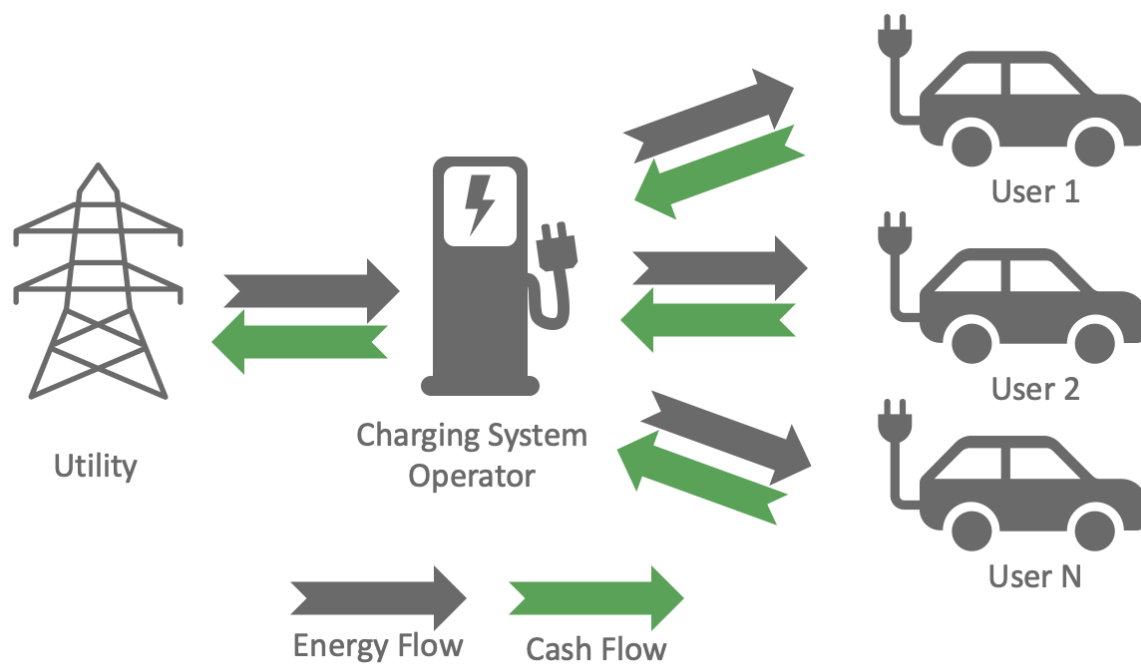
Obr. 1.8: Model systému založený na feedback control algoritmu. Operátor implementuje kontrolný algoritmus a agregátor implementuje RL algoritmus, ktorý generuje flexibilitu. Obrázok je prevzatý z [14].

1.9 Cena nabíjania.

Náš spôsob definovania ceny nabíjania je: Na konci každého mesiaca nastavíme ceny nabíjania ako α_i^* , pre každé nabíjanie i v tom mesiaci. Hovoríme, že e_i je dodaná energia pre spotrebiteľa j počas nabíjania i , kde S_j je množina všetkých nabíjaní používateľa j . Potom náklady za nabíjania spotrebiteľa j sa na konci mesiaca vypočítajú takto:

$$\sum_{i \in S_j} \alpha_i^* e_i. \quad (1.5)$$

Narozdiel od konštantných cien a cien závislých na čase (napríklad v [14]) my používame ceny (tarify), ktoré zachytávajú skutočné ceny energie, preťaženie infraštruktúry a na základe dopytu po elektrickej energii. [8]



Obr. 1.9: Systéme toku energie a aj zisky za energiu. Zdroj obrázka je: [21].

2 Návrh riešenia

V tejto kapitole popisujeme naše navrhnuté riešenie problému schedulingu, na základe ktorého vie potom agregátor flexibility vie pridelit' energiu elektrickým vozidlám tak, aby sme optimalizovali tok energie a minimalizovali naše náklady na kapacitu a minimalizovali náklady spotrebiteľov.

2.1 Least Laxity First

2.2 Smoothed Least Laxity First.

2.3 Riešenie problému prípustnosti.

2.4 Návrh nášeho MPC modelu.

3 Implementácia

4 Testovanie

Záver

Literatúra

- [1] Caltech. Acn-data. <https://ev.caltech.edu/dataset>, 2023. Accessed: 3rd December, 2023.
- [2] Niangjun Chen, Christian Kurniawan, Yorie Nakahira, Lijun Chen, and Steven H. Low. Smoothed least-laxity-first algorithm for ev charging, 2021.
- [3] Collimator. What is a control algorithm? <https://www.collimator.ai/reference-guides/what-is-a-control-algorithm#:~:text=Definition%20of%20a%20control%20algorithm&text=These%20algorithms%20are%20designed%20to,is%20either%20undesirable%20or%20impossible.,> 2023. Accessed: 3rd December, 2023.
- [4] Collimator. What is model predictive control? <https://www.collimator.ai/reference-guides/what-is-model-predictive-control>, 2023. Accessed: 3rd December, 2023.
- [5] Z. J. Lee and S. Sharma. acnportal-experiments. <https://github.com/caltech-netlab/acnportal-experiments>, Dec. 2023.
- [6] Z. J. Lee and S. Sharma. adacharge. <https://github.com/caltech-netlab/adacharge>, Dec. 2023.
- [7] Z. J. Lee, S. Sharma, and D. Johansson. acnportal. <https://github.com/zach401/acnportal>, Dec. 2023.
- [8] Zachary Lee, John Pang, and Steven Low. Pricing ev charging service with demand charge. *Electric Power Systems Research*, 189:106694, 12 2020.
- [9] Zachary J. Lee, Daniel Johansson, and Steven H. Low. Acn-sim: An open-source simulator for data-driven electric vehicle charging research. In *2019 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*, pages 1–6, 2019.
- [10] Zachary J. Lee, George Lee, Ted Lee, Cheng Jin, Rand Lee, Zhi Low, Daniel Chang, Christine Ortega, and Steven H. Low. Adaptive charging networks: A framework for smart electric vehicle charging, 2020.

- [11] Zachary J. Lee, Tongxin Li, and Steven H. Low. Acn-data: Analysis and applications of an open ev charging dataset. In *Proceedings of the Tenth ACM International Conference on Future Energy Systems, e-Energy '19*, page 139–149, New York, NY, USA, 2019. Association for Computing Machinery.
- [12] Zachary J. Lee, Sunash Sharma, Daniel Johansson, and Steven H. Low. Acn-sim: An open-source simulator for data-driven electric vehicle charging research, 2021.
- [13] Zachary Jordan Lee. *The Adaptive Charging Network Research Portal: Systems, Tools, and Algorithms*. Dissertation, California Institute of Technology, 2021.
- [14] Tongxin Li, Bo Sun, Yue Chen, Zixin Ye, Steven H. Low, and Adam Wierman. Learning-based predictive control via real-time aggregate flexibility. *IEEE Transactions on Smart Grid*, 12(6):4897–4913, nov 2021.
- [15] microcontrollerslab. Least laxity first (llf) scheduling algorithm. <https://microcontrollerslab.com/least-laxity-first-llf/>, 2023. Accessed: 3rd December, 2023.
- [16] Joy Chandra Mukherjee and Arobinda Gupta. A review of charge scheduling of electric vehicles in smart grid. *IEEE Systems Journal*, 9(4):1541–1553, 2015.
- [17] Vijayshree Shinde and Seema C. Comparison of real time task scheduling algorithms. *International Journal of Computer Applications*, 158:37–41, 01 2017.
- [18] Wanrong Tang, Suzhi Bi, and Ying Jun Angela Zhang. Online charging scheduling algorithms of electric vehicles in smart grid: An overview. *IEEE Communications Magazine*, 54:76–83, 2016.
- [19] TechTarget. round robin. <https://www.techtarget.com/whatis/definition/round-robin>, 2023. Accessed: 3rd December, 2023.
- [20] Jake VanderPlas. *Python data science handbook : essential tools for working with data*. O'Reilly Media, Inc, Sebastopol, CA, 2016.
- [21] Steven H. Low Zachary J. Lee, John Z.F. Pang. Pricing ev charging service with demand charge. <https://resnick.caltech.edu/documents/17395/RSIonline-RH-Lee-Z-SEPT-2020.pdf>, 2020. Accessed: 3rd December, 2023.
- [22] ZDWL. Ev charger levels. <https://zdw1-tec.com/news/ev-charger-levels/>, 2023. Accessed: 3rd December, 2023.

Príloha A: obsah elektronickej prílohy

Príloha B: Používateľská príručka