

Testovanie prototypu vo vzťahu k reálnej mikrosieti

Overenie funkčnosti integrovaného riešenia

*Medzinárodné centrum excelentnosti pre výskum inteligentných a bezpečných informačno-komunikačných technológií a systémov
II. etapa*



Táto publikácia vznikla vďaka podpore v rámci Operačného programu Integrovaná infraštruktúra pre projekt: *Medzinárodné centrum excelentnosti pre výskum inteligentných a bezpečných informačno-komunikačných technológií a systémov II. etapa*, ktorý je spolufinancovaný zo zdrojov Európskeho fondu regionálneho rozvoja.



Grafické informačné systémy

SFÉRA, a.s. • Karadžičova 2 • 811 08 Bratislava

tel.: +421 2 502 13 142

ISBN 978-80-89778-13-3

© SFÉRA, a.s., 2023

Táto publikácia je dielom kolektívu autorov:

Láznička Libor, Holiš Martin

Ostatní autori:

Bauer Pavol, Moško Daniel, Kaňuk Martin, Liptáková Lucia, Vannay David, Minárik Michal, Studenič Róbert, Deák Tomáš, Novotný Jozef

Konzultant: Burget Radim

OBSAH

1	ÚVOD	5
2	POROVNANIE VYTVORENÉHO SOFTVÉRU S APLIKÁCIAMI ANALYZOVANÝMI MEDZINÁRODNOU AGENTÚROU PRE ENERGIU	6
2.1	Prehľad programu HOMER Pro	8
2.2	Prehľad programu iHOGA PRO+	9
2.3	Vybrané kritériá na porovnanie sw nástrojov a ich definícia	10
2.4	Definícia kritérií	12
2.4.1	Prípadové štúdie samostatných mikrosietí	13
2.4.2	Simulácia Austrália	13
2.4.3	Simulácia Holandsko	15
2.4.4	Simulácia USA	18
2.5	Hodnotenie softvérových nástrojov	20
3	DIGITÁLNE DVOJČA	23
3.1	Mriežkové a náhodné vyhľadávanie	23
3.1.1	Opis algoritmov náhodného vyhľadávania	23
3.1.2	Simulované žihanie	26
3.1.3	Metóda gradientného zostupu	27
3.1.4	Particle Swarm Optimization (PSO)	28
4	MICROGRIDBOX® PORTÁL	29
4.1	Správa projektov	29
4.2	Knižnica grafických elementov	31
4.3	Modelovanie	33
4.3.1	Prepojenie prvkov	34
4.3.2	Vymazanie prvkov	34
4.3.3	Modifikácia hodnôt atribútov	34
4.3.4	Ďalšie operácie	35
4.4	Vizualizácia	35
5	TESTOVANIE PROTOTYPU VO VZŤAHU K REÁLNEJ MIKROSIETI	38
5.1	Plánovanie testovania	38
5.2	Návrh testov	39
5.3	Implementácia testov	40
5.4	Hodnotenie a vyhodnotenie výsledkov testov	40
5.5	Oprava a pretestovanie	41
5.6	Testovanie prototypu na reálnych dátach	41
5.6.1	Popis charakteristík prototypu využívajúceho prostredie Microsoft Excel	42
6	OVERENIE FUNKČNOSTI INTEGROVANÉHO RIEŠENIA	48
6.1	Testovanie funkčnosti	48
6.2	Testovanie výkonu	49
6.3	Testovanie použiteľnosti	49
6.4	Testovanie bezpečnosti	50

6.5	Overovanie kompatibility	51
6.6	Grafická vizualizácia integrovaného riešenia Mikrogridboxu	51
6.7	Záver z overovania funkčnosti.....	55
7	ZÁVER	57
8	ZOZNAM OBRÁZKOV	58
9	ZOZNAM TABULIEK.....	59

1 ÚVOD

Dokument bol vytvorený, zavedený a udržiavaný v súlade s cieľom Aktivity č. 6: „Experimentálny vývoj v oblasti nástrojov pre modelovanie a simuláciu inteligentných sietí/mikrogridov“ spoločne pre míľnik č. 5: „Testovanie prototypu vo vzťahu k reálnej mikrosieti“ a míľnik č. 6: „Overenie funkčnosti integrovaného riešenia“.

Dokument nadväzuje na výsledky míľnika č. 4 Aktivity č. 6, ktorý predstavuje etapu vytvorenia a popisu digitálneho dvojčaťa laboratórnej mikrosiete.

V rámci dokumentu sa postupne venujeme analýze Medzinárodnej agentúry pre energiu, kde kolektív vedeckých pracovníkov analyzuje dostupné softvérové riešenia pre problematiku mikrosietí. Tieto softvére následne porovnáme s vyvinutým riešením MicroGridBox® portálu. Rozvíjame problematiku používania rôznych metód výpočtu potrebných pre funkčnosť vyvinutej aplikácie batérievej kalkulačky. Samotný softvér MicroGridBox® portálu je podrobne popísaný z hľadiska architektúry riešenia, implementácie frameworku, prepojenia na databázu, užívateľské rozhranie a samotné funkcie softvéru. V časti testovanie prototypu vo vzťahu k reálnej mikrosieti popíšeme plánovanie, návrh a implementáciu vykonaných testov, taktiež hodnotenie a vyhodnotenie výsledkov testov, a na základe analýzy opravu postupov a opätovné pretestovanie. V záverečnej časti overenia funkčnosti integrovaného riešenia MicroGridBox® sa budeme zaoberať splnením požiadaviek na funkčnosť, výkon, použiteľnosť, bezpečnosť a kompatibilitu.

Významným podkladom pre míľnik č. 5 a č. 6 Aktivity č. 6 sú výsledky míľnika č. 5 Aktivity č. 5, ktoré sa zaoberajú testovaním a funkčnosťami vytvorenej relačnej databázy.

2 POROVNANIE VYTVORENÉHO SOFTVÉRU S APLIKÁCIAMI ANALYZOVANÝMI MEDZINÁRODNOU AGENTÚROU PRE ENERGIU

Medzinárodná agentúra pre energiu – fotovoltaický energetický program (International energy agency – photovoltaic power systems programme) sa vo svojej úlohe č. 18¹ zaoberá elektrickými systémami mimo siete a elektrickými systémami na okraji siete, ktoré zahŕňajú fotovoltaiku. Off-grid označuje elektrické systémy alebo siete, ktoré sú vzdialené od hlavnej elektrickej siete. Tieto siete sú často štátom vlastnenou alebo regulovanou elektrickou infraštruktúrou. Príklady systému mimo siete zahŕňajú:

- Samostatný byt napájaný systémom, ktorý zahŕňa fotovoltaiku,
- Ostrovná minisieť poháňaná obnoviteľnou energiou, ktorá zahŕňa fotovoltaiku,
- Energetický systém, ktorý dodáva elektrinu celej ostrovnej komunite pozostávajúcej zo stoviek ľudí,
- Komunikačný uzol umiestnený vo vzdialenej oblasti.

Hrana siete sa vzťahuje na oblasti, kde môže byť hlavná elektrická sieť nestabilná alebo nevhodná na daný účel a ako riešenie môže slúžiť použitie systémov, ktoré zahŕňajú fotovoltaiku. Oblasti na okraji siete sú často vystavené podobným problémom ako oblasti mimo siete, pokiaľ ide o spoľahlivosť, odolnosť a bezpečnosť a fotovoltaika môže byť súčasťou riešenia pre tieto oblasti. Cieľom úlohy 18 je nájsť technické problémy a bariéry, ktoré ovplyvňujú plánovanie, financovanie, dizajn, výstavbu a prevádzku a údržbu systémov mimo siete a na okraji siete, najmä tých, ktoré sú spoločné medzi krajinami, trhmi a systémom. Škálovať a ponúkať riešenia, nástroje, usmernenia a technické správy na bezplatné šírenie pre tých, ktorí by z nich mohli mať úžitok. Otázky, na ktoré sa zamerali v súvislosti s fotovoltaickým systémom mimo siete a na okraji siete, sa sústreďujú na:

- Spoľahlivosť - systém, ktorý má schopnosť generovať a distribuovať energiu, aby splnil požiadavky tých, ktorí sú spojení s vysokým stupňom dôvery.
- Odolnosť - systém, ktorý dokáže odolať alebo sa rýchlo zotaviť z prírodných katastrof, úmyselného zasahovania, alebo nehôd.
- Bezpečnosť - systém, ktorý je cenovo dostupný z hľadiska trvalej udržateľnosti a poskytuje neprerušovanú dodávku energie, ktorá primerane pokryje súvisiaci dopyt.²

Energetické systémy mimo siete a na okraji siete sa za posledných niekoľko desaťročí zvýšili v prevalencii a zložitosti, pretože náklady na solárnu fotovoltaiku a teraz aj skladovanie energie výrazne klesli. Preto je dôležité pochopiť, ako plánovať projekt, ktorý spĺňa potreby vlastníka aktív, koncových používateľov a všetkých ostatných zainteresovaných strán. Zatiaľ čo typ analýzy sa zvyčajne zameriava predovšetkým na ekonomiku projektu, je dôležité zvážiť aj sociálne, behaviorálne a environmentálne faktory. Správa vypracovaná IEA PVPS Úloha 18 má za cieľ poskytnúť usmernenie pre zainteresované strany, ktoré sa podieľajú na navrhovaní a vývoji projektov mimosieťovej fotovoltaiky (PV) poskytnutím komplexného sprievodcu štúdiami uskutočniteľnosti.

Je určený na použitie ako referenčný nástroj v počiatočných štádiách procesov plánovania a rozhodovania pre takéto projekty.³

V publikovanej práci „Nástroje na myslenie pre návrh a vývoj mikrogridov“ mal tím vedeckých pracovníkov* (Gautam Rituraj, Niccolo Ficarelli, Jorge Ortiz, Gautham Ram Chandra Mouli, Paul Rodden, Christopher Martell, Xavier Vallve, Pavol Bauer) za úlohu porovnanie softvérových nástrojov na návrh a vývoj samostatných mikrosietí, pričom sa sústredili na identifikáciu hlavných výhod, slabých stránok a existujúcich nedostatkov softvérových nástrojov na základe definovaných kritérií porovnania (ako sú funkčnosť, použiteľnosť, podporovateľnosť, efektívnosť, optimalizovaný návrh systému atď.). Túto úlohu viedla výskumná skupina DCE&S na Katedre udržateľnej energie TU Delft, ktorú podporovali rôzni medzinárodní partneri, konkrétne Trama Tecnoambiental (TTA) Barcelona, Sapin; Ekistica,

¹ IEA – PVPS, Task 18 (04/2023)

² <https://iea-pvps.org/research-tasks/off-grid-and-edge-of-grid-photovoltaic-systems/>

³ <https://www.pv-magazine.com/2023/03/15/optimizing-off-grid-pv-systems-for-different-design-criteria/>

Austrália; GSES Global Sustainable Energy Solutions, Austrália. Pre túto prácu boli stanovené štyri nasledujúce etapy:

- Výber softvérových nástrojov a definovanie kritérií na porovnanie týchto nástrojov,
- Prípadové štúdie samostatných mikrosietí s meraniami a obchodnými údajmi,
- Simulácie a analýza výsledkov,
- Hodnotenie softvérových nástrojov a odporúčania.

Pri navrhovaní a optimalizácii samostatnej/mikrosiete je výber správnych softvérových nástrojov nevyhnutný na zabezpečenie efektívneho a nákladovo efektívneho systému. V literatúre sa uvádza množstvo softvérových nástrojov, z ktorých každý má svoje silné a slabé stránky, na analýzu mikrosietí, napríklad technickú, ekonomickú a environmentálnu.^{4, 5, 6}

Medzi kritické faktory, ktoré zväžili pri výbere softvérových nástrojov na návrh autonómnych mikrosietí, patrila zložitosť systému, typ a počet zdrojov energie a záťaží a požadovaná úroveň riadenia a monitorovania.

Tabuľka 1 Prehľad charakteristík/funkcií všeobecne používaných softvérových nástrojov na simuláciu a/alebo optimalizáciu mikrosietí

Softvér	HOMER Pro	iHOGA PRO+	RETScreen	HYBRID2	HYBRIDY	SAM
Funkcie / Nástroje						
PV	✓	✓	✓	✓	✓	✓
Ukladanie batérií	✓	✓	✓	✓	✓	✓
Vietor	✓	✓	✓	✓	✓	✓
Dieselový generátor	✓	✓	✓	✓	✓	
Hydro	✓	✓	✓	✓	✓	
Palivový článok	✓	✓		✓	✓	
Elektrolyzér	✓	✓		✓	✓	
Vodíková nádrž	✓	✓		✓	✓	
Zaťaženie vodíkom	✓	✓		✓	✓	
Tepelné zaťaženie	✓		✓	✓	✓	
Stratégie kontroly	✓	✓		✓		
Samostatná stránka	✓	✓	✓			
Pripojenie do siete	✓	✓	✓			
Simulácia	✓	✓	✓	✓	✓	✓
Meteorologické údaje od NASA	✓	✓	✓			
Technická analýza	✓	✓				
Ekonomická analýza	✓	✓	✓	✓	✓	✓
Analýza emisií	✓	✓	✓			
Analýza citlivosti	✓	✓	✓			
Analýza pravdepodobnosti		✓				

⁴ D. Kaur a P. S. Cheema, "Software tools for analyzing the hybrid renewable energy sources:- A review," in International Conference on Inventive Systems and Control (ICISC), Coimbatore, 2017

⁵ K. Anoune, M. Bouya, A. Astito a A. B. Abdellah, "Sizing methods and optimization techniques for PV-wind based hybrid renewable energy system: A review," Renewable and Sustainable Energy Reviews, vol. 93, pp. 652-673, 2018

⁶ P. Shaikh, A. Shaikh, Z. Memon, A. Lashari a Z. Leghari, "Microgrids: J Energy Res., roč. 45, č. 9, s. 12564-12597, 2021

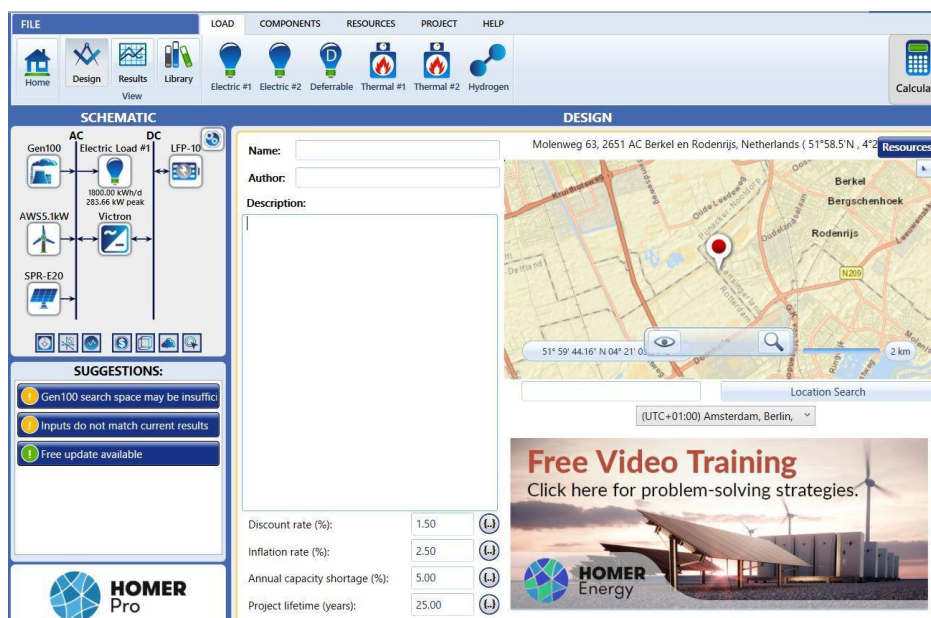
Softvér	HOMER Pro	iHOGA PRO+	RETScreen	HYBRID2	HYBRIDY	SAM
Funkcie / Nástroje						
Viacnásobný alebo jedinečný cieľ - optimalizácia	✓	✓				

Z uvedeného zoznamu softvérov sa v analýze podrobnejšie zaoberali programami HOMER Pro a iHOGA PRO+.

2.1 Prehľad programu HOMER Pro

HOMER Pro je softvér spoločnosti HOMER Energy, ktorý sa špecializuje na optimalizáciu mikrosystémov.⁷ Tento softvér vyvinulo Národné laboratórium pre obnoviteľné zdroje energie (NREL). Umožňuje používateľom vyhodnotiť a porovnať rôzne konfigurácie mikrosietí vrátane obnoviteľných zdrojov energie, systémov skladovania a riadenia energie, konvertorov, sietí a tradičných generátorov s cieľom určiť nákladovo najefektívnejšiu a najúčinnnejšiu možnosť. Okrem toho poskytuje aj možnosť vykonať analýzu citlivosti. Dá sa použiť na vykonanie komplexnej finančnej (ktorá zahŕňa výpočet vnútorného výnosového percenta, čistej súčasnej hodnoty (NPV) a ďalších finančných ukazovateľov) a environmentálnej analýzy.⁸ Program HOMER Pro využíva dve rôzne optimalizačné techniky. Prvá z nich, algoritmus prehľadávania mriežky, vyhodnocuje všetky možné nastavenia systému definované priestorom prehľadávania. Druhý algoritmus, optimalizátor HOMER, využíva jedinečný algoritmus, ktorý používa vlastný derivát - voľný algoritmus na nájdenie systému s najnižšími nákladmi. Program HOMER Pro potom predloží zoznam konfigurácií systému, ktoré sú zoradené podľa čistých súčasných nákladov (NPC, náklady na životný cyklus). Tento zoznam možno využiť na porovnanie rôznych alternatívnych návrhov.

Softvér sa dá využiť v rôznych aplikáciách vrátane vzdialených komunít, ostrovov, vojenských základní, areálov a komerčných a priemyselných zariadení. Podľa aplikácií sú pre tento softvér k dispozícii rôzne možnosti licencií.



Obrázok 1 Grafické používateľské rozhranie (GUI) softvéru Homer Pro zobrazujúce domovskú kartu, kde je možné vidieť architektúru simulovaného systému a návrhy

⁷ HOMER Energy. (2022). [Online]. Dostupné na: <https://www.homerenergy.com/>

⁸ Príručka HOMER Pro. (2022). [Online]. Dostupné na: <https://www.homerenergy.com/products/pro/docs/3.14/index.html>

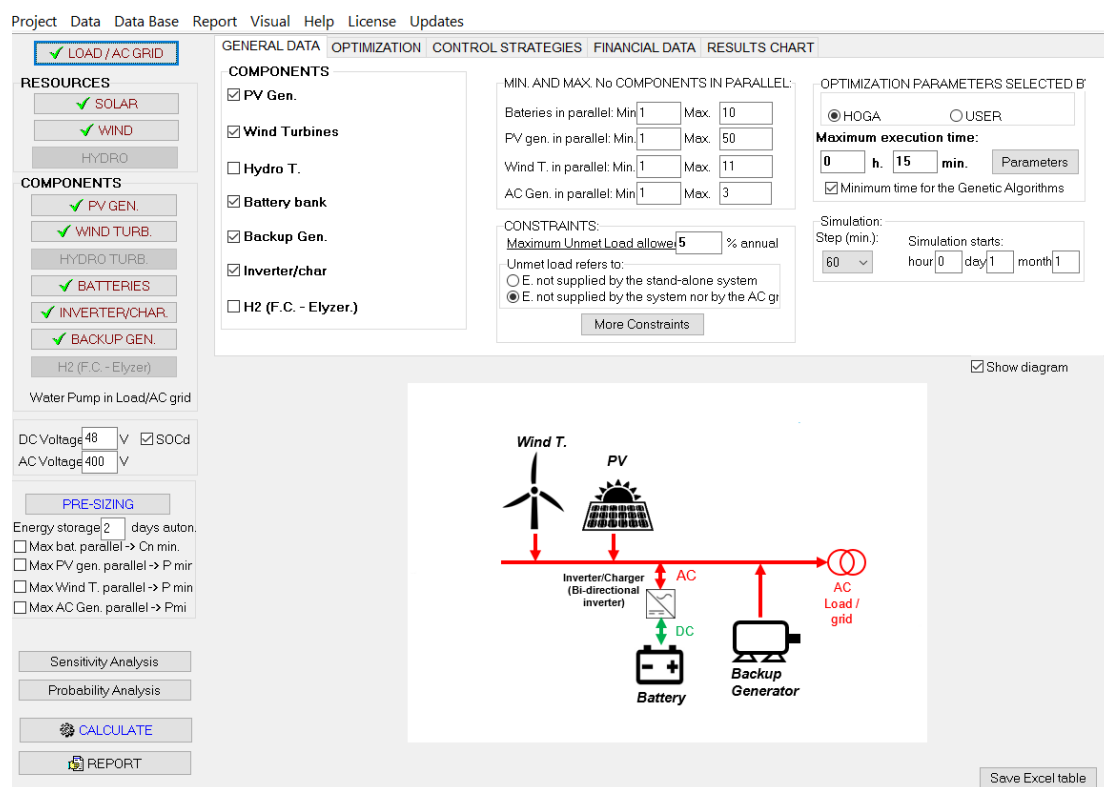
2.2 Prehľad programu iHOGA PRO+

iHOGA je jednou z verzií softvéru Hybridná optimalizácia pomocou genetických algoritmov (HOGA), ktorý vyvinuli výskumníci z Univerzity v Zaragoze (Španielsko) pod vedením hlavného výskumníka Dr. Rodolfa Dufa Lópeza. Na vytvorenie tohto softvéru sa používa jazyk C++. Používa sa na simuláciu a optimalizáciu hybridných systémov obnoviteľných zdrojov, ktoré využívajú kombináciu obnoviteľných zdrojov, ako je slnečná, veterná a vodná energia ako aj možnosti skladovania, ako sú batérie, záložné generátory a palivové články (v kombinácii alebo bez elektrolyzéra a vodíkovej nádrže).⁹

S týmto nástrojom existuje možnosť pripojenia k sieti a výroby vodíka na predaj. Veľkosť systému, ktorý sa má optimalizovať v systéme iHOGA, musí byť do 5 MW. Pre vyšší výkon je možné použiť softvér MHOGA (iná verzia HOGA). Na efektívne určenie optimálnej kombinácie komponentov a riadiacich stratégií sa používajú genetické algoritmy.

V systéme iHOGA možno dosiahnuť dve formy optimalizácie:

Minimalizáciu NPC pre scenáre s vysokým zaťažením alebo pre systémy mimo siete (autonómne) alebo maximalizáciu NPV pre systémy výroby elektrickej energie pripojené k sieti. Umožňuje nám definovať rôzne prípady Net Metering a Net Billing. Softvér obsahuje pokročilé funkcie, ako je viacperiodová simulácia a optimalizácia, zohľadnenie zmien v zaťažení a výrobe elektriny z obnoviteľných zdrojov počas životnosti systému, viacúčelová optimalizácia, simulácia v časových krokoch od 1 minúty do 1 hodiny, analýza citlivosti, analýza pravdepodobnosti pomocou simulácie Monte Carlo a ďalšie. iHOGA sa dá používať aj pre rôzne aplikácie (ako je uvedené v prípade HOMER Pro) len s jednou platenou licenciou (t. j. verzia PRO+). Verzia EDU je však bezplatná pre školiace alebo vzdelávacie oblasti s podmienkami. Podrobné informácie o tomto softvéri možno nájsť v ich používateľskej príručke.



Obrázok 2 Grafické rozhranie softvéru iHOGA PRO+ zobrazujúce rôzne záložky a architektúru simulovaného systému

⁹ Dufo R (2022). Používateľská príručka IHOGA. [Online]. Dostupné na: https://ihoga.unizar.es/Desc/iHOGA_User_manual.pdf

Požiadavky na vstupné údaje softvérového nástroja iHOGA PRO+ na návrh autonómnej mikrosiete (s fotovoltaikou, dieselovým generátorom, batériou a záťažou) sú uvedené v **prílohe A**.

2.3 Vybrané kritériá na porovnanie sw nástrojov a ich definícia

Na hodnotenie vybraných softvérových nástrojov (t. j. HOMER Pro a iHOGA PRO+) pre návrh autonómnych mikrosietí možno zvoliť nasledujúce kritériá, ako je uvedené v tabuľke 2. Jeho definícia je uvedená za ním. Tieto kritériá sú založené najmä na funkčnosti, prenosnosti, použiteľnosti, podpore, licenciách a cenách, efektívnosti a použitej optimalizačnej technike.¹⁰ V tabuľke 2 sú uvedené dva typy kritérií rozdelené od 1 do 8 a od 9 do 22 a zvýraznené rôznymi farbami. Kritériá od 1 - 8 budú porovnávať softvérové nástroje na základe jednotlivých prípadových štúdií. Na spravodlivé porovnanie budú v každom softvérovom nástroji na simuláciu každého prípadu zadané rovnaké vstupy a obmedzenia, ako sú vlastné v tabuľke 3. Potom sa na zber údajov na hodnotenie softvéru použijú kritériá 1 - 8 (z tabuľky 2). Zatiaľ čo kritériá 9 - 22 sa použijú len na porovnanie softvérových nástrojov, nezávisle od diskutovaných prípadových štúdií, a v tejto správe budú uvedené len raz.

Tabuľka 2 Vybrané kritériá pre porovnanie softvérových nástrojov 11, 12

Č.	Kritériá	Parametre	Jednotka	SW nástroje	
				HOME PRO	iHOGA PRO+
1	Optimálna veľkosť systému samostatnej mikrosiete (pre rovnaké vstupy) navrhnutá jednotlivými softvérmi	Batériový menič	kVA		
		PV	kW		
		Vietor	kW		
		DG	kW		
		Kapacita batérie	kWh		
2	Elektrická sieť	Elektrická energia vyrobená z fotovoltaiky	kWh/rok		
		Elektrická energia vyrobená z WT	kWh/rok		
		Elektrická energia vyrobená z DG	kWh/rok		
		Prebytočná elektrická energia	kWh/rok v %		
		Neuspokojené elektrické zaťaženie	v %		
3	Autonómia		hr		
4	Ekonomika	NPC	\$		
		Vyrovnané náklady na energiu	\$/kWh		
		Prevádzkové náklady	\$		
		Počiatkové kapitálové náklady	\$		
		Náklady na palivo	\$		
5	Faktory využívania OZE	Podiel obnoviteľných zdrojov	%		
		Obnoviteľná časť	%		
		Pokrytie zaťaženia OZE	%		
6	Emisie	CO ₂	kg/rok		
		SO ₂	kg/rok		

¹⁰ A. Kumar, A. R. Singh, Y. Deng, X. He, P. Kumar a R. C. Bansal, "Integrated assessment of a sustainable microgrid for a remote village in hilly region" (Integrované posúdenie udržateľnej mikrosiete pre odľahlú dedinu v kopcovitom regióne), Energy Conversion and Management, vol. 180, pp. 442-472, 2019

¹¹ T. Illes, A. Herrmann, B. Paech a J. Rückert, "Criteria for Software Testing Tool Evaluation – A Pohľad orientovaný na úlohy," 2005

¹² M. Jackson, S. Crouch a R. Baxter, "Software Evaluation: Criteria-based Assessment," Inštitút pre udržateľnosť softvéru, 2011

Č.	Kritériá	Parametre	Jednotka	SW nástroje	
				HOME PRO	iHOGA PRO+
		NO ₂	kg/rok		
7	Sociálna sieť	Vytváranie pracovných miest Index ľudského rozvoja			
8	Simulácia	Čas	sek		
9	Optimalizácia - použitá technika				
10	Optimalizácia - typ				
11	Typ zaťaženia				
12	Požiadavky na HW na inštaláciu SW	Požadovaný operačný systém			
		Procesor			
		RAM			
		Pevný disk			
13	Zložitosť súborov projektu	Obsahuje projekt jednotlivé/násobné súbory?			
14	Používateľské rozhranie	Vzhľad a ovládanie SW - karta Domov			
		Prezentácia informácií v každom karta			
		Ako ľahko/ťažko sa dostanete k informáciám na minimálnom počte kariet?			
15	Možnosť inštalácie	Je inštalácia jednoduchá?			
16	Učiteľnosť	Príručka na začatie používania obsahuje základný príklad používania SW			
		Pokyny sú k dispozícii pre mnohé základné prípady použitia			
		K dispozícii sú výukové videá či už bezplatné, alebo platené			
		Referenčné príručky sú k dispozícii pre všetky možnosti konfigurácie			
		Ako jednoduché je naučiť sa základnú funkčnú úlohu?			
		Ako jednoduché je naučiť sa pokročilú funkčnú úlohu?			
17	Licencie a ceny	Bezplatné alebo komerčné - Typy licencií (Std./Acad./Pro.)			
		Cenové rozpätie rôznych licencií - typy			
		Trvanie licencie -			
		Je k dispozícii bezplatná skúšobná verzia?			
18	Interoperabilita	Interoperabilita s inými potrebný/súvisiaci softvér ?			
19	Dostupnosť vstupných údajov	Údaje o vetre -			
		Údaje o slnečnom žiarení -			
		Údaje o teplote			
		Profil Laod -			
20	Prístupnosť výstupných súborov	Formát výsledkov -			
		Dokáže vygenerovať správu o optimalizovanom dizajne?			

Č.	Kritériá	Parametre	Jednotka	SW nástroje	
				HOME PRO	iHOGA PRO+
		Dokáže zobrazit' rôzne grafy v samotný softvérový nástroj ?			
		Vygenerovať cenovú ponuku pre projekt?			
21	Frekvencia aktualizácie softvéru a manuál				
22	Zariadenie zákazníckej podpory SW	Reagujú vývojári na e-mail/volania?			
		Akceptujú spätnú väzbu?			
		Jednoduché/rýchle získanie technickej podpory od vývojára na akýkoľvek problém týkajúci sa funkčnosti/fungovania nástroja.			

Tabuľka 3 Vstupné požiadavky softvérových nástrojov.

Č.	Vstupné parametre	Jednotka	SW nástroje		Komentáre
			HOME PRO	iHOGA PRO+	
1	Špecifikácia jedného PV modulu				
2	Ukladanie jednej batérie špecifikácia				
3	Špecifikácia jednej veternej turbíny				
4	Špecifikácia DG				
5	Vlastností paliva				
6	Špecifikácia jedného meniča				
7	Ekonomické vstupy				
8	Parametre regulátora systému				
9	Profil slnečného žiarenia				mesačne priemer
10	Teplotný profil				mesačne priemer
11	Profil zaťaženia				mesačne priemer
12	Profil vetra				mesačne priemer
13	Umiestnenie lokality				
14	Obmedzenia: Minimálny počet obnoviteľných zdrojov - frakcia				
15	Obmedzenia: Nesplnené zaťaženie (max.)				

2.4 Definícia kritérií

Definícia kritérií uvedených v tabuľke 2 je predmetom tejto časti. Porovnáva sa Optimálna veľkosť mikrosiete navrhnutá jednotlivými softvérovými nástrojmi, Elektrické parametre, Autonómia batérovej banky, Ekonomika, Faktory využívania OZE, Emisie, Sociálna sieť, Čas simulácie, Použitá optimalizačná technika, Typ optimalizácie, Typ zaťaženia, Komplexnosť projektového súboru v každom softvérovom nástroji, Systémové požiadavky na inštaláciu softvérového nástroja, Používateľské rozhranie (UI) softvérového nástroja, Možnosť inštalácie, Učiteľnosť, Licencovanie a ceny,

Interoperabilita, Dostupnosť vstupných údajov pomocou softvérového nástroja, Prístupnosť výstupného súboru, Frekvencia aktualizácie softvéru a príručky, Zariadenie zákazníckej podpory softvéru.¹³

2.4.1 Prípadové štúdie samostatných mikrosietí

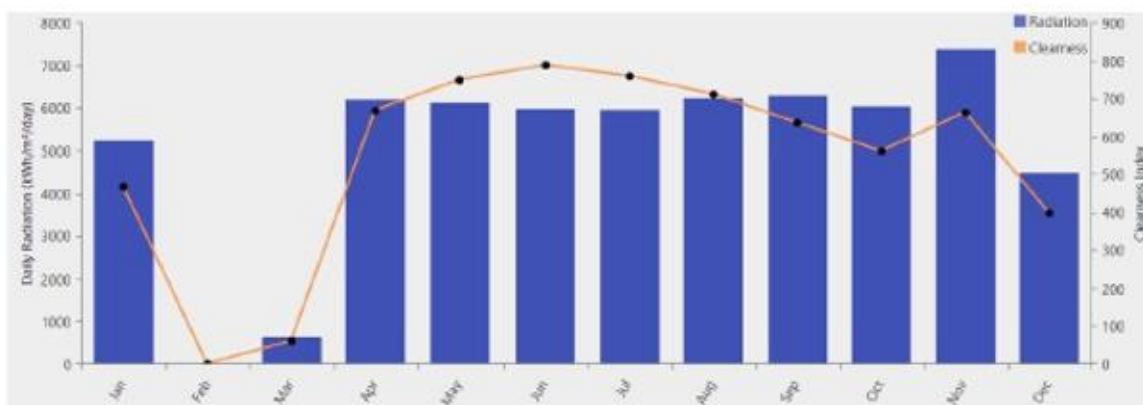
Medzi faktory, ktoré môžu ovplyvniť porovnanie rôznych softvérových nástrojov pri formulovaní prípadových štúdií pre návrh autonómnych mikrosietí, patria:

- a) zaťaženie,
- b) zdroj výroby (fotovoltaika, vietor, nafta atď.),
- c) skladovanie energie,
- d) počasie.

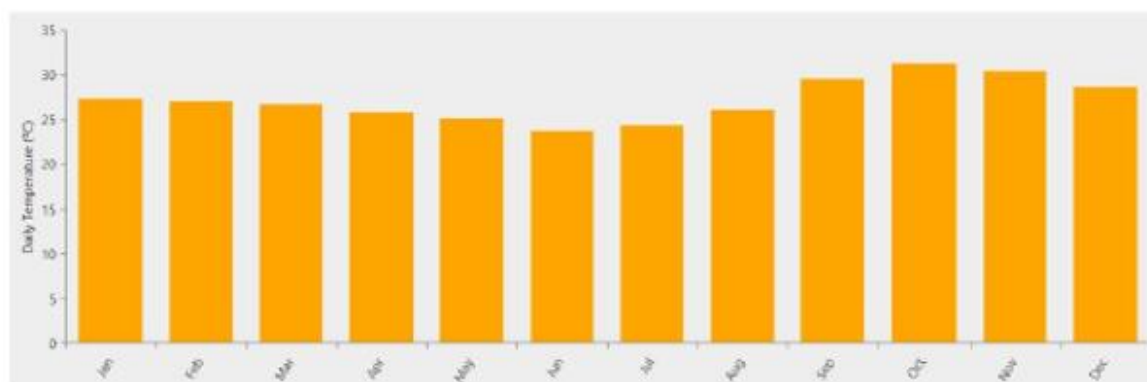
Preto je pre účinnejšie porovnanie softvérových nástrojov nevyhnutné zohľadniť vyššie uvedené faktory pre rôzne geografické lokality. To sa dá dosiahnuť formulovaním prípadových štúdií pre rôzne geografické oblasti. Vedecký tím použil lokácie v Austrálii, Holandsku a Spojených štátoch. Okrem toho bol cieľ každej prípadovej štúdie nasledovný: *Zvýšiť úspory nafty o viac ako 50 % využitím obnoviteľného zdroja výroby a batériového úložiska na uspokojenie dopytu po záťaži pri optimalizovaných nákladoch na elektrinu.*¹⁴

2.4.2 Simulácia Austrália

Skutočný solárny profil lokality Daly River v roku 2018 (snímka programu HOMER Pro)



Teplotný profil lokality Daly River získaný z programu HOMER Pro. (snímka z programu HOMER Pro)

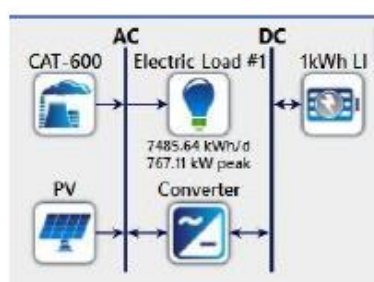


¹³ IEA – PVPS, Task 18 (04/2023)

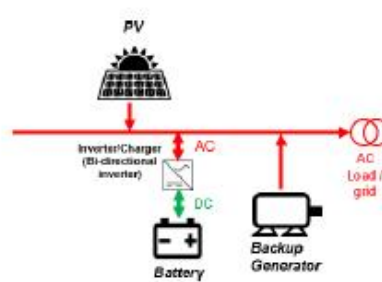
¹⁴ IEA – PVPS, Task 18 (04/2023)

Tabuľka 4 Špecifikácie rôznych komponentov použitých na obrázku 3 a vstupy pre softvérové nástroje

Č.	Vstupné parametre	Jednotka	SW nástroje		Komentáre
			HOME PRO	iHOGA PRO+	
1	Špecifikácia PV modulu	kW	1024	1024	JA Solar 320
2	Špecifikácia batériového úložiska	kWh	1987	1987	
3	Špecifikácia DG	kW	560	560	Caterpillar 3512
4	Vlastnosti paliva		Nafta 1,5 USD/L	Nafta 1,5 USD/L	
5	Špecifikácia meniča	kW	800	800	
6	Ekonomické vstupy	%	2,4	2,4	reálna diskontná sadzba
7	Parametre regulátora systému		Zaťaženie na stránke	Zaťaženie na stránke	Príloha B
8	Profil slnečného žiarenia		obrázok 9	obrázok 9	mesačne priemer
9	Teplotný profil		obrázok 10	obrázok 10	mesačne priemer
10	Profil zaťaženia		obrázok 4	obrázok 4	mesačne priemer
11	Umiestnenie lokality		Daly River, Austrália (13 st, 42,7 m j. š.; 130 st 41,3 m v. d.)	Daly River, Austrália (13,751 j. š.; 130,689 v. d.)	
12	Obmedzenia: Minimálny počet obnoviteľná časť	%	10	10	
13	Obmedzenia: Neuspokojivé zaťaženie (max.)	%	5	5	



(a)



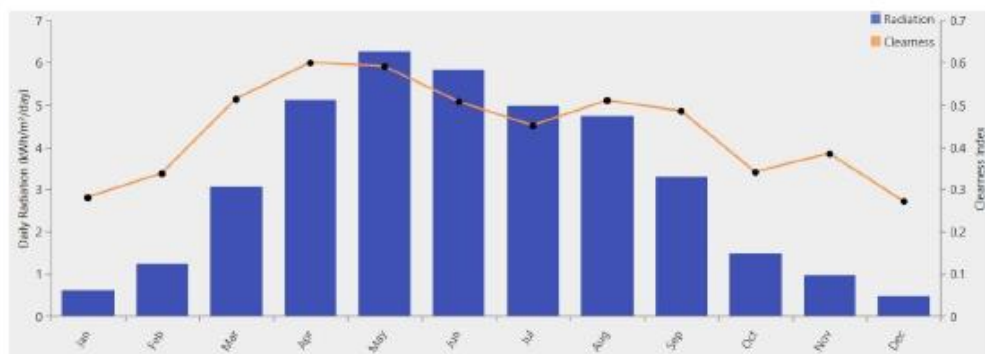
(b)

Obrázok 3 Ilustrácia architektúry z obrázka 3 (t. j. prípad 1) viditeľná v (a) HOMER Pro a (b) iHOGA PRO+

Tabuľka 5 Porovnanie optimálnych riešení získaných z programu HOMER Pro a iHOGA PRO+ pre prípad 1

Č.	Kritériá	Parametre	Jednotka	SW nástroje	
				HOME PRO	iHOGA PRO+
1	Optimálna veľkosť systému samostatnej mikrosiete (pre rovnaké vstupy) navrhnutá jednotlivými softvérmí	Batériový menič	kV	800	800
		PV	kW	1024	1024
		DG	kW	560	560
		Kapacita batérie	kWh	1987	1987
2	Elektrická sieť	Elektrická energia vyrobená z fotovoltaiiky	kWh/rok	1333808	1330048
		Elektrická energia vyrobená z WT	kWh/rok	1529062	1526543
		Prebytočná elektrická energia	kWh/rok	60292	50725
			v %	2,11	1,78
		Neuspokojené elektrické zaťaženie	kWh/rok	11449	11452
		v %	0,419	0,42	
3	Autonómia		hr	6,05	6,05
4	Ekonomika	NPC	\$	17635306	14376422
		Vyrovnané náklady na energiu	\$/kWh	0,3454	0,28
		Prevádzkové náklady	\$	1227236	1259369
		Počiatkové kapitálové náklady	\$	2183520	2183520
		Náklady na palivo	\$	12895096	9642101
5	Faktory využívania OZE	Podiel obnoviteľných zdrojov	%	49	48,9
		Obnoviteľná časť	%	43,8	43,7
		Pokrytie zaťaženia OZE	%		
6	Emisie	CO ₂	kg/rok	1198082	1596310
		SO ₂	kg/rok	2936	
		NO ₂	kg/rok	652	
		Index ľudského rozvoja			
7	Simulácia	Čas	sek	2	<1

2.4.3 Simulácia Holandsko



Obrázok 4 Solárny profil mesta Delft na rok 2021. Údaje sú prevzaté z programu iHOGA Pro a použité v programe HOMER Pro (snímka programu HOMER Pro)



Profil rýchlosti vetra v Delfte získaný z iHOGA PRO+.

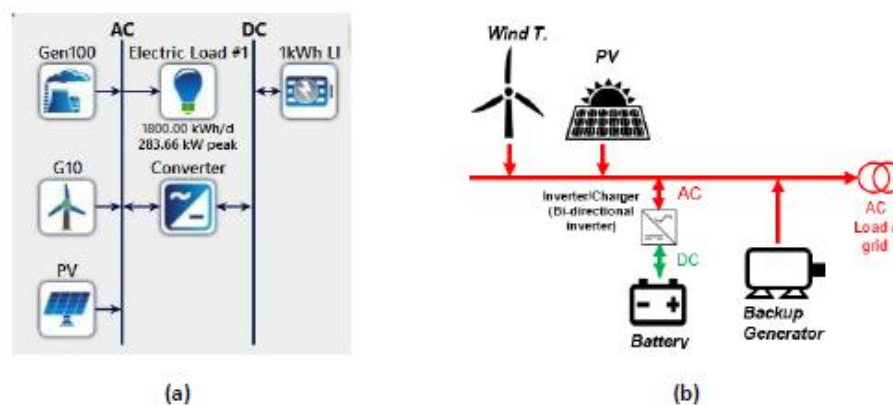


Teplotný profil mesta Delft získaný z programu HOMER Pro.

Tabuľka 6 Špecifikácie rôznych komponentov použitých na obrázku 5 a vstupy pre softvérové nástroje

Č.	Vstupné parametre	Jednotka	SW nástroje		Komentáre
			HOME PRO	iHOGA PRO+	
1	Špecifikácia PV modulu	kW	1	1	Generic
2	Špecifikácia batériového úložiska	kWh	6,9	6,9	Samsung
3	Špecifikácia veternej turbíny	kW	10	10	Generic
4	Špecifikácia DG	kW	10-60	10-30	Vlastná veľkosť
5	Vlastnosti paliva		Nafta; 1,5 USD/L	Nafta; 1,5 USD/L	
6	Špecifikácia meniča	kW	1	1	Generic
7	Ekonomické vstupy	%	2,1	2,1	Reálna diskontná sadzba
8	Parametre regulátora systému		zaťaženie	zaťaženie	Príloha C
9	Profil slnečného žiarenia		Obrázok 12	Obrázok 12	Mesačný priemer
10	Profil rýchlosti vetra		Obrázok 13	Obrázok 13	
11	Teplotný profil		Obrázok 14	Obrázok 14	Mesačný priemer

Č.	Vstupné parametre	Jednotka	SW nástroje		Komentáre
			HOME PRO	iHOGA PRO+	
12	Profil zaťaženia		Obrázok 6	Obrázok 6	Mesačný priemer
13	Umiestnenie lokality		Delft, (44 st, 52,9 m s. š.; 72 st 56,6 m z. d.)	Delft, (44,869 s. š.; 72,935 st. z. d.)	
14	Obmedzenia: Minimálny počet obnoviteľná časť	%	50	50	
15	Obmedzenia: Neuspokojivé zaťaženie (max.)	%	5	5	



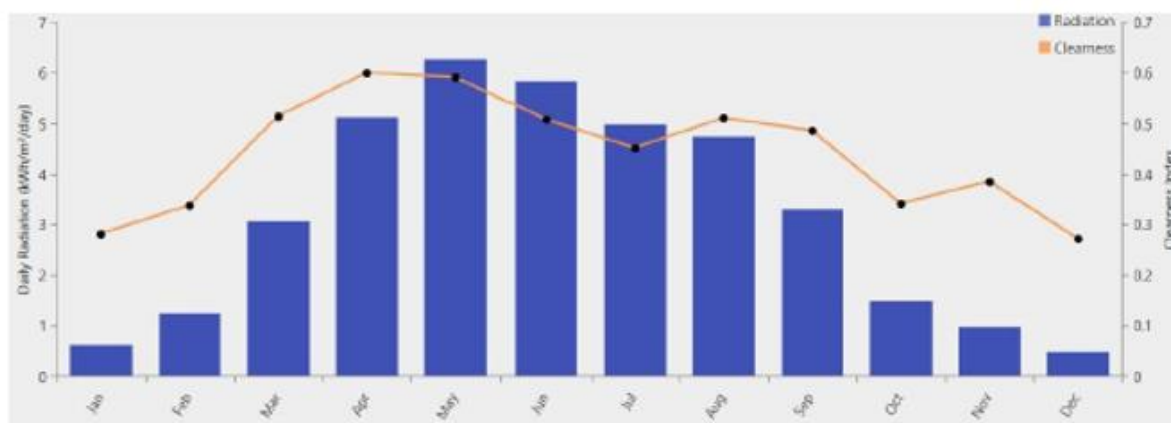
Obrázok 5 Ilustrácia architektúry z Obrázka 5 (t. j. prípad 2) viditeľná v (a) HOMER Pro a (b) iHOGA PRO+

Tabuľka 7 Porovnanie optimálnych riešení získaných z programu HOMER Pro a iHOGA PRO+ pre prípad 2

Č.	Kritériá	Parametre	Jednotka	SW nástroje	
				HOME PRO	iHOGA PRO+
1	Optimálna veľkosť systému samostatnej mikrosiete (pre rovnaké vstupy) navrhnutá jednotlivými softvérmi	Batériový menič	kV	146	150
		PV	kW	381	300
		Vietor	kW	760	480
		DG	kW	50	70
		Kapacita batérie	kWh	1290,3	835
2	Elektrická sieť	Elektrická energia vyrobená z fotovoltaiiky	kWh/rok	373997	304273
		Elektrická energia vyrobená z WT	kWh/rok	840538	518168
		Prebytočná elektrická energia	kWh/rok	602188	278791
			v %	47,4	29,79
		Neuspokojené elektrické zaťaženie	kWh/rok	33445	32777
	v %	5,09	4,99		
3	Autonómia		hr	13,8	8,9
4	Ekonomika	NPC	\$	2798027	2249457

Č.	Kritériá	Parametre	Jednotka	SW nástroje	
				HOME PRO	iHOGA PRO+
		Vyrovnané náklady na energiu	\$/kWh	0,2299	0,18
		Prevádzkové náklady	\$	282923	196413
		Počiatkové kapitálové náklady	\$	1507391	1057960
		Náklady na palivo	\$	562575	668091
5	Faktory využívania OZE	Podiel obnoviteľných zdrojov	%	195	132
		Obnoviteľná časť	%	91,1	77,76
		Pokrytie zaťaženia OZE	%		
6	Emisie	CO ₂	kg/rok	50793	133770
		SO ₂	kg/rok	123	
		NO ₂	kg/rok	27,4	
		Index ľudského rozvoja			
7	Simulácia	Čas	sek	400	170

2.4.4 Simulácia USA



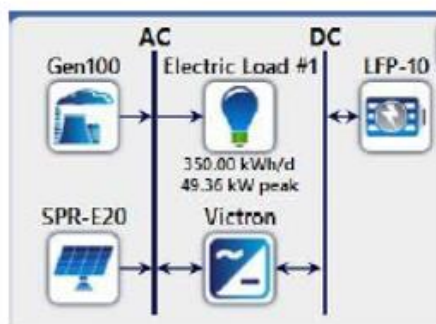
Slnecný profil Sheldonu na rok 2021 získaný z iHOGA PRO+. (snímka HOMER Pro).



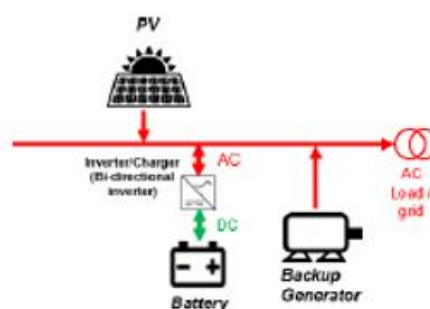
Teplotný profil mesta Sheldon na rok 2021 získaný z programu HOMER Pro.

Tabuľka 8 Špecifikácie rôznych komponentov použitých na Obrázku 7 a vstupy pre softvérové nástroje

Č.	Vstupné parametre	Jednotka	SW nástroje		Komentáre
			HOME PRO	iHOGA PRO+	
1	Špecifikácia PV modulu	kW	327	327	SunPower E20-327
2	Špecifikácia batériového úložiska	kWh	9,6	9,6	Pevnosť Power LFP-10
3	Špecifikácia DG	kW	0-30	10/20/30	Veľkosť podľa vlastného výberu
4	Vlastnosti paliva		Nafta; 1,5 USD/L	Nafta; 1,5 USD/L	
5	Špecifikácia meniča	kW	10	10	Victron Quattro
6	Ekonomické vstupy	%	2,1	2,1	Reálna diskontná sadzba
7	Parametre regulátora systému		zaťaženie	Zaťaženie	Príloha D
8	Profil slnečného žiarenia		Obrázok 16	Obrázok 16	Mesačný priemer
9	Teplotný profil		Obrázok 17	Obrázok 17	Mesačný priemer
10	Profil zaťaženia		Obrázok 8	Obrázok 8	Mesačný priemer
11	Umiestnenie lokality		Sheldon, VT, USA (44 st, 54,6 m s. š.; 72 st 98,1 m z. d.)	Sheldon, VT, USA (44.9105 s. š.; 72.9795 st. z. d.)	
12	Obmedzenia: Minimálny počet obnoviteľná časť	%	50	50	
13	Obmedzenia: Neuspokojivé zaťaženie (max.)	%	5	5	



(a)



(b)

Ilustrácia architektúry z obrázka 7 (t. j. prípad 3) v (a) HOMER Pro a (b) iHOGA

Tabuľka 9 Porovnanie optimálnych riešení získaných z programu HOMER Pro a iHOGA PRO+ pre prípad 3

Č.	Kritériá	Parametre	Jednotka	SW nástroje	
				HOME PRO	iHOGA PRO+
1	Optimálna veľkosť systému	Batériový menič	kV	41,2	40
		PV	kW	155	110

Č.	Kritériá	Parametre	Jednotka	SW nástroje	
				HOME PRO	iHOGA PRO+
	samostatnej mikrosiete (pre rovnaké vstupy) navrhnutá jednotlivými softvérmi	DG	kW	10	10
		Kapacita batérie	kWh	268,8	211
2	Elektrická sieť	Elektrická energia vyrobená z fotovoltaiky	kWh/rok	175797	132841
		Elektrická energia vyrobená z WT	kWh/rok	14317	17552
		Prebytočná elektrická energia	kWh/rok	63993	16870
			v %	33,3	
		Neuspokojené elektrické zaťaženie	kWh/rok	5166	3517
v %	4,04		2,75		
3	Autonómia		hr	15,7	
4	Ekonomika	NPC	\$	566541,2	428645
		Vyrovnané náklady na energiu	\$/kWh	0,2368	0,18
		Prevádzkové náklady	\$	17554	
		Počiatkové kapitálové náklady	\$	269513,4	203792
		Náklady na palivo	\$	12349,81	90680
5	Faktory využívania OZE	Podiel obnoviteľných zdrojov	%	145	108
		Obnoviteľná časť	%	88,3	87,8
		Pokrytie zaťaženia OZE	%		
6	Emisie	CO ₂	kg/rok	11032	18930
		SO ₂	kg/rok	27	
		NO ₂	kg/rok	6	
7	Sociálna sieť	Vytváranie pracovných miest			
		Index ľudského rozvoja			
8	Simulácia	Čas	sek	28	150

2.5 Hodnotenie softvérových nástrojov

V tejto časti sa hodnotenie vybraných softvérových nástrojov vykonáva na základe kritérií 9 - 22, ako je uvedené v tabuľke 2. V tabuľke 10 sa porovnáva HOMER Pro a iHOGA+ na základe uvedených kritérií v tabuľke 2 a doplnili sme údaje za vytvorenú aplikáciu MicroGridBox® portál. V schopnostiach týchto nástrojov však boli pozorované určité rozdiely, ktoré sú uvedené nižšie:

Tabuľka 10 Porovnanie programov HOMER Pro a iHOGA PRO+ a MicroGridBox® portálu na základe kritérií uvedených v tabuľke 2.15

¹⁵ IEA – PVPS, Task 18 (04/2023)

Č.	Kritériá	Parametre	SW nástroje		MicroGridBox®
			HOME PRO	iHOGA PRO+	
9	Typ zaťaženia		AC, DC, tepelné, vodík	AC, DC, tepelné, voda, vodík	-
10	Typ optimalizácie		Monoobjektívne (ekonomické)	multiobjektívne až 3 ciele (ekonomický, elektrický, emisie CO ₂ , HDI).	--
11	Použitá optimalizačná jednotka		algoritmus prehľadávania mriežky (Space search) a algoritmus bez derivátov (HOMER Optimizátor)	genetické algoritmy	-
12	Požiadavky na HW na inštaláciu SW	Požadovaný operačný systém	Windows (7 alebo novšie), odporúča sa 10	System Windows (XP alebo vyšší)	Windows 10, Mac OS alebo Linux
		Procesor	intel core i3 alebo ekvivalent	-	-
		RAM	2 GB	-	-
		Pevný disk	450 MB	-	-
13	Zložitosť súborov projektu	Obsahuje projekt jednotlivé/násobné súbory?	jeden	viacnásobné	jeden
14	Používateľské rozhranie	Vzhľad a ovládanie SW - karta Domov	vynikajúce	dobré	vynikajúce
		Prezentácia informácií v každom karta	jednoduché	komplexné	jednoduché
		Ako ľahko/ťažko sa dostanete k informáciám na minimálnom počte kariet?	zložito	jednoducho	jednoducho
15	Možnosť inštalácie	Je inštalácia jednoduchá?	áno	áno	nie je potrebná
16	Učiteľnosť	Príručka na začatie používania obsahuje základný príklad používania SW	Nie	nie	nie
		Pokyny sú k dispozícii pre mnohé základné prípady použitia	áno	áno	nie
		K dispozícii sú výukové videá či už bezplatné alebo platené	áno	Nie	nie
		Referenčné príručky sú k dispozícii pre všetky možnosti konfigurácie	áno	áno	nie

Č.	Kritériá	Parametre	SW nástroje		MicroGridBox®
			HOME PRO	iHOGA PRO+	
		Ako jednoduché je naučiť sa základnú funkčnú úlohu?	Veľmi jednoduché	Komparátory ťažké	
		Ako jednoduché je naučiť sa pokročilú funkčnú úlohu?	jednoduché	Komparátory ťažké	jednoduché
17	Licencie a ceny	Bezplatné alebo komerčné -Typy licencií (Std./Acad./Pro.)	Komerčné Std, Acad. A Pro	Zadarmo a komerčné EDU (zadarmo) a PRO+	Zadarmo
		Cenové rozpätie rôznych licencií - typy	125 USD, 249 USD, 379 USD	500 € (pol roka), 2800€ (trvalo)	-
		Trvanie licencie -	Mesačne a raz ročne	6 – 12 mesiacov, natrvalo	-
		Je k dispozícii bezplatná skúšobná verzia?	áno	áno	áno
18	Interoperabilita	Interoperabilita s inými potrebný/súvisiaci softvér ?	Áno (s MATLABom na modul prepojenia)	nie	nie
19	Dostupnosť vstupných údajov	Údaje o vetre -	áno	áno	nie
		Údaje o slnečnom žiarení -	áno	áno	áno
		Údaje o teplote	áno	áno	nie
		Profil Laod -	áno	áno	nie
20	Prístupnosť výstupných súborov	Formát výsledkov -	XLSX, WORD, PDF, HTML	XLSX, PDF, RTF	HTML
		dokáže vygenerovať správu o optimalizovaný dizajn?	áno	áno	áno
		dokáže zobrazit' rôzne grafy v samotný softvérový nástroj ?	áno	áno	áno
		vygenerovať cenovú ponuku pre projekt?	áno	nie	nie
21	Frekvencia aktualizácie softvéru a manuálu		nepravidelné	časté	nepravidelné
22	Zariadenie zákazníckej podpory SW	reagujú vývojári na e-mail/ volania?	áno	áno	áno
		počúvajú spätnú väzbu?	áno	nie	áno
		jednoduché/rýchle získanie technickej podpory od vývojára na akýkoľvek problém týkajúci sa funkčnosti/fungovania nástroja	Jednoduché, ale pomalé	Veľmi jednoduché a rýchle	jednoduché

3 DIGITÁLNE DVOJČA

Batériová kalkulačka (podrobne popísaná v míľniku č. 4 aktivity č. 6) používa na výpočet optimalizácie metódu mriežkového vyhľadávania parametrov. Táto metóda je vhodná pri prehľadávaní menšieho priestoru možných riešení. Ak sa vyhľadávajú dva parametre, priestor možných riešení je možné definovať dvojrozmernou maticou. V prípade ak tri parametre, priestor je možné definovať trojrozmernou maticou. Výpočtová náročnosť algoritmu sa zvyšuje s nárastom počtu prvkov v matici hľadaného priestoru, čo znamená, že sa zvyšuje rozsah možných hodnôt napríklad matica, ktorá má 10 riadkov a 10 stĺpcov môže, vzhľadom na nárast počtu možných riešení, mať aj 1 000 riadkov a 1 000 stĺpcov. V takomto prípade sa čas hľadania optimálnej kombinácie dvoch parametrov exponenciálne predĺži. Riešením tohto problému môže byť aplikácia náhodného hľadania (*ang. random search*) do batériovej kalkulačky, prípadne ďalších metód popísaných v tejto kapitole. Táto kapitola poskytuje náčrt možných riešení pri optimalizácii pomocou batériovej kalkulačky.

3.1 Mriežkové a náhodné vyhľadávanie

Mriežkové vyhľadávanie (*ang. grid search*) je algoritmus vyhľadávania v dátových štruktúrach, ktorý je používaný na efektívne lokalizovanie alebo vyhľadávanie prvkov v priestore. Je to jednoduchý, ale výkonný spôsob, ako organizovať a vyhľadávať prvky v dátach s viacdimensionálnym priestorom.

Idea mriežkového vyhľadávania spočíva v tom, že priestor je rozdelený na pravidelnú mriežku (*ang. grid*), kde každá bunka mriežky zodpovedá určitému regiónu v priestore. Každý prvok v dátach je potom priradený do príslušnej bunky mriežky na základe svojej polohy.

Pri vyhľadávaní sa najprv identifikuje bunka mriežky, v ktorej by sa mohol nachádzať hľadaný prvok. Následne sa vykoná prehľadávanie prvkov v tejto bunke a prípadne aj v jej susedných bunkách, ak je potrebné. Tým sa získava podmnožina prvkov, medzi ktorými je možné nájsť hľadaný prvok. V konečnom dôsledku sa aplikuje presné porovnanie alebo ďalšie vyhľadávacie metódy na túto podmnožinu prvkov.

Mriežkové vyhľadávanie je jednoduché a intuitívne, a preto je často používané v rôznych oblastiach, kde je potrebné rýchle vyhľadávanie v priestorových dátach. Patrí sem napríklad počítačová grafika, geografické informačné systémy, detekcia kolízií, simulácie a ďalšie aplikácie, kde je potrebné spracovávať a vyhľadávať prvky v rozmerovom priestore.

Je dôležité poznamenať, že mriežkové vyhľadávanie je jednou z možností vyhľadávania v priestore a existujú aj ďalšie techniky a dátové štruktúry, ktoré sa používajú na riešenie podobných problémov, ako sú oktetové stromy (octrees), kd-stromy, BSP stromy a podobne. Voľba správnej metódy závisí od konkrétneho problému, veľkosti dát, požadovanej presnosti a výkonu aplikácie.

3.1.1 Opis algoritmov náhodného vyhľadávania

Stochastické metódy sú metódy, ktoré obsahujú niektoré stochastické prvky. To znamená, že buď je výsledok metódy náhodnou premennou, alebo sa samotná účelová funkcia považuje za realizáciu stochastického procesu. Preto sa obetuje možnosť absolútnej záruky úspechu. Namiesto toho je zvyčajne možné dokázať, že s rastúcim úsilím do nekonečna sa nájde prvok *B* alebo *S* s pravdepodobnosťou jedna.

Metódy náhodného hľadania sú tie stochastické metódy, ktoré sa spoliehajú výlučne na náhodný výber postupnosti bodov v realizovateľnej oblasti problému podľa určitého vopred špecifikovaného rozdelenia pravdepodobnosti alebo postupnosti rozdelení pravdepodobnosti.

Tieto metódy sú použiteľné na veľmi širokú triedu problémov a majú zaručenú asymptotickú výkonnosť. Preto sa tieto metódy v poslednom desaťročí tešia rastúcemu záujmu pre svoju schopnosť riešiť problémy, ktorých matematická štruktúra je zložitá (alebo nežiaduca, alebo dokonca nemožné) analyzovať.

Koncepčné metódy

Metódy, o ktorých sa tu hovorí, majú koncepčný charakter v tom zmysle, že v súčasnosti neexistuje efektívna implementácia týchto metód. Teoretické výsledky, ktoré možno pre tieto metódy získať, sú

však zaujímavé samy o sebe. Navyše sa ukázal ich potenciál inšpirovať (alebo teoreticky podporiť) praktické algoritmy globálnej optimalizácie.

(Čisté) náhodné vyhľadávanie

Najjednoduchšou stochastickou metódou pre globálnu optimalizáciu je čisté náhodné vyhľadávanie (*ang. pure random search*). Táto metóda spočíva v generovaní postupnosti nezávislých a identicky rozdelených rovnomerných bodov v realizovateľnej oblasti S , pričom sa sleduje najlepší nájdený bod. V pseudokóde je táto metóda uvedená nižšie.

Postupnosť bodov vygenerovaná touto metódou konverguje ku globálnemu optimu s pravdepodobnosťou jedna. Konkrétne pravdepodobnosť, že sa bod v S_ϵ dosiahne počas prvých N iterácií, sa rovná

$$1 - (1 - \varphi(S_\epsilon))^N,$$

kde φ označuje rovnomerné rozdelenie na S . Inými slovami, táto metóda ponúka pravdepodobnostnú asymptotickú záruku.

Nižšie je uvedený pseudokód pre čisté náhodné vyhľadávanie.

```
PROCEDÚRA čisté náhodné vyhľadávanie ()
  Vstupná inštancia();
  Nastav  $y = -\infty$ ;
  VYKONAJ
  Vygenerujte bod  $x$  z rovnomerného rozdelenia
  V rámci  $S$ ;
  Nastav  $y = \max(y; f(x))$ ;
  VYKONÁVAJ POKIAL';
  vráť( $y$ );
  KONIEC čisto náhodné vyhľadávanie;
```

V histórii optimalizácií bola zavedená trieda náhodných vyhľadávacích metód, ktorých najdôležitejšou vlastnosťou v porovnaní s čisto náhodným vyhľadávaním je, že sú znemožnené zlepšenia prostredníctvom zovšeobecnenia prístupu odmietnutia prijatia. Navyše výber vzorky nie je obmedzený na rovnomerné rozdelenie, ale na (vopred špecifikovanú) postupnosť (absolútne spojité) rozdelení pravdepodobnosti na S . Asymptotická konvergencia s pravdepodobnosťou jedna sa dá opäť dokázať.

Ak je metóda náhodného hľadania adaptívna, t. j. ak dovolíme, aby rozdelenia, z ktorých sa bude vzorkovať, záviseli od hodnôt predtým nájdených riešení, potom konvergenciu vo všeobecnosti nemožno zabezpečiť.

Čisté adaptívne vyhľadávanie

Čisté adaptívne vyhľadávanie sa líši od čistého náhodného vyhľadávania a podobá sa metódam náhodného vyhľadávania tým, že si v každej iterácii vynúti zlepšenie. Zlepšujúca sila je však oveľa silnejšia v tom, že algoritmus je skutočne adaptívny, bez použitia akejkoľvek formy prijatia-odmietnutia. Konkrétne, bod iterácie sa generuje z rovnomerného rozdelenia na podmnožinu bodov, ktoré sa zlepšujú vzhľadom na predchádzajúce iteračné body. Formálnejšie táto metóda znie:


```

PROCEDÚRA čisté adaptívne vyhľadávanie ()
Vstupná inštancia();
Nastav  $y = -\infty$ ;
VYKONAJ
  Vygenerujte bod  $x$  z rovnomerného rozdelenia
  V rámci  $\{x \in S: f(x) > y\}$ ;
  Nastav  $y = f(x)$ ;
VYKONÁVAJ POKIAL';
vrát'(y);
KONIEC čisto adaptívne vyhľadávanie;

```

Táto metóda bola predstavená a analyzovaná v odbornej literatúre zameriavajúcej sa na problémy konvexného programovania a v prácach pojednávajúcich o všeobecnejších globálnych optimalizačných problémoch. Pre Lipschitzove spojité problémy s konvexnými realizovateľnými oblasťami sa očakávaný počet iterácií na dosiahnutie riešenia s danou presnosťou zväčšuje nanajvýš lineárne v dimenzii d problému. Tento výsledok naznačuje, že existuje nádej na efektívnu metódu náhodného hľadania pre globálnu optimalizáciu. V skutočnosti niekoľko metód náhodného vyhľadávania uviedlo empirickú linearitu v dimenzii pre optimalizáciu kvadratických funkcií. Metóda čistého adaptívneho vyhľadávania bola rozšírená na prípad konečnej domény s analogickou výkonnosťou a s podobnými výsledkami.

Adaptívne vyhľadávanie

Vo všeobecnosti je veľmi ťažké vygenerovať bod rovnomerne rozložený v množine úrovní globálneho optimalizačného problému. Metóda, ktorá sa tomuto problému vyhýba za cenu toho, že sa musí generovať z iných rozdelení ako rovnomerné, je adaptívne vyhľadávanie (AV). V tejto metóde sa postupnosť zlepšujúcich sa bodov generuje generovaním bodov podľa postupnosti rozdelení v realizovateľnej oblasti S , pričom sa na dosiahnutie zlepšenia používa prístup prijatia a zamietnutia. Postupnosť generovania rozdelení by sa mala zvoliť tak, aby sa s postupom metódy tieto rozdelenia čoraz viac sústreďovali okolo globálneho optima problému. Príkladom rodiny rozdelení, ktoré majú túto vlastnosť, je rodina Boltzmannových rozdelení. Táto rodina rozdelení, parametrizovaná kladným parametrom T , je absolútne spojitá na S s hustotami

$$g_T(x) \propto e^{-f(x)/T}$$

Je možné si všimnúť, že keď sa parameter T blíži k nekonečnu, postupnosť rozdelení sa blíži k rovnomernému rozdeleniu na množine S . Na druhej strane možno ukázať, že keď sa parameter T blíži k nule, postupnosť rozdelení konverguje k rozdeleniu, ktoré sa sústreďuje na množinu bodov, kde je globálne optimum sa dosiahne optimum. Pri použití tejto rodiny rozdelení metóda adaptívneho hľadania môže byť vyjadrená:

```

PROCEDÚRA adaptívne vyhľadávanie ()
Vstupná inštancia();
Nastav  $y = -\infty$ ;
Nastav  $y = \infty$ ;
VYKONAJ
  Vygenerujte bod  $x$  z boltzmanového rozdelenia
  s parametrom  $T$  kým sa nenájde bod s  $f(x) > y$ ;
  Nastav  $y = f(x)$ ;
  Znížiť  $T$ ;
VYKONÁVAJ POKIAL';
vrát'(y);
KONIEC adaptívne vyhľadávanie;

```

Táto metóda bola zavedená s cieľom študovať správanie simulovaného žihania. Kľúčový výsledok pre čisté adaptívne vyhľadávanie bol zovšeobecný takt. Ak hodnota parametra T závisí monotónne klesajúcim spôsobom od aktuálnej hodnoty záznamu, potom očakávaný počet zlepšujúcich bodov na dosiahnutie riešenia s danou presnosťou rastie nanajvýš lineárne v dimenzii d problému pre širokú triedu

globálnych optimalizačných problémov. Parameter T sa môže použiť na obmedzenie celkového počtu iterácií (vrátane nezáznamových hodnôt generovaných vo fáze akceptácie a zamietnutia). V ideálnom prípade by sa mala zvoliť taká hodnota parametra T , aby sa počas nasledujúcej iterácie pravdepodobnosť získania zlepšujúceho sa bodu rovnala aspoň nejakej pevnej hodnote.

Hesitantné adaptívne hľadanie

Jedným zo spôsobov implementácie čistého adaptívneho vyhľadávania by bolo použitie prístupu prijatia a zamietnutia na generovanie bodov v úrovňových množinách S (za predpokladu, že samotný problém generovania rovnomerne rozloženého bodu v S je relatívne jednoduchý). Z hľadiska celkového počtu iterácií by bol tento prístup ekvivalentný čistému adaptívnemu vyhľadávaniu s nasledujúcou úpravou. Pri každej iterácii sa buď vygeneruje nový iterát čistého adaptívneho vyhľadávania (s určitou pravdepodobnosťou b), alebo sa zopakuje bod predchádzajúcej iterácie (s pravdepodobnosťou b), kde b závisí od aktuálnej hodnoty záznamu. Presnejšie, b je relatívna miera množiny úrovni zodpovedajúca aktuálnej hodnote záznamu. Hesitantné adaptívne vyhľadanie, ktoré bolo uvedené s cieľom študovať lokalizačné vyhľadávacie algoritmy, zovšeobecňuje tento spôsob nazerania na čisté adaptívne hľadanie zmiernením špecifickej voľby b uvedenej vyššie. Explicitné vyjadrenie očakávaného počtu iterácií potrebných na získanie bodu s danou hodnotou účelovej funkcie (alebo lepšou), a dokonca aj úplné rozdelenie tejto náhodnej premennej je možné matematicky odvodiť.

3.1.2 Simulované žihanie

Simulované žihanie (*ang. Simulated Annealing*) je metóda náhodného hľadania, ktorá zabraňuje uviaznutiu v lokálnych maximách tým, že okrem prechodov zodpovedajúcich nárastu hodnoty funkcie akceptuje aj prechody zodpovedajúce poklesu hodnoty funkcie. Tento postup sa vykonáva v obmedzenej miere pomocou pravdepodobnostného kritéria akceptácie. V priebehu maximalizačného procesu pravdepodobnosť akceptovania zhoršení pomaly klesá k nule. Tieto "zhoršenia" umožňujú vzdialiť sa od lokálneho optima a preskúmať uskutočniteľnú oblasť S v celom jej rozsahu. Algoritmus vznikol na základe analógie s fyzikálnym procesom žihania pri hľadaní nízkoenergetických stavov pevnej látky v tepelnom kúpeli. Na základe tejto analógie bol vyvinutý algoritmus na riešenie diskretizácií spojitých globálnych optimalizačných problémov. Mnohé doterajšie aplikácie sa týkali diskretných (kombinatorických) optimalizačných problémov.

Všetky algoritmy z množiny algoritmov simulovaného žihania pre globálne optimalizácie možno považovať za aproximatívne verzie adaptívneho hľadania. Hlavný problém s priamou implementáciou simulovaného žihania spočíva v tom, že vo všeobecnosti bude veľmi ťažké generovať body presne z Boltzmannovho rozdelenia na množine S . Aproximatívny charakter simulovaného žihania spočíva v tom, že algoritmy simulovaného žihania namiesto toho používajú prístup vzorkovania Markovovým reťazcom. To znamená, že na množine S je definovaný Markovov reťazec, ktorý má vlastnosť, že limitným rozdelením tohto Markovovho reťazca je požadované Boltzmannovo rozdelenie. Jeden zo spôsobov, ako to dosiahnuť, je nasledujúci. Najprv sa vytvorí Markovov reťazec na množine S , ktorý má ako limitné rozdelenie rovnomerné rozdelenie. Príkladom takýchto Markovových reťazcov je generátor „hit and run“ a náhodná guľôčková prechádzka. Tento Markovov reťazec sa potom môže filtrovať takto, aby sa limitné rozdelenie zmenilo na Boltzmannovo rozdelenie. Ak je aktuálny iteračný bod x a aktuálna hodnota teplotného parametra je T , potom kandidátsky bod (povedzme z), ktorý je generovaný Markovovým reťazcom, je prijatý s pravdepodobnosťou

$$\min\{1; e^{(f(z) - f(x)/T)}\}$$

(kritérium Metropolis algoritmu). V opačnom prípade kandidátsky bod vyradíme a zostaneme v aktuálnom bode iterácie. V odbornej literatúre bolo preukázané, že hoci pri po sebe nasledujúcich iteráciách môže dôjsť k zhoršeniu hodnoty účelovej funkcie, tieto vplyvy sú za miernych podmienok prechodné, ak Markovov reťazec dosiahne globálne, t. j. ak každý realizovateľný bod možno dosiahnuť z ktoréhokoľvek iného realizovateľného bodu v jednej iterácii. Konkrétne, ak postupnosť teplotných parametrov T klesne s pravdepodobnosťou na nulu, simulované žihanie sa nakoniec absorbuje v ľubovoľne malých susedstvách globálneho maxima.

Harmonogram chladenia riadiaci spôsob znižovania parametra T by sa mal zvoliť v súlade s adaptívnym hľadaním. V praxi to znamená, že teplotný parameter by mal byť úmerný chybe (odhadu) hodnoty zodpovedajúcej aktuálnej rekordnej hodnote. Ak sa hodnota teploty udržiava na konštantnej hodnote 0 (čo znamená, že sa nikdy neakceptuje žiadne zhoršenie), dostaneme zlepšujúci sa algoritmus „hit and

run“. V odbornej literatúre sú odvodené výsledky konvergencie pre algoritmy simulovaného žihania s neglobálne dosahujúcimi Markovovými reťazcami.

Algoritmus simulovaného žihania funguje nasledovne:

- Inicializácia - začiatočné riešenie je vygenerované náhodne alebo podľa určitých pravidiel.
- Iterácie - algoritmus prechádza sériu iterácií, pričom každá iterácia predstavuje potenciálne lepšie riešenie. V každej iterácii sa vykonávajú nasledujúce kroky:
 - a) Generovanie susedného riešenia - susedné riešenie je vytvorené z aktuálneho riešenia prostredníctvom určitých pravidiel, ktoré zabezpečujú, aby bolo blízko k aktuálnemu riešeniu.
 - b) Vyhodnotenie susedného riešenia - susedné riešenie je vyhodnotené pomocou objektívnej funkcie, ktorá kvantifikuje kvalitu riešenia. Táto funkcia môže napríklad merať chybu, náklady alebo iný aspekt problému.
 - c) Akceptovanie alebo zamietnutie susedného riešenia - susedné riešenie je buď akceptované ako nové aktuálne riešenie, ak je lepšie ako predchádzajúce riešenie, alebo je akceptované s určitou pravdepodobnosťou, aj keď je horšie. Pravdepodobnosť akceptovania horšieho riešenia je nastavená tak, aby sa vyhľadával priestor možných riešení a minimalizovala sa pravdepodobnosť uväznenia v lokálnom optime.
 - d) Aktualizácia teploty - teplota je parameter algoritmu simulovaného žihania. V každej iterácii sa teplota postupne zníži podľa určitého režimu ochladzovania. Nižšia teplota znižuje pravdepodobnosť akceptovania horšieho riešenia, čo vedie k väčšej exploarácii priestoru riešení v začiatku a konvergencii k lepším riešeniam neskôr.
 - e) Koniec - algoritmus končí, keď je dosiahnutý určený počet iterácií, dosiahnutá požadovaná úroveň kvality riešenia alebo iný kritérium ukončenia.

Simulované žihanie je často používané na riešenie kombinatorických optimalizačných problémov, ako je napríklad problém obchodného cestujúceho (Traveling Salesman Problem) alebo problém plánovania výroby. Vďaka svojej schopnosti prehľadávať priestor riešení a unikať z lokálnych optimí je schopné dosiahnuť lepšie výsledky ako iné heuristické metódy.

Takže simulované žihanie je algoritmus optimalizácie, ktorý využíva princíp žihania materiálu na hľadanie globálneho optimálneho riešenia.

3.1.3 Metóda gradientného zostupu

Metóda gradientného zostupu (*ang. Gradient Descent*) je optimalizačný algoritmus, ktorý sa používa na minimalizáciu funkcií. Je často využívaný pri strojovom učení a v oblasti optimalizácie parametrov modelov.

Algoritmus gradientného zostupu využíva gradient (smer najväčšieho nárastu) funkcie na postupné aktualizovanie hodnôt parametrov tak, aby sa minimalizovala hodnota funkcie.

Postup metódy gradientného zostupu je nasledovný:

1. Inicializácia - začneme inicializáciou hodnôt parametrov, ktoré chceme optimalizovať. Tieto môžu byť nastavené náhodne alebo na určité preddefinované hodnoty.
2. Výpočet gradientu - v každom kroku algoritmu je potrebné vypočítať gradient funkcie vzhľadom na parametre. Gradient je vektor, ktorý obsahuje parciálne derivácie funkcie podľa každého parametra. Pomocou gradientu získame informáciu o smeroch, ktorými sa musia parametre posunúť, aby sa minimalizovala hodnota funkcie.
3. Aktualizácia parametrov - na základe gradientu a rýchlosti učenia (*ang. learning rate*) aktualizujeme hodnoty parametrov. Rýchlosť učenia určuje, akou veľkou hodnotou sa parametre posunú v smere gradientu. V každom kroku sa hodnoty parametrov aktualizujú nasledujúcim spôsobom: nová hodnota parametra = stará hodnota parametra - (rýchlosť učenia) * (gradient).
4. Opakovanie krokov 2 a 3 - tieto kroky sa opakujú, kým nie je dosiahnuté konvergenčné kritérium. To môže byť napríklad určený počet iterácií, dosiahnutie požadovanej presnosti alebo stabilizácia hodnôt parametrov.

Dôležitým faktorom v metóde gradientného zostupu je rýchlosť učenia. Príliš veľká hodnota rýchlosti učenia môže spôsobiť prekročenie minimálneho bodu a neschopnosť algoritmu dosiahnuť optimálne riešenie. Naopak, príliš malá hodnota rýchlosti učenia môže spomaliť konvergenciu algoritmu. Takže metóda gradientného zostupu je iteratívny proces, ktorý aktualizuje hodnoty parametrov v smere opačnom k gradientu funkcie s cieľom minimalizovať hodnotu tejto funkcie.

3.1.4 Particle Swarm Optimization (PSO)

Particle Swarm Optimization (PSO) je optimalizačný algoritmus inšpirovaný správaním roja častíc. Tento algoritmus je často používaný na hľadanie optimálnych riešení v priestore riešení s veľkým počtom dimenzií a pri problémoch, kde je zložité analyticky vypočítať gradienty.

Algoritmus PSO je možné popísať nasledovne:

- Inicializácia - na začiatku je potrebné inicializovať roj (skupinu) častíc. Každá častica je reprezentovaná v priestore riešení ako bod s určitými parametrami. Parametre sa inicializujú náhodne alebo podľa určitých pravidiel.
- Vyčíslenie hodnoty fitness - pre každú časticu v roji je vypočítaná hodnota fitness, ktorá kvantifikuje kvalitu riešenia. Táto hodnota je získaná vyhodnotením objektívnej funkcie, ktorá závisí od parametrov častice.
- Nájdenie najlepšej hodnoty fitness - pre každú časticu je zaznamenaná jej najlepšia hodnota fitness, ktorú dosiahla doteraz, a prislúchajúce parametre. Tento najlepší osobný výkon (pbest) predstavuje doterajšie najlepšie riešenie danej častice.
- Nájdenie globálneho najlepšieho výkonu - v rámci roja sa vyhľadáva globálne najlepšie riešenie, ktoré predstavuje najlepšie riešenie zo všetkých častíc. Táto hodnota sa nazýva najlepší globálny výkon (gbest).
- Aktualizácia polohy a rýchlosti - pre každú časticu v roji sa aktualizuje jej poloha a rýchlosť na základe jej predošlej polohy, rýchlosti, pbest a gbest. Aktualizácia sa vykonáva podľa nasledujúcich rovníc:
$$\text{Nová rýchlosť} = (\text{inercia}) * (\text{súčasná rýchlosť}) + (\text{kognitívny faktor}) * (\text{náhodné číslo}) * (\text{pbest} - \text{súčasná poloha}) + (\text{sociálny faktor}) * (\text{náhodné číslo}) * (\text{gbest} - \text{súčasná poloha}),$$
$$\text{Nová poloha} = \text{súčasná poloha} + \text{nová rýchlosť}.$$
- V týchto rovniach sa vyskytujú niektoré faktory - inercia určuje mieru, akou sa rýchlosť mení z predošlého kroku, kognitívny faktor určuje vplyv osobného najlepšieho riešenia, sociálny faktor určuje vplyv najlepšieho globálneho riešenia a náhodné číslo je náhodná hodnota v rozsahu (0, 1).
- Opakovanie krokov 2 až 5 - tieto kroky sa opakujú v definovanom počte iterácií alebo kým nie je splnené konvergenčné kritérium. Typicky sa algoritmus PSO opakuje, kým sa nedosiahne požadovaná presnosť alebo nie je dosiahnutý maximálny počet iterácií.

Celkovým cieľom algoritmu PSO je optimalizovať hodnoty parametrov tým, že častice vo vyhľadávanom priestore spolupracujú a vymieňajú si informácie o najlepších riešeniach. Týmto spôsobom sa postupne zlepšuje celková kvalita riešenia a hľadá sa globálne najlepšie riešenie problému.

4 MICROGRIDBOX® PORTÁL

MicroGridBox® portál je nástroj na modelovanie a simulácie elektrických schém. Modelovanie je možné prostredníctvom vkladania grafických prvkov priamo z predpripravenej knižnice prvkov do schémy alebo mapy. Nástroj podporuje základné funkcie pre tvorbu a úpravu grafických prvkov, ako aj vytváranie prepojení medzi nimi. Grafické modely mikrogridov je možné vytvárať vo forme schematickeho (topologického) zobrazenia alebo v prostredí mapových podkladov (topografického zobrazenia). Tento nástroj tiež umožňuje ukladanie a správu projektov týchto modelov pre ich budúce použitie a úpravu. V kontexte grafickej reprezentácie modelu mikrogridu je nástroj schopný zobrazit' aj ďalšie informácie o jednotlivých zariadeniach.

Ciele nástroja v oblasti modelovania a vizualizácie sú nasledovné:

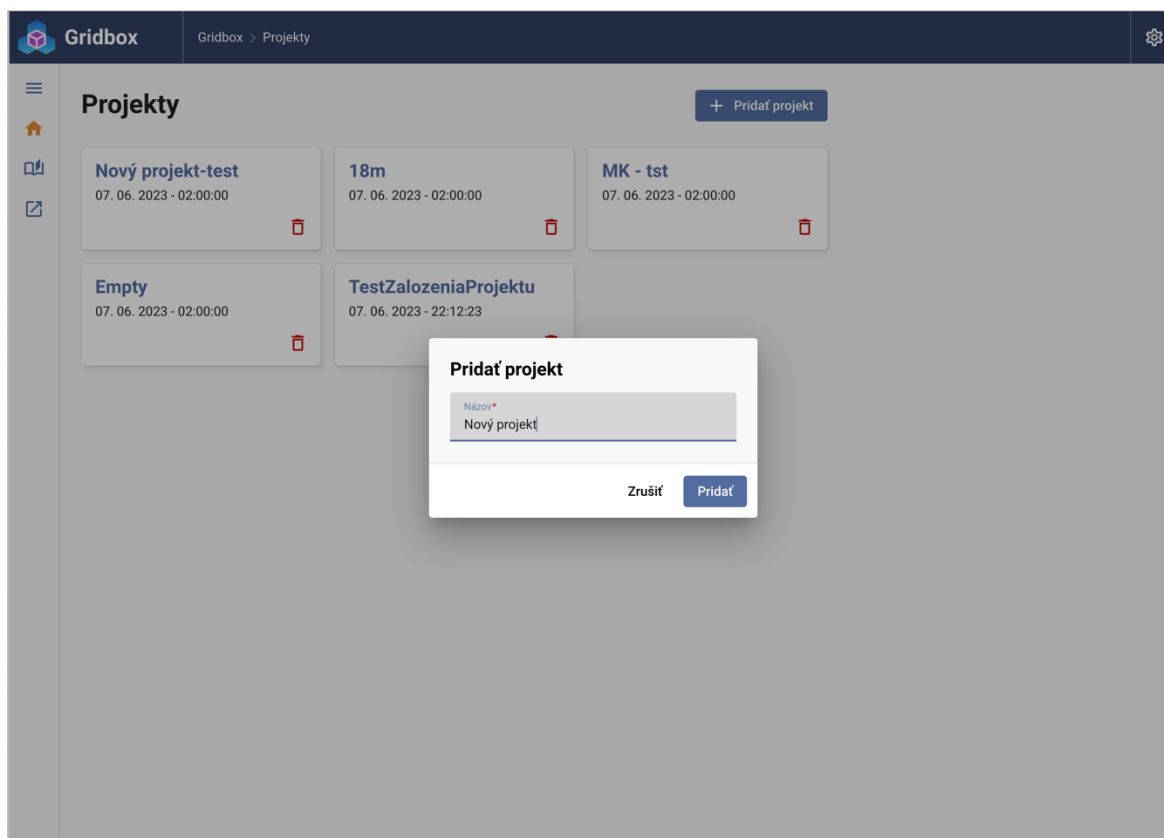
- správa projektov,
- knižnica grafických prvkov,
- modelovanie,
- vizualizácia.



Obrázok 6 Funkčné oblasti časti pre modelovanie a vizualizáciu.

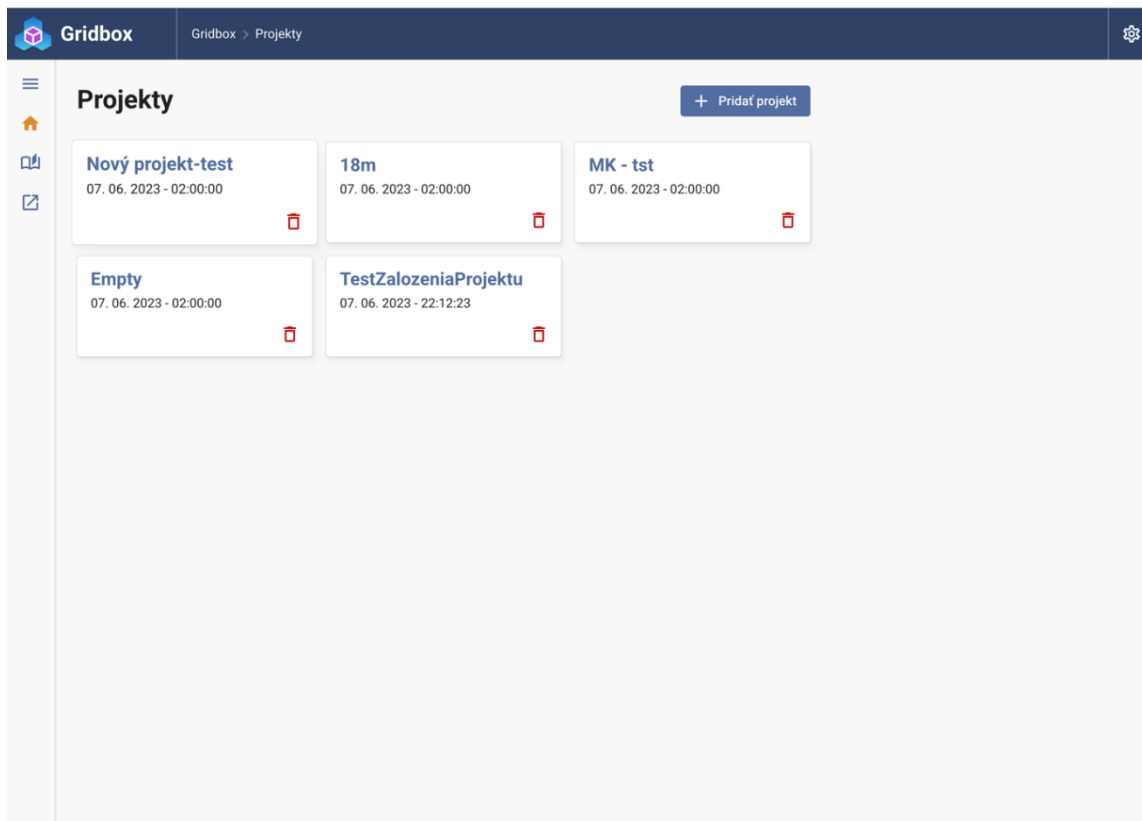
4.1 Správa projektov

Nástroj umožňuje ukladanie jednotlivých sieťových modelov vo forme projektov pre budúce použitie. Projekt obsahuje základné atribúty, ako sú názov, dátum vytvorenia a dátum aktualizácie. V rámci projektu sa nachádzajú zoznamy prvkov, prepojení a skupín grafu. Prvky slúžia na reprezentáciu objektov generujúcich alebo spotrebúvajúcich elektrinu a uchovávajú informácie o svojej pozícii v modeli. Prepojenia spájajú dva prvky a zahŕňajú atribúty, ako je začiatkový a koncový prvok, ktoré sú spojené. Okrem toho môže mať každý prvok aj ďalšie atribúty. Nový projekt je možné vytvoriť cez Gridbox > Projekty, kde je dostupné tlačidlo **+ Pridať projekt**. Po kliknutí na toto tlačidlo sa zobrazí dialóg pre zadanie názvu projektu pre modelovanie.



Obrázok 7 Vytvorenie nového projektu

Cez navigáciu Gridbox > Projekty je možné aj otvoriť/upraviť alebo vymazať už existujúce projekty.



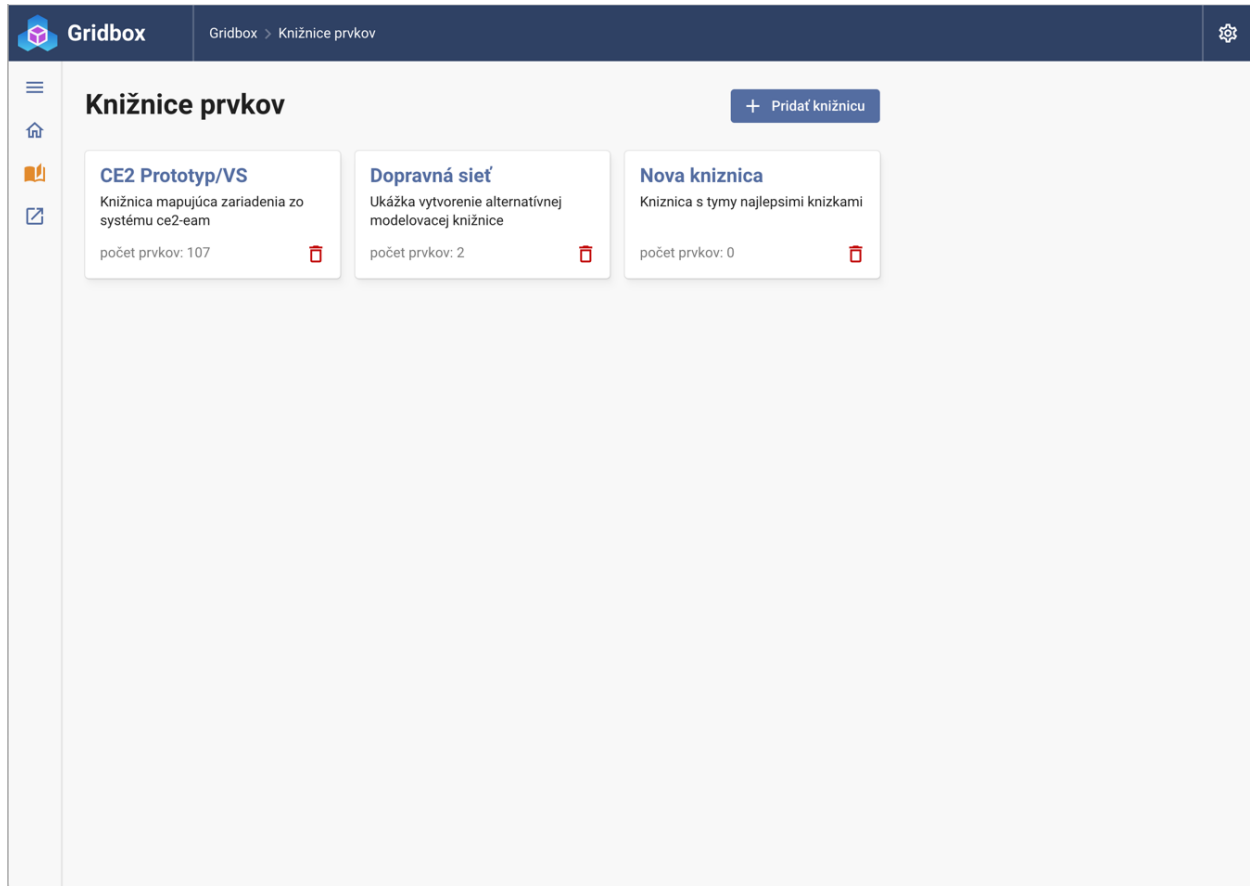
Obrázok 8 Manažment projektov

4.2 Knižnica grafických elementov

V časti nástroja Gridbox > Knižnica je dostupná knižnica prvkov - grafických elementov, teda sadu prvkov a prepojení, z ktorých je možné vytvárať model siete. Jeden projekt môže použiť aj viaceré knižnice prvkov. Každému prvku je možné priradiť grafickú reprezentáciu, ktorá určí vzhľad daného prvku v rámci modelu. Každý prvok má pridelené atribúty, ktoré ho bližšie špecifikujú.

Knižnica prvkov ponúka nasledujúce funkcie:

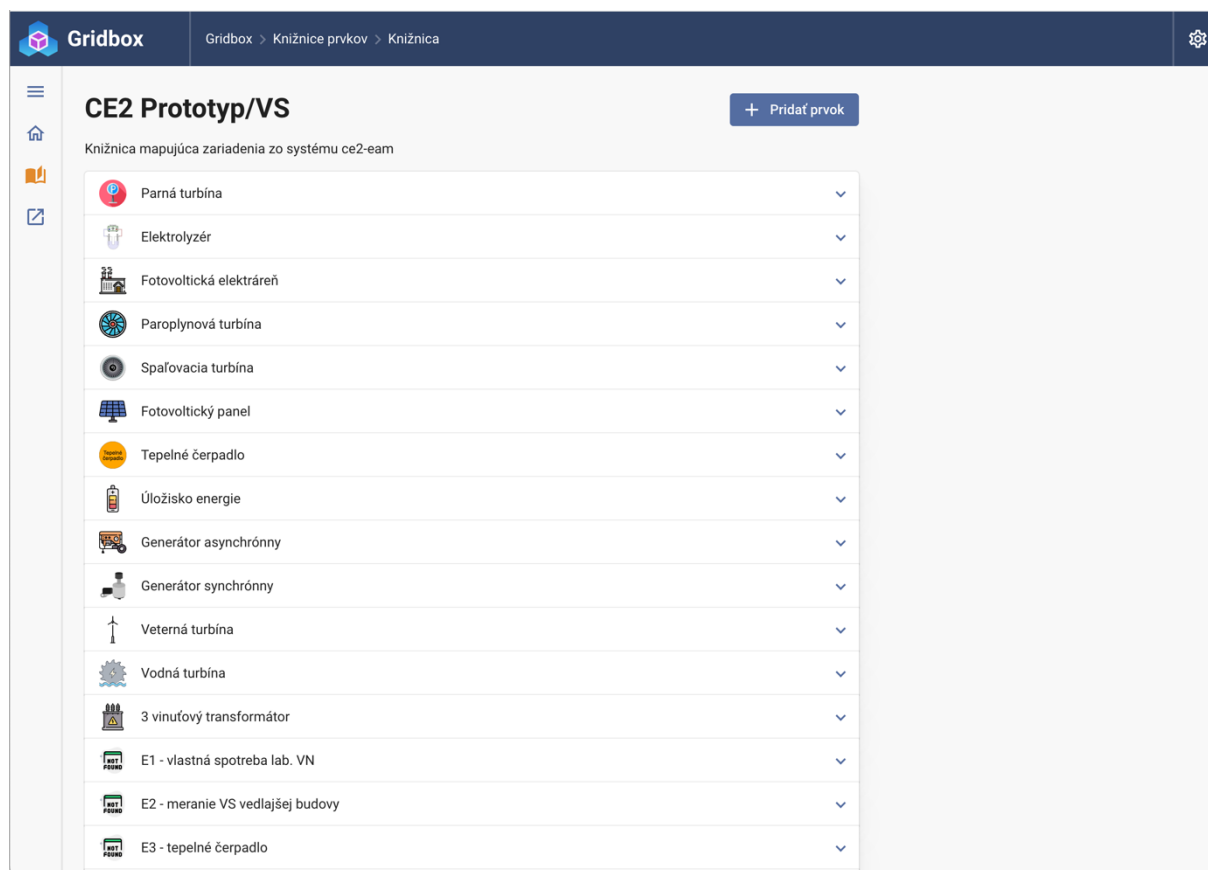
- vytvorenie knižnice prvkov,
- modifikáciu knižnice prvkov,
- mazanie knižnice prvkov.



Obrázok 9 Knižnica prvkov

V rámci manažmentu knižnice prvkov je možné:

- pridávať prvky,
- odobrať prvky,
- prideliť prvku ikonu,
- pridelenie atribútov zo systému CE2 Prototyp/DB.



Obrázok 10 Detail vybranej knižnice prvkov.

Na obrázku vyššie je vidieť obsah vybranej knižnice prvkov s názvom CE2 Prototyp/VS. V pravo hore je tlačidlo s názvom **+ Pridať prvok**, ktorý umožňuje pridávanie prvkov do knižnice.

Na obrázku nižšie je vidieť detail pridaného prvku, ktorému je možné pridať ikonku a atribúty, ktoré sa automaticky stiahnu a pridelia z externého systému CE2 Prototyp/DB.

CE2 Prototyp/VS + Pridať prvok

Knižnica mapujúca zariadenia zo systému ce2-eam

- Parná turbína
- Elektrolyzér
- Fotovoltaická elektrárň
- Paroplynová turbína
- Spaľovacia turbína
- Fotovoltaický panel
- Tepelné čerpadlo

Úložisko energie

Nastavenia prvku

Názov elementu*
Úložisko energie

Identifikátor elementu*
ÚLOŽISKO_ENERGIE

Druh elementu*
nodes

Uložiť

Parametre prvku (12)

Načítať atribúty z

Názov	Typ	Jednotka
Energia uložená v akumulátore na začiatku časového intervalu	number	J/kg
Kapacita akumulátora	text	
Menovitá kapacita akumulátora	text	
Nabijací prúd akumulátora	number	A

Obrázok 11 Úprava existujúceho prvku z knižnice

4.3 Modelovanie

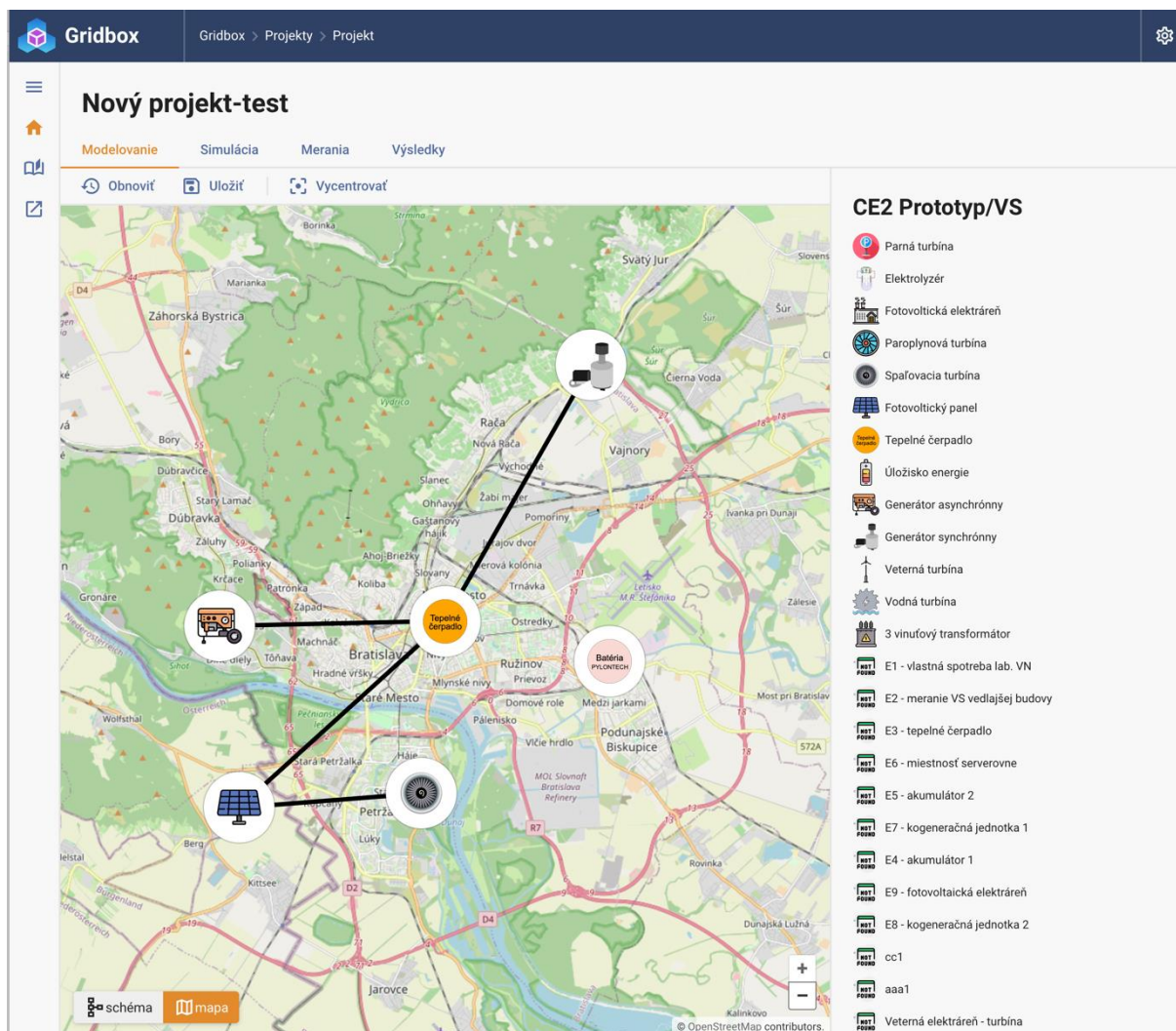
Prototyp poskytuje funkcie pre vytváranie a modifikáciu topológie siete. Sú podporené dva režimy modelovania. Vo forme schémy, kde sa jednotlivé prvky ukladajú a prepájajú v prostredí modelovacej mriežky (gridu), alebo podkladom pre modelovanie je mapa a prvky sa ukladajú podľa ich skutočnej geografickej pozície. Oba režimy modelovania je možné kedykoľvek prepínať. Prvky a prepojenia, ktoré sa používajú pri modelovaní sa vyberajú z knižnice grafických elementov.

Nástroj umožňuje dva režimy modelovania:

1. Schematické – podklad je mriežka,
2. Geografické – podklad je mapa.

Nástroj umožňuje jednoduché prepnutie formy zobrazenia priamo v modelovacom diagrame, pričom nie je potrebné aktuálnu prácu opúšťať ani uložiť.

Nástroj umožňuje vkladať a modelovať prvky aj z viacerých knižníc naraz, pričom musia byť v danom projekte všetky knižnice pripojené. Prvok je možné vkladať do siete dvomi spôsobmi, a to bežne používaným drag&drop prístupom, prípadne len výberom elementu a následným jednoduchým kliknutím myšou na požadované miesto.



Obrázok 12 Zobrazenie mapovej verzie podkladu pod zariadeniami

4.3.1 Prepojenie prvkov

Prepojenia sú dôležité pre vytvorenie samotného grafu a pre definovanie jednotlivých hrán medzi prvkami siete. Prepojiť je možné ľubovoľný prvok s iným, pričom nie je povolené cyklické prepojenie, ktoré by v mnohých prípadoch nemalo zmysel. Týmto prístupom je zabezpečené aj budúce možné hierarchické usporiadanie prvkov v sieti.

4.3.2 Vymazanie prvkov

Vymazanie prvku je možné vykonať pre konkrétne vybraný element, prípadne aj pre množinu viacerých vybraných prvkov a stlačením tlačidla „Del“ na klávesnici. Vymazanie rieši aj odstránenie závislých hrán, ktoré korešpondovali s daným elementom.

4.3.3 Modifikácia hodnôt atribútov

Pri rozkliknutí prvku priamo z modelovacej časti sa vysunie panel so všetkými dostupnými parametrami. V danom formulári je možné hodnoty meniť priamo používateľom systému. Na tomto mieste nemusia byť nutne zobrazené všetky atribúty objektu, ale len hlavné parametre, ktoré boli určené pri administrácii entity používateľom s vyššími oprávneniami. Táto myšlienka však nebola dotiahnutá, a preto sa v aktuálnej verzii zobrazujú všetky atribúty.

The screenshot displays the Gridbox web application interface. At the top, there is a navigation bar with the Gridbox logo and the text 'Gridbox > Projekty > Projekt'. Below this, the main header shows 'Nový projekt-test' and tabs for 'Modelovanie', 'Simulácia', 'Merania', and 'Výsledky'. The 'Modelovanie' tab is active, showing a schematic diagram of a power system on a grid. The diagram includes a solar panel icon, a turbine icon, a battery icon, and a generator icon, all connected by lines. A red 'x' icon is visible near the turbine icon. To the right of the diagram is a 'Parametre' panel with the following data:

Parametre	
Názov	Spaľovacia turbína
Typ	SPALOVACIA_TURBINA
Latitude	48.11580583849226
Longitude	17.12558097670025

Below the parameters is an 'Atribúty' panel with the following data:

Atribúty	
Načítať atribúty z	▼
Id	73
Hrubý spád	17,21
Kavitačný koeficient	0,0520
Maximálna účinnosť turbíny	85,10
Merná energia turbíny	126,77
Pomer priebežných a menovitých otáčok turbíny	1,7140
Prevádzkové otáčky	244
Priebežné otáčky	418
Priemer obežného kolesa turbíny	0,60

At the bottom of the diagram area, there are buttons for 'schéma' and 'mapa', and a zoom control with '+' and '-' symbols.

Obrázok 13 Panel s parametrami zariadenia

4.3.4 Ďalšie operácie

- zväčšenie zobrazenie modelu,
- zmenšenie zobrazenie modelu,
- presun zobrazenie modelu ľubovoľným smerom,
- prepínanie medzi schematickým a geografickým zobrazením,
- vycentrovanie modelu.

4.4 Vizualizácia

Nástroj zabezpečuje zobrazenie digitálneho modelu siete. Pri implementácii sme využili nasledovné technológie a frameworky:

- Angular,
- Sféra UI KID,
- X6 JavaScript Diagramming Library | AntV,
- Rest API,
- Microsoft .NET core,
- PostgreSQL.

Používateľské rozhranie nástroja bolo vytvorené s využitím UI frameworku Angular a UI komponent Sféra UI KID. Na implementáciu nástroja na kreslenie a zobrazenie schém bola vybraná knižnica X6 JavaScript Diagramming Library | AntV.

Angular je open-source webový aplikačný framework vyvinutý spoločnosťou Google. Je určený pre vývoj moderných jednostránkových aplikácií (SPA) a je postavený na jazyku JavaScript/TypeScript. Angular ponúka škálovateľný a komponentovo orientovaný prístup k vývoju webových aplikácií. Angular používa komponentový model, kde aplikácia je zostavená z viacerých nezávislých komponentov, ktoré spolupracujú medzi sebou. Angular poskytuje deklaratívne šablóny, ktoré umožňujú vývojárom definovať používateľské rozhrania pomocou HTML a rozšíreného značkovacieho jazyka (Angular Template Syntax). Angular umožňuje jednoduché a efektívne dvojcestné viazanie dát, čo znamená, že zmeny v modeli automaticky ovplyvnia zobrazenie a zmeny v zobrazení sa odrážajú v modeli. Angular podporuje modulárny prístup k vývoju, čo umožňuje rozdelenie aplikácie do logických a opätovne použiteľných častí. Angular poskytuje mechanizmy pre správu stavu aplikácie, čo zjednodušuje sledovanie a aktualizáciu stavu rôznych komponentov. Okrem toho, Angular je dobre integrovaný so širokou škálou nástrojov a knižníc, ako napríklad Angular CLI pre rýchle vytváranie, správu a nasadenie aplikácií, a RxJS pre prácu s asynchrónnymi dátami. Angular sa stal populárnym frameworkom medzi vývojármi a je často používaný pre vývoj robustných a výkonných webových aplikácií.

Sféra UI KID je sada UI komponent vytvorených spoločnosťou sféra pre efektívnu tvorbu webových aplikácií. Tieto komponenty sú implementované pomocou Angular a Material Design. UI KID obsahuje okrem UI komponent aj príklad efektívneho rozloženia stránok.

X6 JavaScript Diagramming Library je knižnica pre tvorbu interaktívnych a vizuálne bohatých diagramov v jazyku JavaScript. Táto knižnica je vyvíjaná spoločnosťou AntV a poskytuje vývojárom silné nástroje na vytváranie, úpravu a zobrazovanie rôznych typov diagramov, ako sú organizačné štruktúry, grafy, schémy, tokové diagramy a ďalšie. X6 JavaScript Diagramming Library ponúka množstvo funkcií a možností, ktoré umožňujú prispôbenie vzhľadu a správania diagramov. Obsahuje pokročilé nástroje na manipuláciu s prvky diagramu, vrátane možnosti ťahať, zmešovať, zväčšovať a prepojovať prvky. Taktiež umožňuje pridávať interakciu a animácie do diagramov, čo zlepšuje užívateľský zážitok. X6 je navrhnuté tak, aby bolo ľahko rozšíriteľné a prispôbitel'né. Poskytuje možnosti pre rozšírenie funkcií, pridávanie vlastných typov prvkov, definovanie šablón a mnoho ďalších možností prispôbenia podľa potrieb vývojára. Táto knižnica je populárna vo vývojových komunitách, ktoré sa zaoberajú tvorbou diagramov, a poskytuje robustné a flexibilné riešenie pre prácu s vizuálnymi reprezentáciami dát.

REST (Representational State Transfer) je architektúra pre návrh rozhraní pre webové služby, ktorá je založená na princípoch a štýloch HTTP protokolu. REST API (Application Programming Interface) je základom pre implementáciu REST architektúry. REST API poskytuje spôsob ako webové aplikácie a služby môžu komunikovať a vymieňať si dáta. Pomocou REST API je možné vykonávať operácie na vzdialených serveroch cez jednoduché a štandardizované HTTP metódy, ako sú GET, POST, PUT a DELETE. Každá z týchto metód má definovaný zmysel a účel:

- GET - slúži na získanie dát zo servera. Požiadavka GET je bezpečná a nemá žiadny vplyv na stav servera.
- POST - používa sa na vytvorenie nových záznamov na serveri alebo na odoslanie dát na spracovanie. Môže mať vplyv na stav servera.
- PUT - aktualizuje existujúce dáta na serveri. Má za úlohu nahráť celé zdroje.
- DELETE - slúži na odstránenie existujúcich dát zo servera.

REST API je nezávislé od platformy a programovacieho jazyka a umožňuje klientom (ako napríklad webové stránky alebo mobilné aplikácie) komunikovať so serverom pomocou jednoduchých HTTP požiadaviek a odpovedí. Dáta sa často prenášajú vo formáte JSON (JavaScript Object Notation), ktorý je ľahko čitateľný pre ľudí aj stroje. Využívanie REST API umožňuje vytvárať moderné, škálovateľné a distribuované aplikácie, ktoré môžu využívať a integrovať dáta, a funkcionality z rôznych zdrojov a služieb na internete.

Microsoft .NET Core je open-source platforma a vývojový rámec vyvinutý spoločnosťou Microsoft. Je to odľahčená a krížovo-platformová verzia pôvodného rámca .NET Framework. .NET Core je navrhnutý tak, aby bol rýchly, flexibilný a optimalizovaný pre moderné webové a cloudové aplikácie.

Hlavné vlastnosti a výhody Microsoft .NET Core sú:

- **Křížová platformovosť** - .NET Core je navrhnutý tak, aby bežal na viacerých operačných systémoch, vrátane Windows, macOS a rôznych distribúcií Linuxu. To umožňuje vývojárom písať aplikácie pre rôzne platformy s použitím rovnakého kódu.
- **Odlahčenosť** - .NET Core je zameraný na vytváranie efektívnych a odlahčených aplikácií. Obsahuje len minimálny súbor funkcií, čo znamená menšie nároky na pamäť a zdroje, a umožňuje rýchlejšie spustenie a vykonávanie aplikácií.
- **Open-source** - .NET Core je open-source projekt, čo znamená, že jeho zdrojový kód je verejne dostupný a môže byť prispievajú komunitou vývojárov. To umožňuje transparentnosť, spoluprácu a prispôbenie rámca podľa potrieb.
- **Podpora pre moderné scenáre** - .NET Core je optimalizovaný pre webové aplikácie, mikroslužby, cloudové aplikácie a kontajnery. Poskytuje výkonné nástroje a knižnice pre vývoj a nasadenie moderných aplikácií, ako napríklad ASP.NET Core pre vytváranie webových aplikácií.
- **Kompatibilita s .NET Framework** - hoci .NET Core je samostatný rámec, existuje aj kompatibilita s pôvodným .NET Framework. To umožňuje vývojárom zdieľať kód medzi .NET Core a .NET Framework a postupne migrovať svoje aplikácie na novšiu platformu.

Microsoft .NET Core je dôležitým nástrojom pre vývojárov, ktorí chcú vytvárať moderné, výkonné a křížovo-platformové aplikácie. Spolu s bohatou sadou knižníc a nástrojov ponúka .NET Core prostredie na vytváranie inovatívnych softvérových riešení.

PostgreSQL je open-source relačný databázový systém, ktorý je vyvinutý a udržiavaný komunitou. Jeho názov je skratkou pre "Postgres Structured Query Language" (PostgreSQL). PostgreSQL je známy svojou spoľahlivosťou, flexibilitou a schopnosťou spravovať aj veľké objemy dát.

Hlavné vlastnosti a funkcie PostgreSQL zahŕňajú:

- **Relačný model** - PostgreSQL je založený na relačnom modeli databáz, kde dáta sú organizované do tabuliek s riadkami a stĺpcami, a vzťahy medzi tabuľkami sú definované pomocou kľúčových a cudzích kľúčov.
- **Podpora SQL** - PostgreSQL implementuje štandardný jazyk SQL (Structured Query Language), čo umožňuje vývojárom efektívne manipulovať s dátami v databáze. Podporuje množstvo pokročilých funkcií SQL, ako sú komplexné dotazy, agregácie, indexy, triggery a procedúry.
- **Rozšíriteľnosť** - PostgreSQL umožňuje rozšírenie svojich funkcií pomocou používateľsky definovaných typov dát, funkcií, operátorov a procedúr. Taktiež podporuje externé rozšírenia, ktoré umožňujú pridať do systému nové funkcionality.
- **Podpora transakcií** - PostgreSQL poskytuje transakčnú podporu, čo znamená, že množina operácií nad databázou môže byť vykonaná ako jedna atomická jednotka. Akákoľvek chyba alebo prerušenie vykonávania transakcie vedie k jej odvolaniu (rollback) a obnoveniu pôvodného stavu.
- **Bezpečnosť** - PostgreSQL poskytuje rôzne mechanizmy na zabezpečenie dát, vrátane prístupových kontrol, autentifikácie, šifrovania a auditovania.
- **Replikácia a vysoká dostupnosť** - PostgreSQL umožňuje replikáciu dát, čo zabezpečuje ich kópiu na viacerých serveroch. Týmto spôsobom je možné dosiahnuť vyššiu dostupnosť a odolnosť voči výpadkom.

PostgreSQL je široko používaný v rôznych typoch aplikácií, od malých projektov až po veľké podnikové systémy. Jeho kombinácia výkonnosti, rozšíriteľnosti a otvoreného zdrojového kódu ho robí obľúbenou voľbou pre mnoho vývojárov a organizácií.

5 TESTOVANIE PROTOTYPU VO VZŤAHU K REÁLNEJ MIKROSIETI

Testovanie prototypu je dôležitý proces v rámci vývoja MicroGridBoxu® ako softvérového riešenia, ktorý slúži na overenie správnosti, spoľahlivosti a funkčnosti softvéru. Jeho hlavným cieľom je identifikovať chyby, nedostatky a potenciálne problémy v softvéri ešte pred jeho uvoľnením do produkčného prostredia.

Testovanie MicroGridBoxu® bolo vykonané v rámci testovacieho cyklu, ktorý zahŕňal nasledujúce fázy:

1. Plánovanie testovania - definovanie testovacieho plánu, ktorý určuje testovacie stratégie, testovacie scenáre, rozsah testovania a zdroje potrebné na vykonanie testov.
2. Návrh testov - vytvorenie testovacích scenárov a prípadov, ktoré zahŕňajú rôzne aspekty softvéru a jeho funkcionality. Testovacie scenáre by mali pokrývať rôzne situácie a používateľské príbehy.
3. Implementácia testov - vykonávanie testovacích scenárov podľa stanovených plánov a špecifikácií. Testy môžu byť vykonávané manuálne alebo automatizovanými testovacími nástrojmi.
4. Hodnotenie a vyhodnotenie výsledkov - zhodnotenie výsledkov testov a porovnanie s očakávaniami a preddefinovanými kritériami. Identifikácia chýb, nedostatkov a potenciálnych problémov v softvéri.
5. Oprava a pretestovanie - ak sa počas testovania identifikujú chyby, je potrebné ich opraviť a následne vykonať opätovné testovanie, aby sa zabezpečilo, že problémy boli úspešne vyriešené.

Testovanie softvéru obecnne môže zahŕňať rôzne typy testov, vrátane unit testov, integračných testov, systémových testov, regresných testov a výkonnostných testov. Každý typ testovania má svoj vlastný zámer a zameranie.

Dôkladné testovanie softvéru je dôležitým krokom, ktorý pomáha zabezpečiť, že softvér je funkčný, bezpečný a spĺňa požiadavky používateľov. Testovanie tiež prispieva k zlepšeniu kvality softvéru a znižuje riziko výskytu chýb po jeho uvoľnení do produkčného prostredia

5.1 Plánovanie testovania

Plánovanie testovania MicroGridBoxu® bolo dôležitým krokom v rámci celého vývojového procesu, ktorý sa zameriaval na definovanie stratégií, cieľov a postupov pre jeho testovanie. Hlavným cieľom bolo zabezpečiť systematický a efektívny prístup k testovaniu, aby sa dosiahlo čo najväčšia pokrytie funkcionality a zároveň minimalizovali riziká a chyby.

Bolo vytypovaných niekoľko kľúčových aspektov plánovania testovania softvérového riešenia - MicroGridBoxu®:

1. Stanovenie cieľov - definovanie hlavných cieľov testovania. Sem patrí najmä overenie správnosti funkcionality, spoľahlivosti, výkonu, bezpečnosti a použiteľnosti softvéru. Je dôležité identifikovať, čo sa chce dosiahnuť testovaním a aké sú kritériá pre úspech.
2. Identifikácia testovacích prípadov - identifikovanie konkrétnych prípadov testovania, ktoré pokrývajú rôzne aspekty MicroGridBoxu® a jeho funkcionality. Tieto prípady boli založené na požiadavkách a špecifikáciách softvéru v spolupráci s technickými expertmi a zameriavali sa na kritické oblasti a potenciálne riziká.
3. Plánovanie testovacích scenárov - vytvorili sa testovacie scenáre, ktoré popisujú konkrétne postupy a kroky pre vykonávanie testov. Scenáre boli dostatočne podrobné a jasné, aby umožnili opakovateľnosť a reprodukovateľnosť testov.
4. Rozvrhnutie testovacieho cyklu - plánovanie harmonogramu testovania zohľadnilo dostupnosť ľudských zdrojov, časové kolízie s inými projektami spoločnosti a priority. Takto sa zabezpečilo integrované testovanie celého vývojového procesu a dostatočne predchádza plánovanému vydaniu softvéru.
5. Výber testovacích metód a nástrojov - rozhodnutie o vhodných testovacích metódach a nástrojoch, ktoré budú použité na vykonávanie testov. To zahŕňa rozhodnutie o manuálnom

testovaní, automatizovanom testovaní, simuláciách, použití testovacích rámcov a podobne. Výber správnych nástrojov a prístupov ovplyvnilo efektívnosť a efektívnosť testovania. Metódy boli zvolené v koordinácii s odbornými pracovníkmi.

6. Zdroje a prostredia - identifikácia potrebných zdrojov a prostredí pre testovanie, ako napríklad testovacie prostredie, testovacie dáta, testovacie nástroje a testovacie dokumenty prebehla v koordinácii s odbornými pracovníkmi a na základe nameraných dát. Testovacie prostredie bolo v maximálnej možnej miere podobné produkčnému prostrediu a k dispozícii boli potrebné dáta a podklady.
7. Riadenie chýb a sledovanie výsledkov - definovanie procesu pre správu chýb a sledovanie výsledkov testovania. To zahŕňa správu chýb, dokumentáciu chýb, ich sledovanie a ich riešenie v rámci vývojového cyklu.

Plánovanie testovania MicroGridBoxu® sa ukázal ako dynamický proces, ktorý sa prispôbil potrebám a požiadavkám výskumného projektu. Ako opodstatnené sa ukázala integrácia plánovania celkového vývojového procesu a aby bolo pravidelne aktualizované a prispôsobované na základe zmien v softvéri a požiadavkách projektu.

5.2 Návrh testov

Návrh testovania prototypu bol determinovaný ako proces definovania stratégie a plánu pre vykonávanie testovania MicroGridBoxu®. Jeho hlavným cieľom bolo zabezpečiť správnosť testov tak, aby boli dobre navrhnuté, efektívne a pokrývajúce všetky kritické aspekty a funkcionality softvéru. Správny návrh testov pomáha zvýšiť spoľahlivosť testovania a znižuje riziko neodhalenia chýb a nedostatkov.

Boli identifikované nasledujúce aspekty návrhu testov:

1. Analýza požiadaviek a špecifikácií - dôležitým prvým krokom pri návrhu testov bola dôkladná analýza predpokladaných požiadaviek a špecifikácií MicroGridBoxu®. Tým sa získali znalosti o predpokladaných funkcionality, očakávanom správaní a požadovanom výstupe softvérového riešenia. Na základe tejto analýzy sa identifikovali testovacie ciele a prípady, ktoré mali byť pokryté testovaním.
2. Identifikácia testovacích scenárov - testovacie scenáre popisovali postupy a kroky, ktoré mali byť vykonané počas testovania. Zároveň obsahovali vstupy, očakávané výstupy a predpoklady pre testy. Pri návrhu testov sa identifikovali relevantné scenáre, ktoré pokrývali rôzne aspekty softvéru a jeho funkcionality.
3. Výber testovacích metód a prístupov - na základe analýzy požiadaviek a špecifikácií bolo potrebné vybrať vhodné testovacie metódy a prístupy. To zahŕňalo manuálne testovanie, automatizované testovanie, regresné testovanie, záťažové testovanie, testovanie použiteľnosti a ďalšie. Výber správnych metód závisel od charakteru softvéru, dostupných zdrojov a cieľov testovania.
4. Definovanie testovacích dát - testovacie dáta časových radov nameraných hodnôt výroby a spotreby elektrickej energie boli základom pre vykonávanie testov. Pri návrhu testov sa identifikovali potrebné testovacie dáta, vrátane testovacích vstupov, očakávaných výstupov a rôznych kombinácií. Testovacie dáta by mali byť reprezentatívne a pokrývať rôzne scenáre a možné vstupné kombinácie.
5. Vytvorenie testovacích scenárov a prípadov - na základe identifikovaných scenárov a požadovaných vstupov sa vytvorili konkrétne testovacie scenáre a prípady. Zároveň boli navrhnuté tak, aby pokrývali rôzne možnosti a hranice testovanej funkcionality.
6. Plánovanie prioritizácie testov - pri návrhu testov sa určila priorita testovacích scenárov a prípadov. Niektoré scenáre mali vyššiu prioritu z dôvodu ich kritickosti alebo výskytu v reálnych scenároch použitia. Prioritizácia pomohla zameriavať zdroje a čas na kritické oblasti testovania a zabezpečila, že najdôležitejšie testy boli vykonané.
7. Identifikácia prostredí a zdrojov - pri návrhu testov bolo potrebné zabezpečiť dostupnosť nutných testovacích prostredí a zdrojov. To zahŕňalo testovacie prostredie, testovacie nástroje, testovacie dáta nameraných hodnôt a potrebné hardvérové a softvérové prostredia.

Návrh testov sa ukázal ako dôležitý krok v rámci testovacieho procesu, ktorý zabezpečuje efektívne a kvalitné testovanie softvéru. Dobrý návrh testov rýchlo odhalí chyby a nedostatky.

5.3 Implementácia testov

Implementácia testov ako proces zahrnuje vykonávanie navrhnutých testov a získavanie výsledkov testovania. Išlo o fázu, v ktorej sa testy vykonávajú v praxi a overuje sa správnosť a funkčnosť softvérového riešenia.

Činnosti v rámci implementácie testov:

1. Príprava testovacieho prostredia - pred vykonaním testov sa ako dôležité ukázalo pripraviť testovacie prostredie. To zahŕňalo inštaláciu a konfiguráciu potrebného softvéru, nastavenie testovacích dát a prípadov a prípravu testovacieho hardvéru a potrebných sieťových prostredí.
2. Vykonávanie testovacích scenárov a prípadov - po príprave prostredia sa vykonali testovacie scenáre a prípady podľa navrhnutých postupov a krokov. Testy sa vykonali v súlade s definovanými testovacími scenármi, pričom sa zaznamenávajú vstupy, očakávané výstupy a skutočné výsledky testov.
3. Testovanie rôznych scenárov a kombinácií - implementácia testov zahŕňala testovanie rôznych scenárov a kombinácií, aby sa zabezpečilo, že softvér je testovaný vo viacerých možných situáciách. Toto zahŕňa testovanie hraničných hodnôt, neočakávaných vstupov a rôznych kombinácií funkcionality.
4. Správa chýb a sledovanie výsledkov - počas implementácie testov sa spravovali a dokumentovali chyby. Ak sa nájdu chyby alebo nedostatky, správa chýb umožňuje zaznamenať ich, opísať ich a priradiť im prioritné úlohy pre opravu. Taktiež sa sledovali výsledky testov a vyhodnocovalo sa, či boli dosiahnuté očakávané výsledky a ciele testovania.
5. Správa testovacích prostriedkov - implementácia testov zahŕňala aj správu testovacích prostriedkov, ako sú testovacie dáta, testovacie nástroje a prostredia. Bolo dôležité zabezpečiť, že testovacie prostriedky sú aktuálne, dostupné a správne konfigurované pre vykonávanie testov.

Implementácia testov je kritickou fázou v testovacom procese, ktorá zabezpečuje, že testy sú vykonané a výsledky sú zaznamenané. Správna implementácia testov pomáha odhaliť chyby a nedostatky v softvéri, overiť jeho funkčnosť a zabezpečiť, že softvér je pripravený na nasadenie.

5.4 Hodnotenie a vyhodnotenie výsledkov testov

Hodnotenie a vyhodnotenie výsledkov testov bolo dôležitou súčasťou testovacieho procesu. Po vykonaní testov sa zhromaždili a analyzovali výsledky s cieľom posúdiť kvalitu softvéru a identifikovali prípadné nedostatky, chyby alebo problémy.

Tu je niekoľko kľúčových aspektov hodnotenia a vyhodnotenia výsledkov testov:

1. Porovnávanie očakávaných výsledkov - pri hodnotení výsledkov testov bolo dôležité porovnávať skutočné výsledky s očakávanými výsledkami. Očakávané výsledky sa odvodzujú zo špecifikácií, požiadaviek a predpokladov testovacieho scenára. Ak sa skutočné výsledky zhodujú s očakávanými, môže sa takáto situácia považovať za pozitívny výsledok. Ak sa výsledky líšia, môže to naznačovať prítomnosť chýb alebo nedostatkov.
2. Identifikácia a správa chýb - počas testovania sa objavili chyby, pričom prebehla ich identifikácia a dokumentácia. Chyby sa zaznamenávali v systémoch pre správu chýb. Každá chyba mala doprovodné informácie, ako je opis problému, kroky na reprodukciu a prípadne prílohy (logy, snímky obrazovky). Správa chýb umožnila ich sledovanie a pridelenie kým ďalej ich opravuje.
3. Vyhodnotenie pokrytia testovania - hodnotenie výsledkov testov zahŕňalo aj vyhodnotenie pokrytia testovania. To znamená posúdenie, do akej miery testy pokryli rôzne časti softvéru a jeho funkcionalít. Napríklad sa vyhodnocovalo, koľko percent funkcií bolo testovaných, koľko riadkov kódu bolo pokrytých testami alebo aké scenáre boli vykonané. Vyhodnotenie pokrytia testovania pomohlo identifikovať medzery v testovacom pokrytí a umožnilo plánovať ďalšie testy.
4. Vytváranie testovacích správ - testovacie správy sú dokumenty, ktoré obsahujú zhrnutie výsledkov testovania a vyhodnotenie kvality softvéru. Správy zahŕňajú informácie o vykonaných testoch, výsledky, identifikované chyby a odporúčania pre ďalšie kroky. Testovacie správy boli

zdieľané s tímom vývojárov, projektovým manažmentom a technickými pracovníkmi ako prostriedok komunikácie o stave testovania a kvalite softvéru.

5. Vylepšovanie procesu testovania - na základe vyhodnotenia výsledkov testov sa mohli identifikovať oblasti, kde je potrebné vylepšiť proces testovania. To mohlo zahŕňať úpravu testovacích postupov, rozšírenie testovacieho pokrytia, lepšie zdroje pre testovanie alebo zlepšenie automatizácie testov. Hodnotenie výsledkov môže poskytnúť cenné informácie na zlepšenie kvality softvéru a efektívnosti testovacieho procesu.

Hodnotenie a vyhodnotenie výsledkov testov sa ukázalo ako neoddeliteľnou súčasťou testovacieho procesu a pomohlo zabezpečiť, že softvér je spoľahlivý, funkčný a plní požiadavky. Tieto kroky umožňujú identifikovať nedostatky, chyby a problémy, a následne ich opraviť a zlepšiť kvalitu softvéru.

5.5 Oprava a pretestovanie

Oprava a pretestovanie sú dôležité kroky v testovacom procese, ktoré sa vykonali po identifikácii a správe chýb. Tu je niekoľko kľúčových informácií o oprave a pretestovaní:

1. Identifikácia a správa chýb - počas testovania sa identifikujú chyby a nedostatky v softvéri. Každá chyba by mala byť správne zdokumentovaná s dostatočnými informáciami o probléme.
2. Oprava chýb - po identifikácii chyby nasledovala fáza opravy. Bola analyzovaná chyba, identifikovaná príčina a navrhnuté vhodné opravné opatrenie.
3. Verifikácia opravy - po oprave chyby bolo dôležité overiť, či oprava bola úspešná. Verifikácia sa zameriavala na potvrdenie, že chyba bola skutočne odstránená a že softvér funguje správne. V tejto fáze sa overuje, či opravné opatrenia vyriešili daný problém.
4. Pretestovanie - po verifikácii opravy nasleduje fáza pretestovania. V tejto fáze sa vykonávajú relevantné testy, ktoré boli postihnuté chybou alebo súvisia s opravenou časťou softvéru. Cieľom pretestovania je overiť, že oprava chyby nemá negatívny vplyv na ostatnú funkčnosť a že softvér je stále funkčný a kvalitný.

Oprava a pretestovanie sú kritické kroky v testovacom procese, ktoré zabezpečujú, že chyby sú identifikované a opravené, a že softvér je stabilný, spoľahlivý a funkčný.

5.6 Testovanie prototypu na reálnych dátach

Testovanie prototypu a ideového riešenia sa vykonávalo v skorých fázach vývoja softvéru, kde sa vytvoril funkčný model ideovo-softvérového riešenia v prostredí Microsoft Excel. Cieľom bolo overiť, či koncept, funkčnosť alebo algoritmy predpokladaného vyvíjaného softvéru sú správne, zmysluplné a splnia požiadavky a očakávania. Pri testovaní prototypu vo vzťahu k reálnej elektrickej mikrosieti ide o overenie a validáciu funkčnosti softvéru na základe obmedzenej verzie reálnej elektrickej mikrosiete. Testovanie prototypu sa zameriavalo na skúmanie správnosti algoritmov, správneho spracovania dát a identifikáciu potenciálnych problémov.

Opis vykonávaných činností je možné zhrnúť do nasledujúcich bodov:

1. Simulácia mikrosiete - pre testovanie softvérového prototypu bola vykonaná simulácia mikrosiete. Simulácia je proces, v ktorom sa vytvára bilančný model elektrickej mikrosiete v softvéri, napríklad pomocou softvéru na simuláciu elektrickej siete. V tomto prípade bol použitý Microsoft Excel. Tento model zahŕňa rôzne komponenty a prvky, ako sú zdroje energie, spotrebiče, prenosové linky, transformátory a regulátory. Simulácia umožňuje simulovať prevádzku a správanie mikrosiete v rôznych podmienkach a scenároch.
2. Testovanie softvérového prototypu - po vytvorení simulácie mikrosiete je testovaný softvérový prototyp, ktorý je implementovaný v nástroji Excel. Softvérový prototyp môže obsahovať rôzne funkcie a algoritmy pre riadenie, monitorovanie alebo analýzu mikrosiete. Cieľom testovania je overiť správnosť a funkčnosť týchto funkcií v simulovanom prostredí.
3. Bilancia v mikrosieti po simulácii - po spustení simulácie mikrosiete sa analyzujú výsledky a vyhodnocuje sa bilancia v mikrosieti. Bilancia zahŕňa porovnanie produkcie a spotreby energie v mikrosieti a zistenie, či dochádza k nejakým nepriaznivým javom, ako sú nadmerná produkcia, nedostatok energie alebo nerovnomerné rozloženie záťaže v sieti.

4. Výstupné štatistické parametre - po vykonaní simulácie a testovaní sa zhromažďujú a analyzujú výstupné štatistické parametre. Tieto parametre môžu zahŕňať údaje o produkciách a spotrebách energie, odchýlkach medzi spotrebou a výrobou a iných relevantných merateľných veličinách. Štatistické parametre poskytujú kvantitatívne informácie o správaní mikrosiete a pomáhajú pri hodnotení výkonu a kvality softvérového prototypu.

Testovanie softvérového prototypu vo vzťahu k reálnej elektrickej mikrosieti, s využitím nástroja Excel, umožnilo skúmať a analyzovať správanie mikrosiete a overiť funkčnosť implementovaných funkcií a algoritmov. Tento proces je dôležitý pre vývoj a optimalizáciu softvéru určeného pre riadenie a analýzu mikrogridov a pomáha zabezpečiť, že softvérové riešenie je spoľahlivé a efektívne v reálnom prostredí elektrickej mikrosiete.

5.6.1 Popis charakteristík prototypu využívajúceho prostredie Microsoft Excel

Za účelom testovania navrhnutého systému optimalizácie mikrogridu bola v tabuľkovom prostredí Microsoft Excelu skonštruovaná ideová predstava základnej funkčnosti budúceho predpokladaného optimalizačného prostredia. Ideová predstava predpokladala využitie štatistických dát, najmä časových radov výroby a spotreby elektriny zodpovedajúce charakteristikám jednotlivých komponentov. Išlo najmä o charakteristiky zdrojov s premenlivou povahou výroby elektriny, a to fotovoltických a veterných elektrární. Výrobné diagramy kogeneračných jednotiek majú vo väčšine prípadov stabilný charakter výroby. Pre tento typ zdroja sú skôr typické odštoky.

Koncepcne bol prototyp navrhnutý tak, aby pracoval s jedným vstupom vo forme časového radu spotreby elektriny. Kategória spotrebného miesta nebol podmienená, čo znamená, že prototyp pracoval za rovnakých podmienok, či už išlo o diagram spotreby školského objektu alebo priemyselného objektu. Rozhodujúce parametre pre optimalizáciu boli maximálna veľkosť výkonu a objem spotrebovanej elektriny v diagrame.

Časové rady výroby elektriny z fotovoltickej a veternej elektrárne boli dopredu prednastavené z anonymizovanej dátovej sady. Princíp optimalizácie vyžadoval normalizovaný priebeh, pričom počas optimalizácie sa tento priebeh obsahovo zvyšoval alebo znižoval. Ak bol napríklad k dispozícii priebeh pre fotovoltickú elektrárňu s výkonom 1 kW, tak pre výkon 10 kW stačilo vynásobiť daný priebeh číslom 10. Taktu sa rešpektovala variabilita obnoviteľných zdrojov energie pri optimalizácii, ktorá súčasne bola veľmi dôležitá pre to, aby optimalizácia pracovala s autentickou variabilitou obnoviteľných zdrojov. Príklad z tabuľkového prostredia prototypu je uvedený v tabuľke AA a BB.

Ako reálny komponent - odber elektrickej energie bol pre optimalizáciu použitý odberový diagram z výskumnej lokality Trnávka. Tento reálny komponent je možné nahradiť ľubovoľným časovým radom odberu elektrickej energie. Časové rady spotreby elektriny sú si navzájom veľmi podobné, pretože zahŕňajú častokrát podobné charakteristiky ako sezónnosť, kalendárny efekt alebo trend.

Významnou časťou optimalizácie sa ukázalo byť nastavenie vstupných parametrov optimalizácie pre zaistenie zmysluplných a technicky logických výstupov optimalizácie. Potvrdili sa teoretické poznatky z odbornej literatúry z ktorých vyplýva, že proces optimalizácie musí byť pod dohľadom človeka a týka sa to ako vstupov, tak aj výstupov optimalizácie.

V prípade, keď sú limity vstupných parametrov optimalizácie neobmedzené, napríklad veľkosť inštalovaného výkonu fotovoltickej elektrárne nie je v reálnych limitoch alebo limity sú rovno zanedbané, môže nastať situácia, že inštalovaný výkon po optimalizácii bude záporný alebo veľmi nízky. Jeden z dôvodov je, že fotovoltická elektrárňu má nízky koeficient využitia, to znamená, že na jednotku inštalovaného výkonu vyrobí za nejaké obdobie, najčastejšie sa používa rok, oveľa menej elektrickej energie v porovnaní s kogeneračnou jednotkou alebo veternou elektrárnou. Táto charakteristika má potom dôsledok vo forme potlačenia vplyvu tohto typu zdroja na takmer zanedbateľnú úroveň vo výsledkoch optimalizácie. Riešenie tohto problému spočíva v zedefinovaní limitov možného inštalovaného výkonu tak, aby bola splnená podmienka definovaná optimalizačnou praxou. Môže ísť napríklad o zedefinovanie podmienky tak, aby fotovoltická elektrárňu pokryla aspoň 10 % predpokladanej spotreby elektriny za optimalizačné obdobie. Obdobne je vhodné postupovať aj pri nastavovaní limitov pre veternú elektrárňu a kogeneračnú jednotku. Vzhľadom na to, že kogeneračná jednotka má vysoký ročný koeficient využitia inštalovaného výkonu, bude mať tendenciu dominovať vo výrobe energie v rámci výsledku optimalizácie.

Obmedzenie inštalovaného výkonu zdrojov pre výskumnú lokalitu Trnávka je zobrazené v tabuľke CC. Hodnoty obmedzenia vychádzajú z maximálnej a minimálnej hodnoty časového radu spotreby elektriny v lokalite Trnávka.

Tabuľka 11 Obmedzenia inštalovaného výkonu pre výskumnú lokalitu Trnávka

	Obmedzenia inštalovaného výkonu zdrojov [kW]	
Kogeneračná jednotka [kW]	2,19	14,62
Fotovoltaická elektrárň [kW]	2,19	14,62
Veterná elektrárň [kW]	2,19	14,62

Tabuľka 12 - Príklad tabuľkového prostredia prototypu

Dátum	Odber [kW]	Kogeneračná jednotka [kW]	Fotovoltaická elektrárň [kW]	Veterná elektrárň [kW]
10.6.2022 0:00	3,092	2,81	0,00	0,1001
10.6.2022 0:15	3,276	2,81	0,00	0,0875
10.6.2022 0:30	3,504	2,82	0,00	0,0803
10.6.2022 0:45	3,176	2,82	0,00	0,0780
10.6.2022 1:00	3,368	2,81	0,00	0,0735
10.6.2022 1:15	3,42	2,82	0,00	0,0713
10.6.2022 1:30	3,056	2,82	0,00	0,0713
10.6.2022 1:45	3,34	2,82	0,00	0,0757
10.6.2022 2:00	3,268	2,82	0,00	0,0851
10.6.2022 2:15	3,212	2,82	0,00	0,0924
10.6.2022 2:30	2,96	2,82	0,00	0,1081
10.6.2022 2:45	2,712	2,82	0,00	0,1281
10.6.2022 3:00	3,136	2,82	0,00	0,1531
10.6.2022 3:15	3,212	2,82	0,00	0,1771
10.6.2022 3:30	3,032	2,82	0,00	0,2106
10.6.2022 3:45	3,056	2,82	0,00	0,2431
10.6.2022 4:00	3,252	2,81	0,00	0,2782
10.6.2022 4:15	2,74	2,82	0,00	0,3206
10.6.2022 4:30	4,792	2,82	0,00	0,3663
10.6.2022 4:45	3,796	2,81	0,00	0,4151
10.6.2022 5:00	8,468	2,82	0,00	0,4671
10.6.2022 5:15	4,82	2,81	0,00	0,5223
10.6.2022 5:30	4,936	2,82	0,00	0,5739
10.6.2022 5:45	4,192	2,81	0,00	0,6284
10.6.2022 6:00	3,788	2,82	0,00	0,6859
10.6.2022 6:15	4,676	2,81	0,00	0,7469
10.6.2022 6:30	4,9	2,81	0,00	0,8372
10.6.2022 6:45	4,092	2,81	0,00	0,9838
10.6.2022 7:00	4,296	2,82	0,00	1,1514
10.6.2022 7:15	7,02	2,81	0,03	1,3417
10.6.2022 7:30	4,168	2,81	0,24	1,5541
10.6.2022 7:45	3,676	2,81	1,03	1,7875
10.6.2022 8:00	3,58	2,81	1,94	2,0574
10.6.2022 8:15	3,544	2,81	2,99	2,3309
10.6.2022 8:30	3,9	2,81	3,95	2,5291
10.6.2022 8:45	6,864	2,81	4,68	2,6218
10.6.2022 9:00	5,48	2,81	5,48	2,7351
10.6.2022 9:15	9,764	2,81	6,04	2,8505
10.6.2022 9:30	6,2	2,81	6,61	2,9879
10.6.2022 9:45	6,692	2,81	7,10	3,1482
10.6.2022 10:00	5,444	2,81	7,54	3,3119
10.6.2022 10:15	8,524	2,81	7,55	3,4790
10.6.2022 10:30	5,336	2,81	8,01	3,5427
10.6.2022 10:45	3,76	2,81	8,07	3,5002
10.6.2022 11:00	4,74	2,81	8,26	3,4579
10.6.2022 11:15	4,632	2,81	7,48	3,4159
10.6.2022 11:30	3,94	2,81	8,13	3,3533
10.6.2022 11:45	3,608	2,81	7,81	3,2501

Tabuľka 13 - Príklad tabuľkového prostredia prototypu

Hľadané veľkosti inštalovaného výkonu jednotlivých prvkov zdrojovej základne	
Kogeneračná jednotka [kW]	2,94
Fotovoltaická elektrárňa [kW]	13,29
Veterná elektrárňa [kW]	11,68
Štatistické znaky a vstupy pre optimalizáciu	
Odber	
Maximálny štvrťhodinový odber v [kW]	14,62
Maximálny štvrťhodinový odber [kWh]	0,00
Celkový odber [kWh/týždeň]	1 080,25
Celkový odber [MWh/týždeň]	1,08
Priemerný štvrťhodinový odber [kW]	5,62
Kogeneračná jednotka	
Maximálna štvrťhodinová výroba [kW]	2,83
Týždenná suma výroby kogeneračnej jednotky [kWh/týždeň]	429,61
Fotovoltaická elektrárňa	
Maximálna štvrťhodinová výroba [kW]	9,45
Týždenná suma výroby fotovoltaickej elektrárne [kWh/týždeň]	221,36
Veterná elektrárňa	
Maximálna štvrťhodinová výroba [kW]	10,54
Týždenná suma výroby veternej elektrárne [kWh/týždeň]	429,28
Mikrogrid - bilancia	
Maximálna štvrťhodinová výroba [kW]	21,89
Maximálna štvrťhodinová bilancia [kW]	11,27
Celková výroba zdrojov v MG [kWh/týždeň]	1 080,25
Účelová funkcia - minimalizácia rozdielu ročnej výroby a spotreby	
Celkový odber [kWh/rok]	1 080,25
Celkový odber + 20 % [kWh/týždeň]	1296,2976
Celková výroba zdrojov v MG [kWh/týždeň]	1 080,25
Rozdiel - absolútna hodnota	0,00

Tabuľka 14 - Výstupy inštalovaného výkonu jednotlivých zdrojov elektrickej energie po optimalizácii

Hľadané veľkosti inštalovaného výkonu jednotlivých prvkov zdrojovej základne	
Kogeneračná jednotka [kW]	2,94
Fotovoltaická elektrárňa [kW]	13,29
Veterná elektrárňa [kW]	11,68

V tabuľke CC sú uvedené výstupy inštalovaných výkonov po optimalizácii. Tieto hodnoty spĺňajú dodržanie limitov, pri ktorých nemohla výstupná hľadaná hodnota prekročiť maximálnu hodnotu odberu. Minimálna hodnota zodpovedá 15 % z maximálnej hodnoty odberu. Je potrebné poznamenať, že uvedené hľadané hodnoty sú nezaokrúhlené čísla a predstavujú len úvodnú časť výstupu po optimalizácii. Tieto hodnoty sú len orientačné a je potrebné ich ďalšie spracovanie.

Pri dimenzovaní elektrických zariadení, elektrických vedení a elektrárni je vždy potrebné uvažovať s rezervou inštalovaného výkonu napríklad kvôli stratám alebo predpokladanej zvýšenej spotrebe v budúcnosti. Pri určovaní veľkosti inštalovaného výkonu zdrojov v mikrogride je preto vhodné postupovať podobne. Kogeneračná jednotka vzhľadom na svoju schopnosť dodávať stabilný výkon v potrebnom objeme je chrbticou celého mikrogridu. Inštalovaný výkon by mal byť väčší, než je výsledok z optimalizácie. Je vhodné naviazať číslo z optimalizácie na najbližšie vyššie číslo inštalovaného výkonu komerčne vyrábaných kogeneračných jednotiek. Ak napríklad je najbližšie vyššie číslo 3 kW elektrického výkonu, potom by mala byť použitá hodnota pri finálnom návrhu mikrogridu 3 kW. Takto sa zabezpečí aj potrebná rezerva. Táto logika platí aj pri veterných turbínach, ktoré sa tiež vyrábajú v typových radoch. Najbližší vyšší výkon komerčnej turbíny môže byť napríklad 15 kW. Táto hodnota závisí aj od poveternostných podmienok. Inštalovaný výkon fotovoltaickej elektrárne je jednoduchšie zrealizovať, pretože závisí najmä od počtu panelov, ktorý je možné relatívne ľahko meniť. Vzhľadom na degradáciu panelov, je vhodné pridať aspoň 1 kW inštalovaného výkonu navyše. Významnú rolu hrá tiež kategória odberného miesta. Pre priemyselné odbory sa odporúča vypracovať scenáre pričom zdrojová a technická stabilita mikrogridu má prednosť pred ekonomickými kritériami.

Optimalizácia má za cieľ nájsť také hodnoty inštalovaných výkonov, aby predpokladaná sumárna odchýlka výroby a spotreby, za dané optimalizačné obdobie, bola čo najnižšia a mikrogrid tak bol v čo najmenšej miere závislý od odberu elektrickej energie z nadradenej sústavy. Na Obrázku 14 je zobrazený priebeh výroby a spotreby v mikrogride. Je možné si všimnúť, že zdrojová základňa pokrýva najmä základné zaťaženie. Celková sumárna výroba všetkých zdrojov energia za daný týždeň dosiahla 1,08 MWh elektrickej energie.

Výrobné profily fotovoltaickej a veternej elektrárne sú zobrazené na Obrázku 15. Výroba elektrickej energie z vetra má takmer nepretržitý charakter aj v nočných hodinách, kedy fotovoltaická elektrárň má nulovú výrobu. Počas šiesteho dňa prevádzky mikrogridu došlo k výdatnej výrobe z veternej elektrárne, čím v kombinácii s výrobou z fotovoltaickej elektrárne vznikol nadbytok vyrábanej energie. V praxi by takýto nadbytok musel byť uložený do batérového systému alebo injektovaný do nadradenej sústavy. Na obrázku 1 je zobrazené toto obdobie v súčtovej forme aj spolu s diagramom výroby kogeneračnej jednotky. Z obrázku je možné odčítať špičkovú výrobu viac ako 20 kW, pričom maximálny odber bol 14,62 kW.

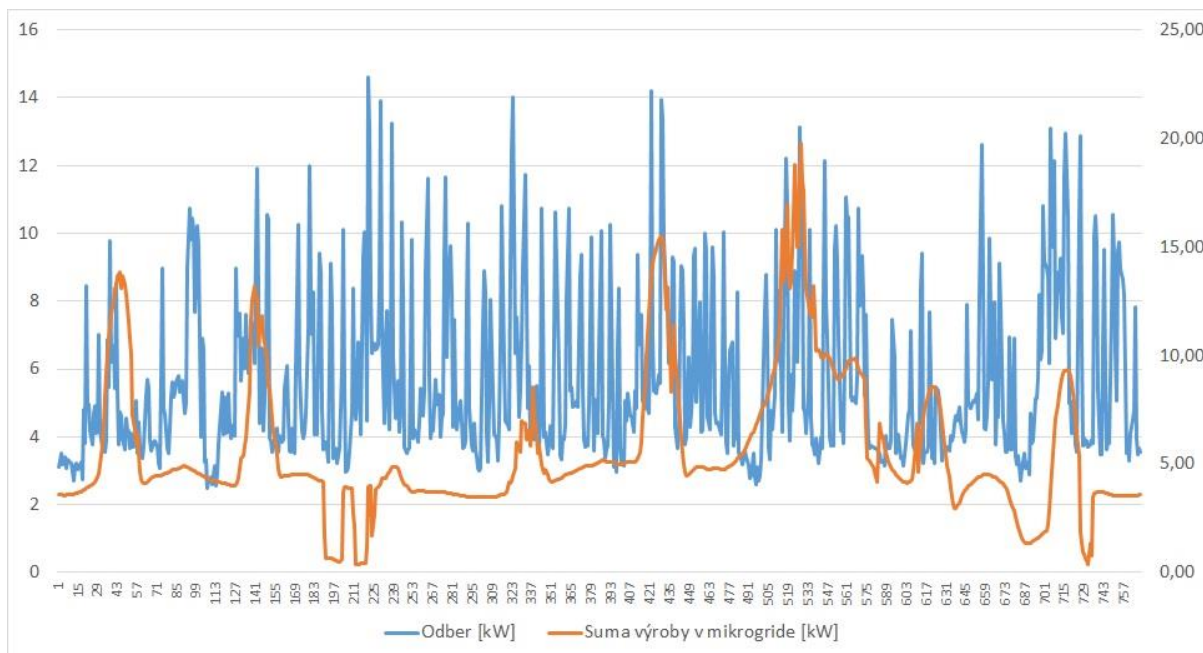
Optimalizácia na základe reálnych nameraných historických dát poskytuje ucelený vzhľad na prevádzku mikrogridu v kvázi reálnej podobe. Na testovanie prototypu sa použil zásuvný model „riešiteľ“, ktorý je bežnou súčasťou prostredia Microsoft Excel. Na Obrázku 16 sú zobrazené parametre pri optimalizácii. Bol použitý nelineárny algoritmus GRG (ang. generalized reduced gradient). "Generalized Reduced Gradient" (GRG) je algoritmus na numerickú optimalizáciu, ktorý je využívaný v softvérovom nástroji Microsoft Excel na riešenie problémov nelineárneho programovania. GRG je metóda hľadania lokálneho extrému v problémoch s obmedzeniami.

Hlavným cieľom GRG algoritmu je nájsť hodnoty premenných, ktoré minimalizujú alebo maximalizujú danú objektívnu funkciu, pričom sú dodržané dané obmedzenia. Algoritmus postupne upravuje hodnoty premenných tak, aby dosiahol čo najlepšie možné riešenie. GRG je založený na kombinácii metód redukovaných gradientov a kvadratického programovania.

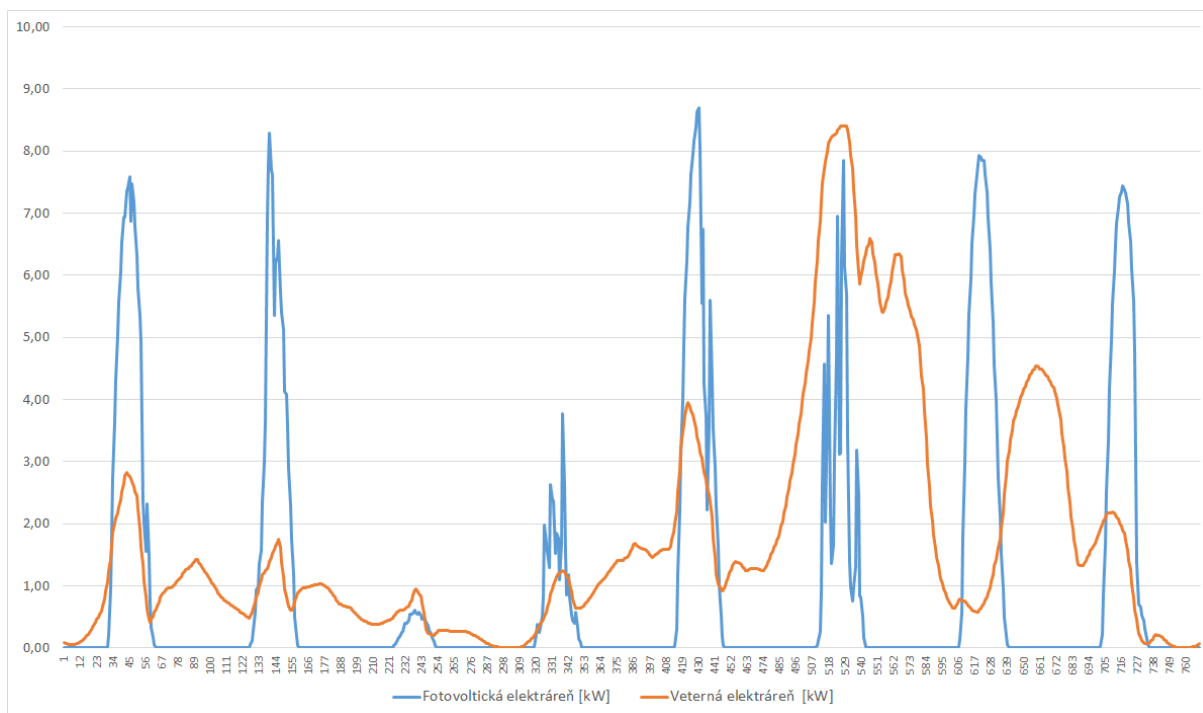
Postup algoritmu GRG je nasledovný:

1. Inicializácia - algoritmus začína s počiatočnými hodnotami pre premenné.
2. Výpočet gradientov - v každom kroku sa vypočítajú gradienty objektívnej funkcie a obmedzení vzhľadom na premenné. Gradient objektívnej funkcie udáva smer rastu alebo klesania hodnoty, ktorú chceme minimalizovať alebo maximalizovať.
3. Aktualizácia premenných - na základe gradientov sa aktualizujú hodnoty premenných tak, aby sa priblížili k optimálnemu riešeniu. Táto aktualizácia je vykonávaná iteratívne, pričom sa menia hodnoty premenných v smeroch, ktoré vedú k zlepšeniu objektívnej funkcie.
4. Kontrola obmedzení - po aktualizácii premenných sa kontrolujú obmedzenia. Ak sú niektoré obmedzenia porušené, algoritmus sa snaží nájsť novú hodnotu premenných, ktorá by obmedzenia dodržiavala.
5. Konvergencia: - algoritmus pokračuje v opakovanej aktualizácii premenných a kontrole obmedzení, až kým nedosiahne určené kritériá zastavenia. Tieto kritériá môžu zahŕňať maximálny počet iterácií, dosiahnutie stanovenej presnosti alebo dosiahnutie stabilného riešenia.

Algoritmus GRG sa používa v Microsoft Excel ako nástroj na optimalizáciu a riešenie problémov nelineárneho programovania s obmedzeniami. Pomocou tohto algoritmu je možné nájsť optimálne hodnoty premenných pre určité problémy, čo je užitočné napríklad pri modelovaní a simulácii rôznych procesov a rozhodovacích situácií.



Obrázok 14 Porovnanie diagramu spotreby a sumárnej výroby v mikrogride



Obrázok 15 Grafické porovnanie výrobných diagramov obnoviteľných zdrojov energie v mikrogride

Parametre doplnku Riešiteľ ✕

Nastaviť cieľ: ↑

Do: Maximum Minimum Hodnota:

Zmenou premenných buniek: ↑

Podlieha obmedzeniam:

SNS2 <= SQS2
 SNS2 >= SP52
 SNS3 <= SQS3
 SNS3 >= SP53
 SNS4 <= SQS4
 SNS4 >= SP54

Pridať

Zmeniť

Odstáňať

Obnoviť všetko

Načítať/Uložiť

Vytvorte nezápornú hodnotu premenných, ktoré sú bez obmedzeni

Vybrať metódu riešenia: Možnosti

Metóda riešenia

Ak chcete v doplnku Riešiteľ riešiť spoj. nelineárne problémy, vyberte nástroj Nelineárny algoritmus GRG. Ak chcete riešiť lineárne problémy, vyberte nástroj Simplex LP algoritmus. Ak chcete riešiť nespojité problémy, vyberte nástroj Evolučný algoritmus.

Pomocník
Riešiť
Zavrieť

Obrázok 16 Príklad obmedzujúcich podmienok pred spustením optimalizácie

6 OVERENIE FUNKČNOSTI INTEGROVANÉHO RIEŠENIA

Overenie funkčnosti integrovaného riešenia je dôležitým procesom v rámci vývoja MicroGridBoxu® a bolo podmienené testovaním prototypu, ktoré je uvedené v predchádzajúcej kapitole. Overenie ako činnosť slúžila na zistenie, či softvérové riešenie ako systém spĺňa požadované špecifikácie, funkčné požiadavky a predpokladané očakávania používateľov. Cieľom overovania bolo identifikovať chyby, nedostatky a problémy v riešení a zabezpečiť, aby bolo spoľahlivé, stabilné a efektívne.

Aspekty overovania v rámci vývoja softvéru:

1. Testovanie funkčnosti - overovanie zahŕňalo testovanie funkčnosti riešenia. To znamená, že sa overovalo, či jednotlivé komponenty a funkcie Mikrogridboxu fungujú správne a vykonávajú očakávané úlohy. Testovanie funkčnosti zahŕňalo jednotlivé moduly, integráciu komponentov a systémové testovanie.
2. Testovanie výkonu - okrem testovania funkčnosti sa venovala pozornosť aj testovaniu výkonu riešenia. Cieľom bolo zistiť, ako dobre sa softvérové riešenie správa pri väčšom zaťažení, ako rýchlo vykonáva úlohy a aké množstvo zdrojov spotrebuje. Testovanie výkonu bolo dôležité pre optimalizáciu riešenia a zabezpečenie jeho efektívneho fungovania v reálnom prostredí.
3. Testovanie použiteľnosti - overovanie zahŕňa aj testovanie použiteľnosti riešenia. Toto sa zameriavalo na skúmanie, ako dobre je softvér použiteľný pre koncových používateľov. Testovanie použiteľnosti zahŕňalo testovanie používateľského rozhrania, navigácie, interakcie a celkovej používateľskej skúsenosti.
4. Testovanie bezpečnosti - v dnešnej dobe je bezpečnosť softvéru kľúčovou prioritou. Overovanie zahŕňa aj testovanie bezpečnosti softvéru, aby sa zistili potenciálne zraniteľnosti, slabé miesta a možné hrozby.
5. Overovanie kompatibility - softvér často musí fungovať v rôznych prostrediach, operačných systémoch a hardvérových konfiguráciách. Preto bolo dôležité overiť kompatibilitu softvéru s rôznymi platformami a zabezpečiť, že funguje správne v rôznych prostrediach.

Overovanie je dôležitou súčasťou vývoja softvéru, ktorá pomohlo identifikovať a odstrániť chyby a zabezpečiť, aby softvér bol spoľahlivý, funkčný a efektívny. Riadne overovanie zvyšuje kvalitu softvéru a zaisťuje, že splní očakávania užívateľov a ich požiadavky.

6.1 Testovanie funkčnosti

Testovanie funkčnosti je proces, ktorý slúžil na overenie, či softvérové riešenie vykonávalo očakávané úlohy a splnilo požadované funkčné požiadavky. Tu je niekoľko dôležitých aspektov testovania funkčnosti:

1. Testovacie prípady - testovanie funkčnosti sa vykonávalo na základe vytypovaných testovacích prípadov. Testové prípady sú definované scenáre, ktoré popisujú očakávané vstupy, kroky a očakávané výstupy pre konkrétnu funkčnosť riešenia. Testovacie prípady umožnili systematické a opakovateľné testovanie, a tým aj identifikáciu chýb a nedostatkov.
2. Testovacie prostredie - pre testovanie funkčnosti sa vytvorilo vhodné testovacie prostredie. To môže byť prostredie s reálnymi dátami, simulované prostredie alebo testovacie prostredie s umelými dátami. Testovacie prostredie simulovalo reálne použitie softvéru a umožnilo testovať rôzne scenáre a podmienky.
3. Správa testovacích výsledkov - po vykonaní testov sa zhromaždili testovacie výsledky. Výsledky sa analyzovali a porovnávali s očakávanými výsledkami. V prípade nájdenia rozdielov alebo nedostatkov sa tieto identifikujú. Správa testovacích výsledkov je dôležitá pre sledovanie postupu testovania a informovanie vývojového tímu Mikrogridboxu o výsledkoch a potenciálnych problémoch.
4. Chybové správy a opravy - testovanie funkčnosti umožnilo odhaliť chyby a nedostatky v softvéri. Pri testovaní sa zaznamenávajú chybové správy, ktoré identifikujú problémy a poskytujú informácie potrebné na ich opravu. Po identifikácii chýb sa vykonáva proces opravy, kde sa chyby odstraňujú a softvér sa znovu testuje na overenie, či boli opravy úspešné.
5. Testovacie metódy a techniky - pri testovaní funkčnosti sa používajú rôzne metódy a techniky. Medzi najpoužívanejšie patria testovanie čiernej skrinky, testovanie bielych skriniek, testovanie

jednotiek a integračné testovanie. Každá metóda má svoje vlastné prístupy a ciele a pomáha pri zistení rôznych typov chýb a nedostatkov.

Testovanie funkčnosti je neoddeliteľnou súčasťou vývoja softvéru a pomohlo zabezpečiť, že softvérové riešenie bolo schopné vykonávať požadované úlohy a dosiahnuť očakávané výsledky. Riadne testovanie funkčnosti zvyšuje kvalitu softvéru a znižuje riziko chýb a problémov v produkčnom prostredí pri nasadení.

6.2 Testovanie výkonu

Testovanie výkonu je proces, ktorý sa zameriava na posúdenie výkonnosti softvéru v rôznych podmienkach zaťaženia. Cieľom testovania výkonu je zistiť, ako rýchlo a efektívne softvér reaguje na určité podnety a ako dobre zvláda väčšie množstvo používateľov alebo náročnejšie úlohy.

Tu je niekoľko dôležitých aspektov testovania výkonu:

1. **Záťažové testovanie** - záťažové testovanie sa vykonáva s cieľom zistiť, ako dobre sa softvér správa pri zvýšenom zaťažení. Simulujú sa situácie, kedy je softvér vystavený väčšiemu počtu súbežných používateľov, veľkému objemu dát alebo vyššiemu zaťaženiu sieťových zdrojov. Cieľom je zistiť, ako sa softvér správa v takýchto extrémnych podmienkach a či udrží svoju výkonnosť a stabilitu.
2. **Testovanie odozvy** - testovanie odozvy sa zameriava na meranie rýchlosti a času, ktorý softvér potrebuje na vykonanie konkrétnej úlohy. Týmto spôsobom sa overuje, ako rýchlo softvér reaguje na používateľské vstupy alebo systémové požiadavky. Testovanie odozvy je dôležité pre zistenie, či softvér splní požadované časové obmedzenia a poskytne rýchle a plynulé používateľské skúsenosti.
3. **Testovanie škálovateľnosti** - testovanie škálovateľnosti sa zameriava na posúdenie schopnosti softvéru rásť a prispôbiť sa zvyšujúcemu sa objemu používateľov, dát alebo zaťaženia. Cieľom je zistiť, ako dobre softvér zvláda zvýšenú záťaž a či dokáže udržať svoju výkonnosť a kvalitu pri raste požiadaviek. Testovanie škálovateľnosti je dôležité pre zabezpečenie, aby softvér bol schopný rásť a prispôbiť sa narastajúcim potrebám a objemom.
4. **Meranie zdrojov** - pri testovaní výkonu sa zisťuje aj miera spotreby zdrojov, ako sú pamäť, procesorový čas, diskový priestor a sieťová šírka pásma. Meranie zdrojov umožňuje identifikovať prípadné úniky pamäte, neefektívne použitie zdrojov alebo obmedzenia, ktoré môžu ovplyvniť výkon softvéru.
5. **Optimalizácia a ladenie** - výsledky testovania výkonu môžu poskytnúť dôležité informácie o oblastiach softvéru, kde je potrebné vykonať optimalizáciu a ladenie. Tieto úpravy a zlepšenia môžu zvýšiť celkový výkon a efektivitu softvéru.

Testovanie výkonu je dôležitý proces, ktorý pomáha zabezpečiť, že softvér je dostatočne výkonný a stabilný pre požadované úlohy. Identifikuje slabé miesta, problémy so škálovateľnosťou a umožňuje ich nápravu pred nasadením softvéru do produkčného prostredia.

6.3 Testovanie použiteľnosti

Testovanie použiteľnosti je proces, ktorý sa zameriava na posúdenie toho, ako dobre softvérové riešenie spĺňa potreby a očakávania používateľov. Cieľom testovania použiteľnosti je zistiť, ako jednoduché a efektívne je softvér používať, ako dobre sa používateľ dokáže orientovať v rozhraní a dosiahnuť svoje ciele.

Aspekty testovania použiteľnosti:

1. **Testovacie scenáre** - testovanie použiteľnosti sa vykonávalo na základe testovacích scenárov, ktoré boli navrhnuté tak, aby simulovali reálne používanie softvéru. Scenáre obsahovali konkrétne úlohy, ktoré majú potenciálni používatelia splniť, a zahrňujú rôzne typické akcie a kroky, ktoré by mohli vykonávať.
2. **Testovacie prostredie** - pre testovanie použiteľnosti sa vytvorilo vhodné testovacie prostredie, ktoré čo najviac simuluje reálne použitie softvéru. To môže zahŕňať rôzne typy zariadení, operačné systémy a konfigurácie, aby sa zabezpečila kompatibilita a použiteľnosť softvéru v rôznych prostrediach.

3. Zbieranie a vyhodnocovanie spätnej väzby - počas testovania použiteľnosti sa zhromaždila spätná väzba od testerov. Táto spätná väzba môže pochádzať z otázok a odpovedí, pripomienok a názorov alebo z pozorovania používateľov pri používaní softvéru. Táto spätná väzba je potom analyzovaná a vyhodnotená, aby sa identifikovali problémy a nedostatky v použiteľnosti softvéru.
4. Testovacie metódy a techniky - pri testovaní použiteľnosti sa používali rôzne metódy a techniky. Medzi najčastejšie v praxi patria testovanie s používateľskými skupinami, kde sa sleduje a vyhodnocuje, ako rôzni používatelia interagujú so softvérom. Ďalšou metódou je heuristické hodnotenie, kde sa aplikuje sada heuristik alebo pravidiel na hodnotenie použiteľnosti softvéru.
5. Iteratívny prístup - testovanie použiteľnosti sa často vykonáva iteratívne, pričom sa spätná väzba a zistenia z jedného kola testovania používajú na vylepšenie softvéru a nasledujúcich kôl testovania. Týmto spôsobom sa postupne zlepšuje použiteľnosť softvéru a znižuje sa množstvo chýb a problémov, ktoré by mohli ovplyvniť používateľskú skúsenosť.

Testovanie použiteľnosti je kritickým procesom, ktorý zabezpečuje, že softvér je intuitívny, ľahko použiteľný a poskytuje pozitívnu používateľskú skúsenosť. Pomáha identifikovať problémy, zlepšovať rozhranie a optimalizovať tok úloh, čo vedie k zvýšeniu prijateľnosti a spokojnosti používateľov s produktom.

6.4 Testovanie bezpečnosti

Testovanie bezpečnosti je proces, ktorý sa zameriava na identifikáciu bezpečnostných slabín a rizík v softvérovom riešení. Jeho cieľom je zabezpečiť, že softvér je odolný voči rôznym druhom útokov a že chráni citlivé dáta a zdroje pred neoprávneným prístupom.

Aspekty testovania bezpečnosti:

1. Penetračné testovanie - penetračné testovanie je metóda, pri ktorej je snaha preniknúť do softvéru alebo systému s cieľom odhaliť bezpečnostné chyby. Tester sa snaží využiť rôzne techniky a nástroje na identifikáciu slabých miest v systéme a na prekonanie bezpečnostných mechanizmov. Týmto spôsobom sa bežne zisťuje, či je softvér odolný voči útokom a či sú implementované bezpečnostné opatrenia účinné.
2. Skenovanie zraniteľností - skenovanie zraniteľností je automatizovaný proces, pri ktorom sa používajú špecializované nástroje na skenovanie softvéru a identifikáciu známych zraniteľností. Tieto nástroje skúmajú softvér na prítomnosť známych bezpečnostných chýb, ako sú napríklad neaktualizované verzie softvéru, nedostatočné zabezpečenie komunikácie alebo zraniteľné konfigurácie. Skenovanie zraniteľností pomáha odhaliť potenciálne problémy, ktoré by mohli byť zneužitú na útok.
3. Testovanie autentifikácie a autorizácie - testovanie autentifikácie a autorizácie sa zameriavalo na overenie, či sú implementované mechanizmy autentifikácie (overovania identity používateľov) a autorizácie (určovanie prístupových práv) spoľahlivé a bezpečné. Tester sa snaží identifikovať možné slabiny, ako sú slabé heslá, neodborne implementované autentifikačné postupy alebo nedostatočné kontroly prístupových práv.
4. Testovanie odolnosti voči útokom - testovanie odolnosti voči útokom je proces, pri ktorom sa simulujú rôzne druhy útokov s cieľom zistiť, ako dobre sa softvér bráni proti nim. Tester sa pokúšajú vykonávať rôzne typy útokov, ako sú napríklad útoky typu DoS (Denial of Service), útoky na únik citlivých dát alebo útoky na bezpečnostné medzery. Týmto spôsobom sa overuje, ako dobre je softvér pripravený na odolávanie reálnym útokom.
5. Revízia zdrojového kódu - revízia zdrojového kódu je proces, pri ktorom sa analyzuje zdrojový kód softvéru s cieľom identifikovať bezpečnostné chyby a slabiny. Skúmanie zdrojového kódu umožňuje odhaliť potenciálne bezpečnostné problémy, ako sú napríklad neodborne implementované funkcie overovania, otvorené cesty k útočníkom alebo nedostatočné spracovanie používateľského vstupu.

Testovanie bezpečnosti bolo nevyhnutnou súčasťou vývoja softvérového riešenia, pretože pomáha zabezpečiť, že softvér je odolný voči útokom a chráni dáta a zdroje. Identifikuje bezpečnostné chyby a riziká, čo umožňuje ich nápravu a zlepšenie bezpečnosti softvéru pred jeho nasadením do produkčného prostredia.

6.5 Overovanie kompatibility

Overovanie kompatibility je proces, ktorý sa zaoberá testovaním softvéru s rôznymi kombináciami hardvéru, operačných systémov, prehliadačov a iných prostredí, aby sa zistilo, či softvér správne funguje a je kompatibilný s danými prostrediami.

Aspekty overovania kompatibility:

1. Overovanie operačného systému - softvér musí byť testovaný na rôznych operačných systémoch, ako sú Windows, macOS, Linux a ďalšie. Testuje sa, či softvér je kompatibilný s daným operačným systémom, správne sa inštaluje, spúšťa a pracuje bez problémov.
2. Overovanie hardvéru - rôzne typy hardvéru, ako napríklad procesory, grafické karty, pamäte a periférne zariadenia, môžu mať rôzne vlastnosti a špecifikácie. Softvér sa testuje na rôznych kombináciách hardvéru, aby sa zistilo, či funguje správne a využíva dostupné zdroje efektívne.
3. Overovanie prehliadačov - ak je softvér určený na webové prostredie, je dôležité testovať jeho kompatibilitu s rôznymi prehliadačmi, ako sú Chrome, Firefox, Safari, Edge a ďalšie. Testuje sa správne zobrazenie a funkčnosť softvéru v rôznych prehliadačoch, aby sa zabezpečilo, že používatelia budú mať konzistentnú skúsenosť bez ohľadu na to, ktorý prehliadač používajú.
4. Overovanie prostredí - softvér môže byť nasadený v rôznych prostrediach, ako sú vývojové, testovacie a produkčné prostredia. Testuje sa, či softvér správne funguje a je kompatibilný s danými prostrediami, či sú splnené všetky požiadavky a či sa zachováva integrita dát a bezpečnosť.

Cieľom overovania kompatibility je zabezpečiť, že softvér je schopný pracovať v rôznych prostrediach a s rôznymi kombináciami hardvéru a softvéru. Tým sa zabezpečuje, že používatelia budú mať konzistentnú a funkčnú skúsenosť pri používaní softvéru, bez ohľadu na to, aké prostredie či kombináciu používajú.

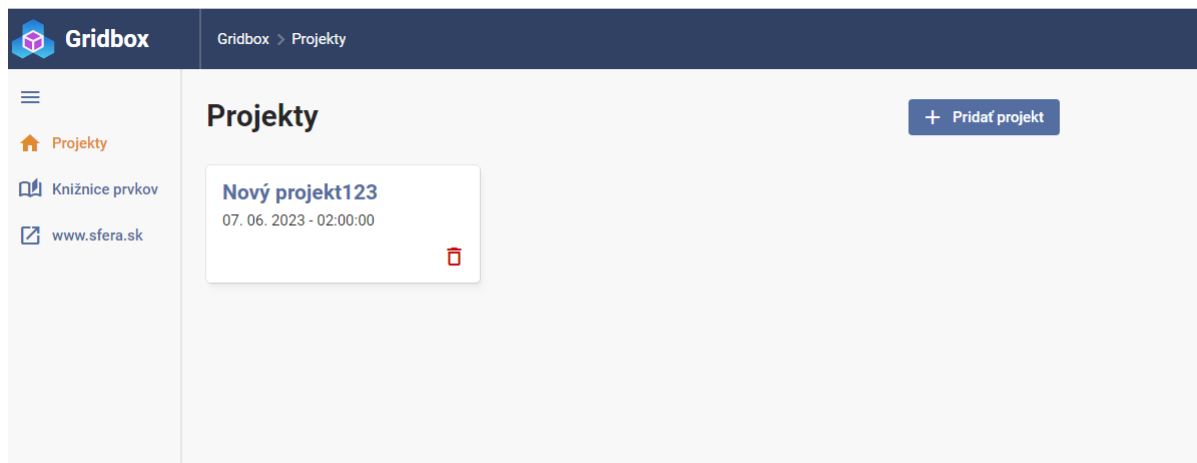
6.6 Grafická vizualizácia integrovaného riešenia Mikrogridboxu

Na Obrázkoch 17 až 21 je zobrazené používateľské rozhranie integrovaného riešenia MicroGridBoxu®. API (Application Programming Interface) je sada definovaných pravidiel, protokolov a nástrojov, ktoré umožňujú komunikáciu medzi rôznymi softvérovými komponentami. Je to rozhranie, cez ktoré môžu aplikácie vzájomne komunikovať a vymieňať si dáta a funkcie. API určuje, akým spôsobom môže iná aplikácia pristupovať k funkcionalitám a službám daného softvérového systému.

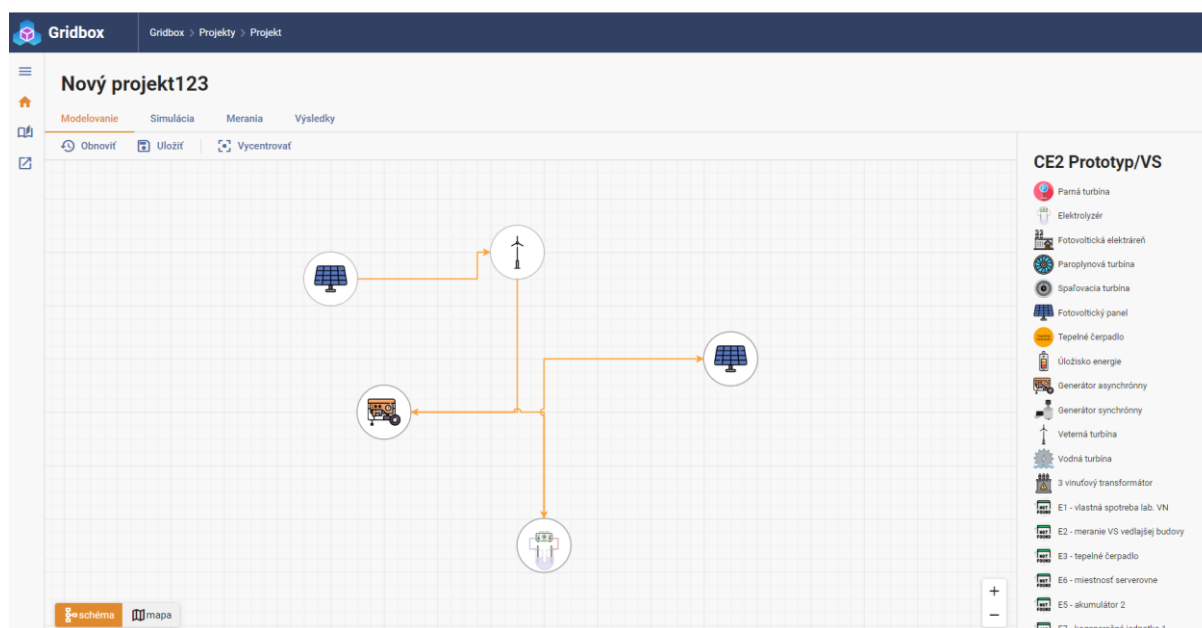
V prípade webového rozhrania MicroGridBoxu®, API definuje a poskytuje sadu služieb a funkcií, ktoré sú dostupné pre používateľov a iné systémy. Tieto služby môžu zahŕňať získavanie a spracovanie dát, vykonávanie optimalizačných algoritmov, správu používateľov a práva prístupu, a ďalšie relevantné operácie pre riadenie mikrogridov.

API teda funguje ako rozhranie medzi webovým rozhraním MicroGridBoxu® a jeho backendovými systémami. Poskytuje definície a štruktúry, cez ktoré môže webové rozhranie pristupovať k potrebným dátam a funkciám, a tým umožňuje efektívnu komunikáciu a interakciu s celým integrovaným riešením MicroGridBox®.

API pre webové rozhranie je vygenerované v technológiách .NET Microsoft a PostgreSQL.



Obrázok 17 Úvodná stránka webového rozhrania



Obrázok 18 Ilustračný príklad schematickej reprezentácie mikrogridu

Gridbox
Gridbox > Projekty > Projekt > Simulácia

☰
🏠
📖
📄

Nový projekt123

Modelovanie
Simulácia
Merania
Výsledky

Základné štatistické jednotky

Odberový diagram

	Deň	Rok
Minimálna hodnota	198 kW	310 kW
Priemerná hodnota	190 kW	225 kW
Maximálna hodnota	117 kW	21 kW

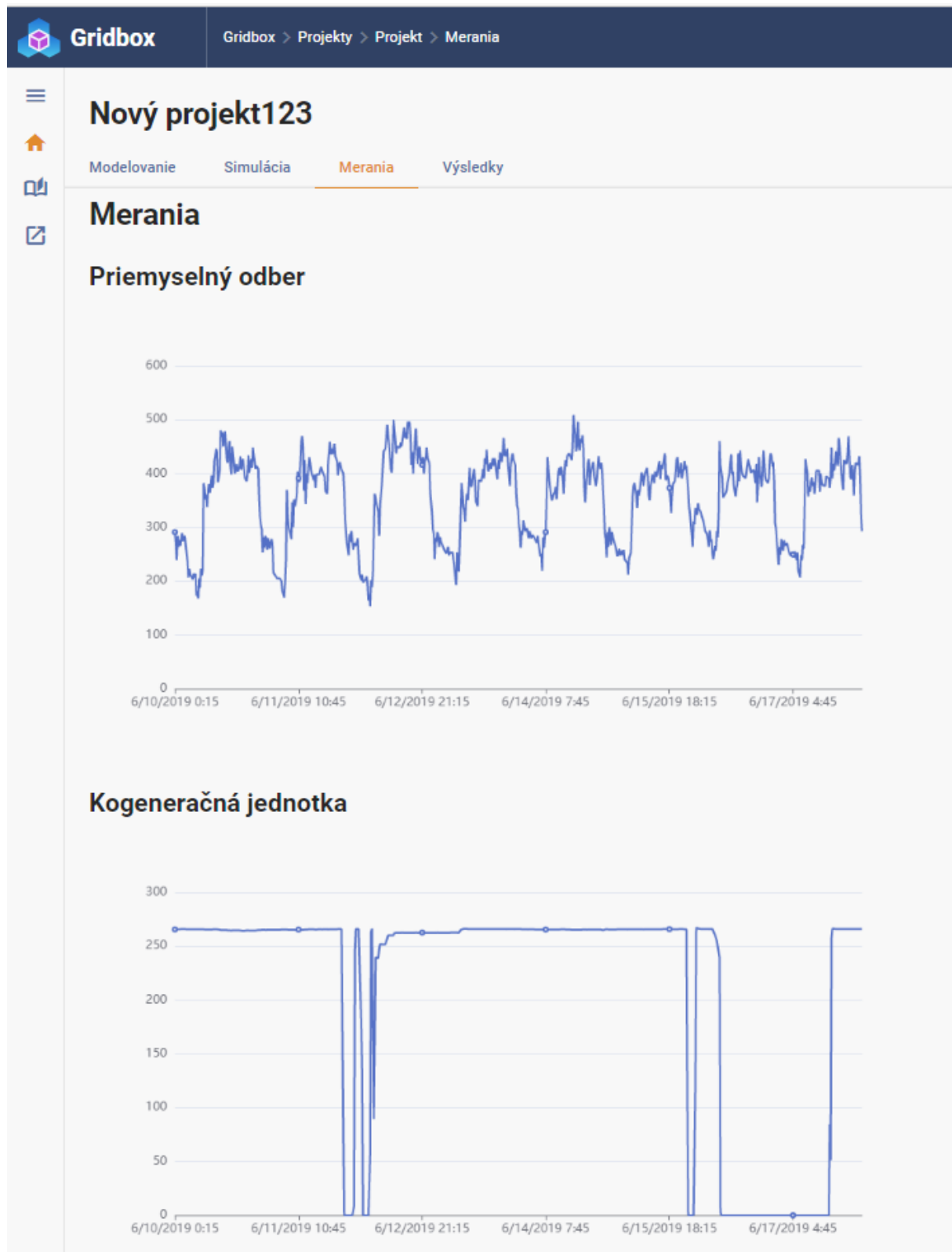
Kogeneračná jednotka

	Deň	Rok
Minimálna hodnota	403 kW	30 kW
Priemerná hodnota	175 kW	195 kW
Maximálna hodnota	476 kW	447 kW

Fotovoltická elektrárň

	Deň	Rok
Minimálna hodnota	264 kW	430 kW
Priemerná hodnota	49 kW	26 kW

Obrázok 19 Ilustračný príklad obsahu modulu Simulácia



Obrázok 20 Ilustračný príklad obsahu modulu Merania

The screenshot shows the Gridbox web application interface. At the top, there is a dark blue header with the Gridbox logo and a breadcrumb trail: 'Gridbox > Projekty > Projekt > Výsledky'. Below the header, a navigation menu on the left contains icons for home, list, and edit. The main content area is titled 'Nový projekt123' and has four tabs: 'Modelovanie', 'Simulácia', 'Merania', and 'Výsledky' (which is active). Below the tabs, there is a sub-header 'Výsledky bilancie po optimalizácii mikrogridu za jeden rok'. A table displays the following data:

Minimálna sumárna štvrthodinová výroba	66	kW
Priemerná sumárna štvrthodinová výroba	178	kW
Maximálna sumárna štvrthodinová výroba	5	kW
Objem elektriny vyrobenej kogeneračnou jednotkou	278	MWh
Objem elektriny vyrobenej fotovoltaickou elektrárnou	21	MWh
Objem elektriny vyrobenej veternou elektrárnou	119	MWh
Sumárny objem elektriny vyrobenej výrobnými jednotkami	82	MWh
Priemerná denná výroba kogeneračnej jednotky	223	MWh
Priemerná denná výroba fotovoltaickej elektrárne	193	MWh
Priemerná denná výroba veternej elektrárne	216	MWh
Celková suma odberu	140	MWh
Celková suma výroby	172	MWh
Rozdiel medzi odberom a výrobou	112	MWh

Obrázok 21 Ilustračný príklad obsahu modulu Výsledky

6.7 Záver z overovania funkčnosti

Overenie funkčnosti integrovaného riešenia MicroGridBox® predstavuje dôležitý proces vývoja softvéru, ktorý zohľadňuje testovanie funkčnosti, výkonu, použiteľnosti, bezpečnosti a kompatibility. Toto overovanie poskytuje záruku kvality, spoľahlivosti a efektívnosti tohto integrovaného riešenia, ktoré slúži na bilančnú optimalizáciu mikrogridov.

Testovanie funkčnosti MicroGridBoxu® zahŕňalo overenie správneho fungovania všetkých jeho hlavných funkcií a procesov. Boli testované scenáre riadenia energie, správne načítanie a spracovanie dát ako aj interoperabilita s ďalšími systémami v mikrogride.

Testovanie výkonu MicroGridBoxu® sa zameriavalo na meranie a hodnotenie jeho výkonnosti pri spracovávaní dát a vykonávaní výpočtových operácií. Boli testované rôzne zaťažovacie scenáre s cieľom identifikovať potenciálne slabé miesta a optimalizovať výkon riešenia.

Testovanie použiteľnosti sa zaoberalo skúmaním používateľského rozhrania MicroGridBoxu®, jeho intuitívnosti, jednoduchosti ovládania a zrozumiteľnosti pre používateľov. Boli vykonané testy použiteľnosti, používateľské prieskumy a získaná spätná väzba od používateľov s cieľom zlepšiť používateľskú skúsenosť.

Testovanie bezpečnosti MicroGridBoxu® sa zameriavalo na identifikáciu potenciálnych bezpečnostných hrozieb a slabín v systéme. Boli vykonané penetračné testy, testy zraniteľnosti a iné bezpečnostné skenovania s cieľom zabezpečiť, že integrované riešenie je odolné voči možným útokom.

Overovanie kompatibility MicroGridBoxu® sa zaoberalo testovaním jeho kompatibility s rôznymi prostrediami, ako sú rôzne operačné systémy, prehliadače, hardvér a ďalšie súvisiace systémy. Bolo testované, či MicroGridBox® správne funguje a je kompatibilný s rôznymi technologickými prostrediami.

Webové rozhranie MicroGridBoxu® je dôležitou súčasťou tohto integrovaného riešenia. Implementácia webového rozhrania MicroGridBoxu® v technológii Angular umožňuje používateľom prístup k rôznym funkcionalitám a možnostiam MicroGridBoxu® prostredníctvom intuitívneho a interaktívneho používateľského rozhrania.

API pre webové rozhranie MicroGridBoxu® je implementované v technológii .NET Microsoft, čo umožňuje efektívnu komunikáciu medzi webovým rozhraním a backendom MicroGridBoxu®. Toto API poskytuje rôzne služby a funkcie pre správu a riadenie mikrogridov, vrátane získavania a spracovania dát, vykonávania optimalizačných algoritmov a komunikácie s ďalšími systémami.

Databázový systém PostgreSQL je využívaný na ukladanie dát a zabezpečenie ich spoľahlivého uchovávaní. PostgreSQL poskytuje robustné a škálovateľné riešenie pre správu a manipuláciu s dátami v rámci MicroGridBoxu®.

Tieto technológie, ako je Angular, .NET Microsoft a PostgreSQL, zabezpečujú spoľahlivú a výkonnú implementáciu webového rozhrania a API pre MicroGridBoxu®. Ich použitie prispieva k vysokému výkonu, bezpečnosti a kompatibilite tohto integrovaného riešenia v rôznych prostrediach a s rôznymi technologickými požiadavkami.

V závere overenia funkčnosti integrovaného riešenia MicroGridBoxu® možno konštatovať, že bolo úspešne splnené testovanie funkčnosti, výkonu, použiteľnosti, bezpečnosti a kompatibility. Toto overenie poskytlo dôveru v kvalitu a spoľahlivosť MicroGridBoxu® ako bilančného optimalizačného riešenia pre mikrogridy. Avšak, vzhľadom na neustále meniace sa technologické prostredie a požiadavky používateľov, je možné ďalej rozvíjať a vylepšovať toto integrované riešenie s cieľom poskytovať ešte lepšiu funkčnosť a efektívne riadenie mikrogridov.

7 ZÁVER

V prvej časti dokumentu analyzujeme publikovanú prácu tímu vedeckých pracovníkov (P. Bauer a kol.) v rámci úlohy č. 18, ktorej zadávateľom bola Medzinárodná agentúra pre energiu – fotovoltaický energetický program, v ktorej sa zaoberali elektrickými systémami mimo siete a elektrickými systémami na okraji siete, ktoré zahŕňajú fotovoltaiku. V práci je riešená úloha porovnania softvérových nástrojov na návrh a vývoj samostatných mikrosietí, pričom sa sústredili na identifikáciu hlavných výhod, slabých stránok a existujúcich nedostatkov softvérových nástrojov na základe definovaných kritérií porovnania (ako sú funkčnosť, použiteľnosť, podporovateľnosť, efektívnosť, optimalizovaný návrh systému atď.). Pre prácu boli stanovené štyri nasledujúce etapy - Výber softvérových nástrojov a definovanie kritérií na porovnanie týchto nástrojov, Prípadové štúdie samostatných mikrosietí s meraniami a obchodnými údajmi, Simulácie a analýza výsledkov, Hodnotenie softvérových nástrojov a odporúčania.

Definujú prehľad charakteristík/funkcií všeobecne používaných softvérových nástrojov na simuláciu a/alebo optimalizáciu mikrosietí. So všetkých analyzovaných softvérových nástrojov podrobne popisujú aplikácie HOMER Pro7 a iHOGA PRO+. Informujú o prípadových štúdiách samostatných mikrosietí v lokalitách v rámci krajín Austrália, Holandsko a USA. Na záver hodnotia výhody a nevýhody týchto softvérov, pričom projektový softvérový výstup MicroGridBox® portál porovnáваме v rámci definovaných kritérií.

V časti batérovej kalkulačky sa zaoberáme používaním výpočtu optimalizácie metódou mriežkového vyhľadávania parametrov. Opisujeme vhodné algoritmy náhodného vyhľadávania, proces simulovaného žihania a metódu gradientného zostupu. Taktiež popisujeme optimalizačný algoritmus inšpirovaný správaním roja častíc - Particle Swarm Optimization (PSO).

MicroGridBox® portál je nástroj na modelovanie a simulácie elektrických schém. V tejto časti dokumentu popisujeme možnosti modelovania, nástroje podporujúce tvorbu a úpravu grafických prvkov a vytváranie prepojení medzi nimi. Nástroj tiež umožňuje ukladanie a správu projektov týchto modelov pre ich budúce použitie a úpravu. V kontexte grafickej reprezentácie modelu mikrogridu je nástroj schopný zobrazit' aj ďalšie informácie o jednotlivých zariadeniach. Nástroj zabezpečuje zobrazenie digitálneho modelu siete na záložkách modelovanie, simulácie, merania a výsledky. Pri implementácii sme využili technológie a frameworky, ktoré podrobne opisujeme (Angular, Sféra UI KID, X6 JavaScript Diagramming Library | AntV, Rest API, Microsoft .NET core, PostgreSQL).

V kapitole Testovanie prototypu vo vzťahu k reálnej mikrosieti popisujeme plánovanie, návrh a implementáciu vykonaných testov. Ďalej popisujeme hodnotenie a vyhodnotenie výsledkov testov, a na základe analýzy opravu postupov a opätovné pretestovanie. Testovanie prototypu prebehlo na reálnych nameraných a zozbieraných dátach.

V záverečnej časti overenia funkčnosti integrovaného riešenia MicroGridBox® konštatujeme, že bolo úspešne splnené testovanie funkčnosti, výkonu, použiteľnosti, bezpečnosti a kompatibility. Toto overenie poskytlo dôveru v kvalitu a spoľahlivosť MicroGridBox® ako bilančného optimalizačného riešenia pre mikrogridy. Avšak, vzhľadom na neustále meniace sa technologické prostredie a požiadavky používateľov, je možné ďalej rozvíjať a vylepšovať toto integrované riešenie s cieľom poskytnúť ešte lepšiu funkčnosť a efektívne riadenie mikrogridov.

8 ZOZNAM OBRÁZKOV

Obrázok 1	Grafické používateľské rozhranie (GUI) softvéru Homer Pro zobrazujúce domovskú kartu, kde je možné vidieť architektúru simulovaného systému a návrhy.....	8
Obrázok 2	Grafické rozhranie softvéru iHOGA PRO+ zobrazujúce rôzne záložky a architektúru simulovaného systému	9
Obrázok 3	Ilustrácia architektúry z obrázka 3 (t. j. prípad 1) viditeľná v (a) HOMER Pro a (b) iHOGA PRO+	14
Obrázok 4	Solárny profil mesta Delft na rok 2021. Údaje sú prevzaté z programu iHOGA Pro a použité v programe HOMER Pro (snímka programu HOMER Pro)	16
Obrázok 5	Ilustrácia architektúry z Obrázka 5 (t. j. prípad 2) viditeľná v (a) HOMER Pro a (b) iHOGA PRO+	17
Obrázok 6	Funkčné oblasti časti pre modelovanie a vizualizáciu.	29
Obrázok 7	Vytvorenie nového projektu	30
Obrázok 8	Manažment projektov.....	30
Obrázok 9	Knižnica prvkov	31
Obrázok 10	Detail vybranej knižnice prvkov.....	32
Obrázok 11	Úprava existujúceho prvku z knižnice.....	33
Obrázok 12	Zobrazenie mapovej verzie podkladu pod zariadeniami	34
Obrázok 13	Panel s parametrami zariadenia	35
Obrázok 14	Porovnanie diagramu spotreby a sumárnej výroby v mikrogride.....	46
Obrázok 15	Grafické porovnanie výrobných diagramov obnoviteľných zdrojov energie v mikrogride	47
Obrázok 16	Príklad obmedzujúcich podmienok pred spustením optimalizácie	47
Obrázok 17	Úvodná stránka webového rozhrania	52
Obrázok 18	Ilustračný príklad schematickej reprezentácie mikrogridu	52
Obrázok 19	Ilustračný príklad obsahu modulu Simulácia	53
Obrázok 20	Ilustračný príklad obsahu modulu Merania	54
Obrázok 21	Ilustračný príklad obsahu modulu Výsledky.....	55

9 ZOZNAM TABULIEK

Tabuľka 1	Prehľad charakteristík/funkcií všeobecne používaných softvérových nástrojov na simuláciu a/alebo optimalizáciu mikrosietí	7
Tabuľka 2	Vybrané kritériá pre porovnanie softvérových nástrojov ,	10
Tabuľka 3	Vstupné požiadavky softvérových nástrojov.	12
Tabuľka 4	Špecifikácie rôznych komponentov použitých na obrázku 3 a vstupy pre softvérové nástroje	14
Tabuľka 5	Porovnanie optimálnych riešení získaných z programu HOMER Pro a iHOGA PRO+ pre prípad 1	15
Tabuľka 6	Špecifikácie rôznych komponentov použitých na obrázku 5 a vstupy pre softvérové nástroje	16
Tabuľka 7	Porovnanie optimálnych riešení získaných z programu HOMER Pro a iHOGA PRO+ pre prípad 2	17
Tabuľka 8	Špecifikácie rôznych komponentov použitých na Obrázku 7 a vstupy pre softvérové nástroje	18
Tabuľka 9	Porovnanie optimálnych riešení získaných z programu HOMER Pro a iHOGA PRO+ pre prípad 3	19
Tabuľka 10	Porovnanie programov HOMER Pro a iHOGA PRO+ a MicroGridBox® portálu na základe kritérií uvedených v tabuľke 2.	20
Tabuľka 11	Obmedzenia inštalovaného výkonu pre výskumnú lokalitu Trnávka.....	44
Tabuľka 12	Príklad tabuľkového prostredia prototypu.....	44
Tabuľka 13	Príklad tabuľkového prostredia prototypu.....	45
Tabuľka 14	Výstupy inštalovaného výkonu jednotlivých zdrojov elektrickej energie po optimalizácii..	45