

BUDAPESTI MŰSZAKI ÉS GAZDASÁGTUDOMÁNYI EGYETEM
TERMÉSZETTUDOMÁNYI KAR
NUKLEÁRIS TECHNIKAI INTÉZET
ATOMENERGETIKA TANSZÉK

FEKETE DEZSŐ DOMONKOS

5. FÉLÉVES ENERGETIKAI MÉRNÖK BSC HALLGATÓ
TUDOMÁNYOS DIÁKKÖRI KONFERENCIA
DOLGOZAT

Kapcsolt energiarendszerek elemzése egy trigenerációs kis moduláris reaktor alkalmazásához

Témavezetők:

Biró Bence
PhD hallgató

Prof. Dr. Aszódi Attila
Egyetemi tanár

Budapest, 2023.

TARTALOMJEGYZÉK

1. Bevezetés	2
2. Előzmények – A modellek áttekintése.....	7
2.1. A PyPSA	7
2.2. Calliope	9
2.3. NEMO	10
2.4. Switch.....	11
2.5. URBS	12
4. Eredmények bemutatása	23
5. Összefoglalás és Eredmények értékelése.....	25
6. Felhasznált források.....	27
7. Mellékletek	32
7.1. A vizsgált erőmű jellemzők	32
7.2. A Pythonban megírt kód	34

Absztrakt

A globális felmelegedés megfékezése céljából a világ energiatermelésének karbonsemleges technológiákra kell áttérnie. A villamosenergia-, a hőenergia- és a hidrogén-termelés egyik hatékony megoldása lehet a trigenerációs kis moduláris reaktor, ami egyszerre képes villamos energia, hő és hidrogén előállítására. Ezen típusú erőművek azonban még csak kísérleti vagy demonstrációs fázisban vannak, így a jövőbeli energiapiacra betöltött szerepük modellezése elengedhetetlen.

Kutatásomban megvizsgálom és összevetem a piacon elérhető energiapiac-modellező programokat, melyek megfelelően tudják mind az energiapiacokat, mind a kis moduláris reaktorokat modellezni. Az irodalomkutatásom alapján a PyPSA program bizonyult a legmegfelelőbbnek, azonban ez a program sem elégíti ki minden modellezési igényt, így a kutatásom során fejleszteni fogom a PyPSA kódját a trigenerációs reaktorok legpontosabb modellezése érdekében. A szükséges módosítások végrehajtása után megépíttem Magyarország és a szomszédos országok 2030-as villamosenergia-rendszerét, valamint Magyarország jövőbeli távhő és hidrogén igényének megtermelését is modellezem. A dolgozat eredményeként létrejött modell megerősíti, hogy a trigenerációs kis moduláris reaktorok fejlesztése a hagyományos atomerőművek és a megújuló energiaforrások mellett egy fenntartható villamosenergia-, hő- és hidrogén termeléshez vezethet Magyarországon.

Köszönetnyilvánítás

Ezúton szeretnék köszönetet mondani a témavezetőimnek, Bíró Bencének és Prof. Dr. Aszódi Attilának a rengeteg segítségért, az utolsó pillanatos konzultációkért, munkám lelkiismeretes ellenőrzéséért, illetve a bizalomért és sok türelemért a dolgozat elkészítése során.

Emellett szeretném megköszönni Várkonyi Zsófiának az 1. ábra elkészítésében nyújtott segítségét.

1. BEVEZETÉS

Napjainkban a globális felmelegedéssel, és az egyre inkább megcsappanó kőolaj, földgáz és szénkészletekkel megnövekedett az igény a karbonsemleges energiaforrásokra történő átállásra. Ezek többek között megújuló és nukleáris technológiákat jelentenek. A nukleáris és megújuló kapacitások együttműködésének elemzése nagyon fontos, hiszen így kaphatunk képet a jövő energiapiacáról. Ezen elemzés eszközei lehetnek a nagy (legalább órás felbontású) energiapiaci modellek.

2022-ben a Magyarországon megtermelt villamos energia 44,7%-a nukleáris és 20,3%-a megújuló forrásból érkezett, ezzel az országban megtermelt villamos energia 65%-a származott karbonsemleges forrásból [1]. A távhő esetében a termelt hőenergia 12,7%-a biomasszából, 10%-a geotermikus energiából, 0,1%-a egyéb megújulókból, 1,1%-a pedig nukleáris forrásból érkezett, ezzel a távhő 23,9%-át termelték karbonsemleges technológiát alkalmazó létesítményben a 2021-es évben Magyarországon [2]. A Nemzeti Energiastratégia 2030-ra azonban olyan távhőrendszert tervez, amelyben a hőenergia termelés vagy legalább 50%-ban megújuló energia vagy 50%-ban hulladékhő, vagy 75%-ban kapcsolt energiatermelésből származik, vagy 50%-ban ilyen energiaforrások kombinációjának a felhasználásából. A hidrogén technológia még ipari felhasználásban sok területen csak kutatási fázisban van, így még Magyarországon sem alakult ki 2023-ban komoly hidrogénpiac, az ország karbonsemleges hidrogéntermelése elhanyagolható. Magyarország Nemzeti Hidrogénstratégiája [3] 2030-ra 36 000 t/év mennyiségre becsüli az ország hidrogénigényét, melynek mintegy 44,4%-a lesz teljesen karbonsemleges forrásból származó hidrogén a dokumentum szerint.

Az energiaszektor jelenleg igen nagy CO₂ kibocsátó, mellyel jelentősen hozzájárul a globális felmelegedés felgyorsulásához. Nem is beszélve a konvencionális erőművek (főként a szén alapú erőművek) kén-oxid kibocsátásáról, melyek a savas esők létrejöttét is erősen segítik [4]. Ezen tények miatt van szükség a fosszilis tüzelésű erőművek kiváltásra.

A villamos energia tekintetében Európában alapvetően kettő különböző módon próbálják megoldani ezt a problémát. Több, a nukleáris energiát ellenző országban (pl. Németország, Ausztria, Spanyolország) a megújuló energiaforrások telepítését szorgalmazzák

[5]. Ezek közül az utóbbi években a legnagyobb fellendülés a fotovoltaiikus villamos-energia-termelésben volt tapasztalható [6]. Nem elhanyagolhatóak még a szárazföldi szél-erőművek, emellett a tengerparti országokban a tengeri szél-erőművek telepítései sem. Az északi országokban a megtermelt villamos energia jelentős része vízenergiából származik, azonban ehhez speciális földrajzi adottságokra van szükség. Emellett nem túl nagy részarányban a megújuló energiamixekben a hullám és árapály és biomassza erőművek is megjelennek, a geotermikus energia felhasználása mellett. Néhány ország a megújulókon túl a nukleáris technológiákat is a karbonsemlegesség elérésének egyik megoldásának tekinti. Franciaország villamosenergia-termelésének jelentős része származik atomerőművekből [7] a lengyelek szél-erőműveiket klasszikus „nagy” atomerőművekkel és kis moduláris reaktorokkal tervezik kiváltani [8] emellett Magyarországon is zajlanak az új paksi blokkok építéséhez [9] és a Paksi Atomerőmű üzemidő hosszabbításához kapcsolódó munkálatok [10], illetve Finnországban is adtak át nemrégiben atomreaktort [11]

A távhő vonatkozásában [12] nem áll rendelkezésre ennyi karbonsemleges technológia. Alapvetően egy jó megoldást biztosít a távhőtermelésre a geotermikus energia, emellett a biomassza és a biogáz is opció lehet. A hétköznapiakból ismert napkollektorok akár használhatóak ipari méretekben is, emellett ipari hőszivattyúkkal is találkozhatunk, melyek villamos energiából képesek hőt előállítani közvetlen CO₂ kibocsátás nélkül. Ezen technológiák mellett még a nukleáris erőművekben is termelhető hő, például Paksot és környékét az atomerőmű látja el a turbina megcsapolásából származó távhővel.

A hidrogén előállítása [13] jelenleg nagyrészt fosszilis energiahordozók segítségével történik, az Európai Unióban előállított hidrogén 95%-a fosszilis forrásból származott 2020-ban. Az uniós iránymutatások alapján a cél a zöld hidrogén előállítása, mely megújuló energiaforrásból, elektrolízissel előállított hidrogént jelent. Karbonsemleges azonban még a kék hidrogén, mely a földgáz gőzreformációjával, a CO₂ megkötésével, illetve a nukleáris forrásból elektrolízissel előállított rózsaszín hidrogén, továbbá a földgáz pirolízisével előállított türkiz hidrogén is. Ezek azonban sok hő vagy villamos energiát igénylő folyamatok [14].

Ezen három energiaforrás tetszőleges kombinációban együttesen is előállítható. Közülük talán a legszélesebb körben alkalmazott technológia a kombinált hő- és villamosenergia-

termelés. Ahogy a fentiekben már láttuk a 2030-as Nemzeti Energiastratégia a távhő termelésnek egyik lehetőségeként adja meg a kapcsolt energiatermelést. Erre vonatkozóan vannak EU-s célok is, így a jövő energiapiacában biztosan érdemi szerepe lesz a kapcsolt energiatermelésnek. Hatalmas előnyük, hogy a kapcsolt energiatermelés teljes hatásfoka akár a 80%-os értéket is elérheti. A kapcsolt hő és villamosenergia-termelésben résztvevő erőművek beépített teljesítménye 15 kW és 400 MW között mozoghat, és jelenleg a földgázzal működő erőművek az elterjedtek. Azonban van néhány hátrányuk: az elektromos és hőterhelés szoros összehangolására van szükség, a helyi szén-dioxid terhelés a jelenlegi technológiákkal jelentős, a hálózatra való kapcsolódásuk költséges lehet, továbbá a kapcsolt erőművek beruházási költsége is magas [15].

A kombinált hő és hidrogéntermelés is létező technológia. Az egyik opció a protoncserélő membrános elektrolízis. A folyamat során villamos energia felhasználásával 60-80%-os hatásfokkal hidrogént termel a rendszer, a folyamat során keletkező hőt pedig egy hőcserélőn keresztül a távhőhálózatra tudja a berendezés termelni. A protoncserélő membrános elektrolizálók maximális teljesítménye jelenleg 20 MW, így a technológiával nem feltétlenül oldhatóak meg a távhőtermelés kihívásai [16].

Az alumínium vízzel történő oxidációja is rejteget lehetőségeket a kombinált energiatermelésben. A víz és alumínium reakciója során alumínium oxid, hidrogén és hőenergia keletkezik. Ez lehetőséget biztosít egyrészt a távhő hálózatra történő termelésre, másrészt a folyamatban keletkezik elegendő hő a víz elforrálásához, így egy villamosenergia-termelő körfolyamat is megvalósíthatóvá válik. Ezen folyamatban azonban 1 kg alumínium oxidálása során 17.6 MJ energia keletkezik, így a konstans, nagy teljesítményű villamosenergiatermeléshez nagyon nagy mennyiségű alumíniumra lenne szükség [17].

A szénelgázosító hidrogéntermelő folyamat összekapcsolható a villamosenergia-termeléssel. Az erőmű egy klasszikus gőzkörfolyamattal termel áramot, azonban a gőz ebben az esetben nemcsak a turbinát, hanem a szénelgázosítót is táplálja. Ezen technológia nagy hátránya, hogy jelentős CO₂ kibocsátással jár, azonban itt a teljesítmény nem jelent akkora problémát [18].

Azonban ezek a kapcsolt hő és villamosenergia-termelést leszámítva a gyakorlatban nem széleskörűen alkalmazott eljárások, sőt van amelyik csak koncepció szinten létezik.

A trigenerációs rendszerek a három energiahordozót együttesen képesek előállítani. Ezek vonatkozásában azonban sok kutatás még nem született. S. P. Cicconardi és munkatársai biogázon alapuló rendszerre vizsgáltak már hatásfokokat és költségeket, különböző szcenáriókra. Maximális hő-, hidrogén- és villamosenergia-termelés eseteit vizsgálták. Ekkor 69 % és 72 % közötti teljes hatásfokok jöttek ki, melyek nagyon kedvezőek [19]. Giorgio Locatelli és munkatársai kis moduláris reaktorok vonatkozásában terheléskövető reaktorokat vizsgáltak, melyek villamos energia és hidrogén kapcsolt termelésére voltak alkalmasak, különböző hidrogéntermelő folyamatokkal. A kutatás alapján összességében elmondható, hogy csak akkor fog hidrogént termelni a reaktor, ha a hidrogén ára kilogrammonként minimum 0,4 - 0,5 Euróba kerül és a villamos energia ára relatíve alacsony. Jelenleg azonban a hidrogén ára 5 Euró/kg-nál minden piacon magasabb, így a jelenlegi energiapiacokon gazdaságilag megéri az SMR-ekkel kapcsolt energiatermelést végezni a kutatás alapján [20], [21]. Miika Rämä és munkatársai Franciaországban a hagyományos atomreaktorok kapcsolt hő- és villamosenergia-termelését vizsgálták a rugalmasság vonatkozásában. Három várost elemeztek a kutatásban, amely kutatás eredménye az lett, hogy a nagy teljesítményű francia EPR reaktorok nem feltétlen tudják egy ilyen kapcsolt energiatermelésben a rugalmasságot biztosítani, azonban feltételezéseik alapján a kis moduláris reaktorok esetében ezen vizsgálatnak lenne relevanciája, de ez a mai napig nem készült el [22].

A fentiekben ismertetett rendszerek piaci modellezése azonban szükséges lehet megtérülésük elemzése érdekében, ennek eszköze pedig az energiapiac modellező kódok lehetnek. A modellek az energiaszükséglet és a beépített kapacitások (legyen szó akár erőműről, fűtőműről vagy éppen tároló kapacitásról) ismeretében optimális teljesítményáramlást számítanak az adott rendszerben, és meg is jelenítik annak jellemző paramétereit.

Az energiapiacok modellezése nem újkeletű törekvés. Változatos kutatások vizsgáltak különböző forgatókönyveket az energiapiacra vonatkozóan, optimális megoldásokat keresve. Dogan Keles és munkatársai Németország esetében [23] például különböző szcenáriókat vizsgáltak többek között a CO₂ kibocsátás csökkentésére, az elszaladó fosszilis üzemanyagárakra, illetve az akkor még csak tervben lévő nukleáris erőművek kivezetésére vonatkozóan. Emellett David Shropshire a kis moduláris reaktorok (angolul small modular reactor, röviden SMR) gazdasági megvalósíthatóságát is elemezte már piacmodellekkel [24]. Itt kis és közepes méretű (<700 MW_e) reaktorok európai villamosenergia-

piacba történő integrálását vizsgálta különböző piaci scenáriókban, 45 és 150 Euró/MWh közötti villamos energia árakkal. A [24] -ben bemutatott kutatás eredménye azt mutatja, hogy az SMR-ek versenyképesek lehetnek a jövő villamosenergia-piacán. Jelen munka motivációja is hasonló, azonban az áttekintésnél az SMR-ek szerepét a hő- és hidrogénpiacon is megvizsgáltam a villamosenergia-piac mellett. A dolgozatban áttekintem az energiapiac-modellezésre alkalmas nyílt forráskódú, Python alapú modelleket. Ezek közül kiválasztom a fenti céloknak legmegfelelőbb modellt, majd azon kisebb-nagyobb módosításokat elvégezve elemzem és értékelem a trigenerációs kis moduláris reaktorok szerepét Magyarország és szomszédos országainak 2030-as energiapiacán.

Hő, hidrogén és villamos energia együttes termelését kevés cikkben dolgozták fel, olyan publikációt pedig nem találtam az irodalomkutatásom során, ahol mindezt nukleáris energiaforrással valósították meg. Ezen okból munkámban a kis moduláris reaktorok együttes hő-, villamosenergia- és hidrogén-termelését elemzem, bemutatva, hogy a trigenerációs SMR-ek hozzá tudnak-e járulni a villamosenergia-piacon a terheléskövetéshez, továbbá bemutatom szerepüket a 2030-as energiapiacokon. Ezen elemzés elkészítésére a PyPSA energia-piaci modellező kódot használtam, és alakítottam át az SMR megfelelő modellezése érdekében.

A dolgozat 2. fejezetében bemutatom és összehasonlítom a jelenleg szabadon hozzáférhető energiapiac-modellező programokat, a bemutatott tulajdonságok alapján pedig kiválasztom a számomra legmegfelelőbb programot. A 3. fejezetben a modellezéshez használt adatokat, illetve magának a kódnak a felépítését mutatom be. A 4. fejezetben bemutatom a kutatásom során kapott eredményeimet, az 5. fejezetben pedig összegzem a dolgozatot.

2. ELŐZMÉNYEK – A MODELLEK ÁTTEKINTÉSE

A kutatásom első lépése a piacon fellelhető energiapiac és energiarendszer modellezésre alkalmas szoftverek áttekintése volt. Hans-Kristian Ringkjøb és munkatársai „A review of modelling tools for energy and electricity systems with large shares of variable renewables” című publikációja [25] jelentette az irodalomkutatás alapját. A kutatásban 75 energiarendszer modellezésére alkalmas programot, illetve szoftvert mutatnak be. A cikkben számtalan tulajdonságot feltüntettek az adott modellhez kapcsolódóan, illetve iránymutatást adnak arra is, hogy milyen megfontolások alapján lehet kiválasztani a megfelelő modellt. Elérhetővé tették az egyes szoftverek tulajdonságait összefoglaló Excel táblázatot is. Ebben a táblázatban szűrtem rá a számomra megfelelő tulajdonságokra [25].

Kezdetben nyílt forráskódú, Python alapú, legalább óránkénti felbontásra képes modelleket kerestem. Ezen feltételeknek a következő öt szoftver felelt meg: PyPSA, Calliope, NEMO, Switch, URBS. Megvizsgáltam, hogy mely program mire használható, korábban milyen kutatásokhoz használták.

2.1. A PyPSA

A Python for Power System Analysis (röviden PyPSA) [26] program az energiaáramlás optimalizációjára szolgál. Az optimalizáció lehet lineáris is és nemlineáris is, a szoftver mindkettő kezelésére alkalmas. Összességében pedig az erőművek és tárolók termelésének legkisebb költségű meghatározására alkalmazható, figyelembe véve a különböző biztonsági és technológiai korlátokat. Ezen túl a termelés, tárolás, átvitel és egyéb infrastruktúrákkal kapcsolatos beruházások értékelésére, optimalizációjára is alkalmas. A program alapból fel van készítve többszörösen összekapcsolt AC és DC hálózatok modellezésére, időben változó teljesítményű erőművek (pl. nap, szél, víz) kezelésére, vannak definiálva tárolók, melyekben a veszteségek is kezelhetők. Ezen túl több különböző energiahordozó együttes kezelésére is alkalmas, továbbá a villamosenergia-piacon túl egyéb piacok modellezése, azok összekapcsolása is lehetséges. Többek között kapcsolt hő és villamosenergia-termelés, hőszivattyús rendszerek, elektromos járművek, de akár Fischer-Tropsch eljárások kezelésére is alkalmas. A rendszer optimalizálása a következők figyelembevételével történik:

- a termelés és tárolás optimalizálása, figyelembe véve a termelési, tárolási és átviteli rendszer kapacitásait, idősoros adatait,
- a modell rugalmatlan terhelést feltételez, melyet minden időpillanatban ki kell elégíteni,
- egy optimalizálási problémához tetszőleges számú előre meghatározott kényszer megadható,
- a termelők, tárolók és átviteli hálózat elemeinek tőke és összevont változó költségeket kell átadni,
- az optimalizációhoz folytonos változókat használ a modell, választható solverrel,
- a célfüggvény pedig az optimális rendszer költsége (ezt igyekszik minimalizálni).

A választott modellek közül ez a legszéleskörűbben dokumentált szoftver, több kutatás alapjául is szolgált a PyPSA program.

Ahogy az a fentiekben már felmerült, a program alkalmas a villamos energián túl egyéb energiahordozók és a CO₂ kibocsátással kapcsolatos problémák kezelésére is. Ezt kihasználva egy német kutatásban [23] a megújuló energiaforrások segítségével előállított kémiai energiahordozók importjára vonatkozó optimalizációs problémát oldottak meg a PyPSA programmal. Egy energiaellátási lánc legfontosabb összetevőire vonatkozó beruházások optimalizációját végezték az éves energiaellátás fedezésére. A modellben deklarált komponensek között megtalálhatók voltak az erőművek, a megújuló termelésre vonatkozó történelmi időjárásadatok, az átalakító eljárások, a tárolók, illetve az országok közötti energiaátvitel adatai is. Így ezekkel a villamos energia és kémiai energiahordozók piaca modellezhetővé vált.

A PyPSA alkalmas igen nagy felbontású (akár több ezer csomópontból álló) energiarendszer modellezésre is. A kutatásban [27] azt vizsgálták, hogy a felbontás finomítása, vagy éppen durvábbá tétele milyen hatással lehet az energiapiaci viszonyokra, hogyan változhat ekkor a villamos energia ára. A konklúzióban pedig arra jutottak, hogy a felbontás finomításával, vagyis a valósághoz egyre inkább közelítő modellekkel akár 27%-os különbségek is megjelenhetnek a villamos energia árában.

A modell az optimális energiaáramlást keresi, azonban állhatnak fenn olyan scenáriók is, mely során az optimálisához közeli megoldás például egy beruházás esetében sokkal könnyebben és praktikusabban megvalósítható. A PyPSA modellekkel ennek lehetőségét

is vizsgálták a szárazföldi szél erőművek és a naperőművek helyettesítésére irányuló kutatásban [28].

2.2. Calliope

A Calliope [29] programot olyan energiarendszerek elemzésére tervezték, melyek nagy részarányban tartalmaznak megújuló energiaforrásokat. A szoftver tetszőleges térbeli és időbeli felbontást tesz lehetővé, minden eszköz adott benne az idősoros adatok kezelésére. A kód jól strukturálható, a bevitt adatok jól elválaszthatóak egymástól így a modulás bővítés megvalósítható, emellett az összetett, nagy teljesítményigényű modellek is kiszámíthatóak a szoftverrel a többszörös futásnak köszönhetően.

Ezen modellből készült egy európai szintű verzió is: az Euro-Calliope [30], mellyel szimulálhatóvá válik a teljes európai villamosenergia-rendszer. Itt 3 féle térbeli felbontás érhető el: kontinens szintű, ország szintű (34 országgal), régió szintű (közel 500 régióval).

A csomópontokon megújuló kapacitások építhetők be, és hozzájuk valamilyen tárolókapacitások társíthatók (pl. hidrogén vagy akkumulátor, de akár szivattyús-tározós erőművek is kiépíthetőek). A megtermelt, illetve tárolt energia az összes helyszín energiaigényének fedezésére használható, a helyszíneket korlátlan kapacitású távvezeték köti össze. Ekkor a modell egy olyan lineáris optimalizálási problémát old meg, ahol az optimalizáció célja a rendszer összes költségének minimalizálása.

A szoftverben az energiarendszer felépítése és optimalizálása a következők alapján valósítható meg:

- a megtermelt, tárolt és elhasznált energiamennyiségek idősoros adatai
- az energiatermelő, szállító, transzformáló berendezések jellemző paraméterei (pl. beépített kapacitások, árak),
- a villamosenergia-rendszer elemeinek térbeli elosztása, átviteli hálózat modellje
- különböző kényszerek által létrejövő scenáriók.

A fentiek alapján a modell a legkisebb költségű rendszer optimalizációs számítását futtatja. A program különösen jól fel van készítve az idősoros adatok vizualizációjára.

Alapvetően a PyPSA-tól rosszabbul dokumentált szoftver, azonban több publikációhoz is használták a korábbiakban, melyek közül néhány[31]–[33]2030-as, 2040-es energia-rendszerek modellezésére irányult.

A Calliope energiapiac-modellező szoftver a PyPSA-hoz hasonlóan igen jól dokumentált, azonban elsősorban a villamosenergia-piac modellezésére alkalmas.

2.3. NEMO

A NEMO [34] modellező szoftver a vizsgált programok közül a legrosszabbul dokumentált program volt. Pythonban íródott a kód és könyvtárként telepíthető (ahogyan egyébként ez az előző kettő programról is elmondható). A kódról a fentiekhez hasonló összefoglalókat nem találtam, használatát egyedül egy ausztrál kutatás [34] mutatja be. Ez Ausztrália villamosenergia-rendszerének és villamosenergia-piacának optimalizálását célozta meg. A modellben 100%-ban a víz- és szélenergia-felhasználására koncentráltak, kémiai energiatárolók nélkül. A megújulók termelését a történelmi időjárás adatok alapján határozták meg.

A modellben a következők figyelembevételével optimalizálták a rendszert:

- Energiaköltségek: az egyes technológiák kiépítése, állandó és változó költségeik, a költségekhez definiált 5-10%-os kamatlábak,
- az egyes régiók erőművei, a régiók összekötését biztosító átviteli hálózat és a régiók közötti energiapiac sajátosságai,
- az erőművek termelésének óránkénti bontása.

Az bemeneti adatok alapján a többi modellhez hasonlóan itt is költségminimalizálással optimalizálnak. Lineáris és nem lineáris egyenletek is megjelentek az optimalizációs folyamatban.

A fentiek alapján eredményként a következő eredményeket kapták meg:

- A piacon kialakuló villamos energia ára,
- az energiapiac felépítése, a különböző megújulók energiamixben való részaránya, illetve területi megoszlása.

A kutatásban modelleztek konvencionális erőműveket is, a CO₂ kvóták figyelembevételével. A végkonklúzió pedig az lett, hogy a 100% megújuló rendszer költségoptimalizáció szempontjából jobb lesz, mint a nem karbonsemleges eset.

A program alapvetően erre a fent ismertetett problémára lett megírva, és bár valószínűleg kiegészíthető lenne, az előző kettő szoftvernél lényegesen kisebb képességekkel rendelkezik.

2.4. Switch

A Switch [35], [36] egy nyílt forráskódú, villamosenergia-rendszer modellező program, melynek elsődleges célja a jövő energiarendszerének tanulmányozása és elemzése. Főként olyan rendszerek modellezésére van felkészítve, melyben nagy a megújuló részaránya és tárolókapacitásokat is tartalmaz. A szoftver az optimalizálással ebben az esetben is a legkisebb költségű eseteket keresi. Az optimális rendszer keresése során figyelembe vehetővé válik a különböző célok/vállalások teljesíthetőségének vizsgálata is: CO₂ kvóták és megújuló célok, a megbízható villamosenergia-rendszer fenntartása.

A programban lehetőség van különböző elemzési esetek futtatására, mellyel a dekarbonizációs és egyéb modern technológiák költségei vizsgálhatók. A Switch olyan mögöttes matematikai modellt használ, mely több beruházási időszakot és azok időbeli sorrendjét is figyelembe képes venni. Ezzel pedig lehetővé válik a megújuló energiaforrásokat nagy részarányban tartalmazó rendszerek hosszútávú elemzése is.

A modell moduláris elemekből épül fel, melyek közül választhatunk előre megírottakat, illetve magunk is adhatunk hozzá ilyen elemeket, ezzel egészen személyre szabható modellt kapva.

A Switch program az optimalizációt a következők figyelembevételével végzi:

- a rendszerben lévő erőművek paraméterei,
- a villamosenergia-fogyasztásának és -termelésének idősoros adatai (opcionálisan tartalékok is megadhatóak),
- az energiatermelés gazdasági adatai (pl. üzemanyag és CO₂ kvóták árai)
- az átviteli hálózat, illetve a kibocsátás korlátjai.

Ezen bemeneti adatokkal pedig egy minimális költségű rendszert számít ki a modell lineáris optimalizációval. Az eredményekben megjelenhetnek az erőművek termelésének idősoros adatai, a tárolók aktuális töltöttségei, továbbá az elemzési esetekkel a program képes javaslatot adni az optimálisabb rendszer fejlesztési lehetőségeire akár átviteli, akár

erőművi szinten. A megújuló energiatermelés modellezésére több kutatásban [37], [38] is használták a szoftvert.

2.5. URBS

Az urbs [39] modellt azért hozták létre, hogy az energiaszektorban a beruházásokat, fejlesztéseket optimalizálja. Az URBS az előző programokhoz hasonlóan lineáris optimalizációs modell, mely több terméket tartalmazó energiapiacokat képes figyelembe venni, és ezen piacokon keresi a minimális költségű rendszereket. A modell óránkénti felbontással képes dolgozni, és Python Pandas alapú, így alapvetően egy jól bővíthető programról van szó. Az URBS funkcióit tekintve leginkább a PyPSA programhoz hasonlít, az optimalizációt a következők figyelembevételével végzi:

- az átviteli hálózat paraméterei,
- az erőművek és tárolókapacitások jellemzői,
- az időjárási és fogyasztási adatok,
- rendelkezésre álló területek,
- különböző kényszerek (például CO₂ kibocsátási limitek, maximális költségkeret).

A bemenő adatok alapján lineáris optimalizációval számítja ki a rendszer paramétereit választható solverrel.

Az URBS szoftver alkalmazása igen széleskörben lehetséges, és nem csak klasszikus energiapiaci problémák megoldására használható [40] [41]

Az általános áttekintésen túl specifikus funkciókat is kerestem a különböző szoftverekhez, majd a végső döntést is az alapján hoztam meg. A villamosenergia-piac modellezésében széleskörben használt, az Energy Exemplar PLEXOS elnevezésű szoftverjében [42] is elérhető funkciókat kerestük. Ezen funkciók az A mellékletben találhatóak meg.

Az egyes szoftverekben megtalálható funkciókat egy táblázatban foglaltam össze. Pirossal jelöltem azon funkciókat, melyek egyáltalán nem találhatóak meg, narancssal azokat, melyekhez hasonló funkció van implementálva, de relatíve nehezen átírható. Sárga jelölést kapott a programban csak részben megtalálható, azonban kis munkával átírható funkció és zöldet, melyek bármilyen átdolgozás nélkül, azonnal használhatók. Ezen összefoglaló az 1. táblázatban látható.

1. táblázat A különböző modellekben az erőművek tulajdonságainak modellezhetősége

Tulajdonságok rövidítése	PLEXOS	PYPSA	CALLIOPE	Nemo	Switch	URBS
BT						
MSS						
η						
FS						
LS						
MFS						
MLS						
MFI						
MÁI						
IP						
LP						
ÜK						
VÜéKK						
ÁÜéKK						
ÜÁR						
MIKÜÁK						
ÜÁÁI						
ÜÁMiI						
ÜÁMaI						
NVKA						
JÁI						
JMiI						
JMaI						

Az atomerőmű specifikus tulajdonságok egyik programban sem találhatók meg, így ezen tulajdonságok alapján a dolgozatban végül nem végeztünk vizsgálatot. Az 1. táblázat adataiban látható, hogy a NEMO modell nem alkalmas a legtöbb funkció elvégzésére, így bebizonyosodott, amit már az általános összefoglaló alapján is sejteni lehetett: a vizsgált modellek közül messze ez a legszerényebb képességű. Egzaktul az indítási profilok sem voltak implementálhatóak egyik modellben sem, így ezek sem szerepelnek a dolgozatban. A bruttó teljesítményt, a minimális stabil szintet, a hatásfokot, a minimális futási és állási időket, továbbá a költségeket mindenképpen szerettem volna elhelyezni a modellben, ezért az URBS és Calliope programokat is kizártam a további elemzésből.

A Switch és a PyPSA programok igen hasonló funkciókkal rendelkeznek. A PyPSA szoftverben azonban a dokumentációk alapján lényegesen könnyebben modellezhető a villa-

mos energián túl a távhő és a hidrogén termelése, illetve piaca. Emellett az egyetlen hátránya a PyPSA-nak a Switch-csel szemben a költségek kezelése. A PyPSA-ban üzemanyag és karbantartási költség helyett határköltséget adhatunk meg. Ez azonban egy igen rövid kóddal felírható az üzemanyag és egyéb változó költségek összegére, mellyel a hiányzó funkció hamar kiegészíthető.

A PyPSA rendelkezik tehát azon funkciók közül a legtöbbel, melyek alapján el szeretnénk végezni a kis moduláris reaktorok termelésének vizsgálatát. Emellett talán ez a program rendelkezett a legszéleskörűbb dokumentációval, számtalan mintakóddal, ráadásul a vizsgált szoftverek közül ez szerepelt a legtöbb publikációban, így én is ezzel dolgoztam tovább.

3. ADATOK ÉS MÓDSZEREK

Az energiapiac modellt 7 országra építettem fel: Magyarország, Ausztria, Szlovénia, Horvátország, Szerbia, Románia és Szlovákia. Magyarország szomszédjaiban csak a villamosenergia-termelését és -fogyasztását modelleztem, Magyarországon azonban a villamos energia mellett a távhő és hidrogén termelését és fogyasztását is vizsgáltam. Ennek érdekében először adatokat kellett gyűjtenem a különböző erőművi technológiákra, a megújulók termelésére, illetve a fogyasztási adatokra vonatkozóan.

Munkám során elsődlegesen az energiapiacokon végbemenő folyamatokra voltam kíváncsi, így az átviteli hálózat további elemeit (például távvezetéseket, transzformátorokat, alállomásokat) nem építettem bele a modellbe. A rajtuk fellépő veszteségekkel, ezen bevezetések költségeivel nem számoltam.

3.1. A villamosenergia-piac modellezéséhez szükséges adatok

A korábban említett hét ország atom-, lignit-, földgáz-, olaj-, víz-, szél-, nap-, geotermikus-, biomassza-, és biogázerőművi kapacitásait [43]-ben meghatározott adatok alapján adtam át modellnek. Ugyanezen publikáció alapján a villamosenergia-fogyasztására, illetve a szél-, nap- és vízerőművek termelésére vonatkozó adatokat órás bontásban adtam meg. Az adatok 1980 és 2019 közötti statisztikák alapján készültek, melyek közül a vizsgálatot a 2019-es év adataival végeztem. Az egyes országok 2030-as vállalásai az energetikai szektorban publikusak, azonban a megújulók termelését, továbbá a fogyasztási adatok órás bontású becslései nem állnak még rendelkezésre 2030-ra vonatkozóan, ezért választottam a legfrissebb rendelkezésre álló adatokat.

A modellben implementált erőművek és technológiák tulajdonságai közül a minimum stabil szintet, a fajlagos hőfogyasztást (hatásfok formájában), változó üzemeltetési és karbantartási költséget, üzemanyag árat, minimum futási időt és minimum állási időt építettem bele a kódba. A [43] dolgozat a blokkok fel és leterhelési sebességére vonatkozóan is közöl adatokat. Ezeket már alaphelyzetben is képes kezelni a PyPSA, azonban órás felbontásban ezen értékekkel nincs értelme számolni, ugyanis egy órán belül bármely erőművi technológia esetén megvalósítható a 0-ról 100%-ra történő felszabályozás, illetve a 100%-ról 0%-ra történő leszabályozás. Negyedórás, illetve finomabb felbontásban azonban ennek kezelése is releváns lehet.

Az összes vizsgált országban definiáltunk villamosenergia-tárolókat. Magyarországon 100 MW teljesítményű 2 órás, azaz 200 MWh-nyi kapacitású akkumulátort, a környező országokban pedig a [43] dolgozat alapján víztározókkal összekapcsolt szivattyús-tározós erőműveket definiáltam a modellben.

Az országok közötti határkeresztező kapacitásokat is átadtam a modellnek. A PyPSA azonban jelenleg nem képes 2 ország között különböző határkeresztező kapacitás kezelésére, így a [43] alapján megadott határkeresztező kapacitásokat a 2. táblázat szerint módosítottam. Ez azonban nem jelentett túl nagy módosítást, csak az Ausztria – Magyarország és Magyarország - Románia közötti kapacitásokat kellett megváltoztatni.

2. táblázat A határkeresztező kapacitások

Honnan	Hová	Eredeti		Módosított	
		Export [MW]	Export [MW]	Import [MW]	Import [MW]
AT	HU	1200	800	800	800
AT	SI	1200	1200	1200	1200
HR	HU	2000	2000	2000	2000
HR	RS	600	600	600	600
HR	SI	2000	2000	2000	2000
RO	HU	1400	1300	1300	1300
RS	HU	600	600	600	600
SI	HU	1200	1200	1200	1200
SK	HU	2000	2000	2000	2000
RO	RS	1300	1300	1300	1300

A villamosenergia-tárolókra vonatkozó költségeket és hatásfokokat a korábban már hivatkozott dolgozat alapján adtam meg [43]. Így az akkumulátorra 2 \$/MWh, a szivattyús tározós erőműre 0,51 \$/MWh változó üzemeltetési és karbantartási költséget adtam meg. Az akkumulátor töltési hatásfokát 93 %-nak, a kisütés hatásfokát pedig 93,46 %-nak vettem. A szivattyús tározós erőműveknél mindkét értéket 88,02 %-ként vettem figyelembe.

3.2. A távhőpiac modellezéséhez szükséges adatok

A távhő piacra vonatkozó adatok gyűjtése jóval nehezebb feladat volt, mint a villamosenergia-piac esetén, ugyanis óránkénti bontásban nem rendelkezünk hőigényekkel. A Központi Statisztikai Hivatal (KSH) és a Magyar Energetikai és Közmű Hivatal (MEKH) [2], [44]jelentései csak az egész naptári évre vonatkozóan adnak meg szolgáltatott energiamennyiséget.

Egy belga kutatás [45] azonban a hőmérsékleti, földrajzi és társadalmi adatok alapján megpróbálták óránként modellezni az egyes országok teljes hőigényét. A kutatásban figyelembe vették az egyes országokban a hőmérséklet alakulását, a régiókra vonatkozó építészeti technológiákat, illetve jellemző fogyasztói magatartásokat is, így megkapva a távhőfogyasztásra vonatkozó adatokat. Németországban az egyik gázszolgáltató validálta is a kutatás helyességét a helyi viszonyokra.

A cikkben megadják ipari létesítmények, családi házak és társasházak hőigényét külön-külön is. Az eredményeket pedig egy nagy Excel táblában publikálták [45].

A lakossági fogyasztást alapul véve összegeztem a családi házak és társasházak összes hőigényét minden vizsgált órára vonatkozóan, legyen ez a teljesítményérték $P_{hő;i}$.

A MEKH beszámolója alapján [2] 2021-ben a lakossági hőigény ~15%-át fedezik távhővel, így a lakossági távhőigény:

$$P_{távhő;lakosság} = 0,15P_{hő;i} \quad (1)$$

Azonban az országos távhőhálózatra ipari fogyasztók is kapcsolódnak, 2018-ban és 2021-ben is a szolgáltatott távhő ~75%-át fogyasztotta a lakosság. Így a teljes távhőigény Magyarországra:

$$P_{össz;Mo.} = \frac{P_{távhő;lakosság}}{0,75} \quad (2)$$

Ezen számításokat minden órára elvégeztem a belga kutatásból számított Excel táblázatban, így megkaptam a 2019-es évre Magyarország óránkénti hőszükségletét MW-ban. Itt a csúcsterhelés egyik napon sem érte el a 2 500 MW-ot, így az összes beépített távhőkapacitásnak is 2 500 MW-ot választottam. A fentebb már említett 2021-es MEKH kiadvány [2] alapján viszont a termelt távhő technológiai eloszlásához hozzá tudtam férni. Itt a 2%-nál kisebb arányban megjelenő technológiákat (pl. a nukleáris fűtőelemeket) figyelmen kívül hagytam. 70 %-ban gázzal, 13 %-ban biomasszával, 11 %-ban geotermikus energiával, 6 %-ban hulladékkal állították elő a távhőt Magyarországon 2021-ben. Egy korábbi kutatással kiegészülve a távhőtermelő egységekre a 3. táblázat bemeneti adatait használtam [6].

3. táblázat A távhőtermelő egységek bemenő adatai

	Beépített kapacitás [MW]	Hatásfok [-]	Fenntartási költség [\$/MWh]
Gáz	1 750	0,89	0,44
Hulladék	150	0,79	1,2
Geotermikus	275	0,1	1,14
Biomassza	325	0,89	1,41

3.3. A hidrogénpiac modellezéséhez szükséges adatok

Jelenleg Magyarországon a hidrogénnek nincsen kialakult piaca, néhány vegyipari üzemnek van csak szüksége hidrogénre az ammónia gyártásához. Magyarország 2030-ra azonban évi 36 000 tonna hidrogén termelését célozta meg. A hidrogénigény óránkénti profilja olyan adat, mely konkrétan nem elérhető, és a távhőhöz hasonló számításokkal sem adható meg. Így nagyon egyszerűen azt feltételeztem, hogy a hidrogénigény minden órában konstans, vagyis az óránként megtermelendő hidrogén a következőképpen számítható:

$$m = \frac{36\,000\text{ t/év} \cdot 10^3\text{ kg/t}}{365\text{ nap/év} \cdot 24\text{ óra/nap}} = 4\,109\text{ kg/óra} \quad (3)$$

Annak érdekében, hogy a különböző technológiák hidrogéntermelése a modellben értelmezhetővé váljon, 1 kg hidrogén energiatartalmát 35 kWh-nak [46] feltételeztem. Konstans teljesítményű hidrogéntermelést feltételezve ekkor a 100% hatásfokkal előállított hidrogén óránkénti teljesítményigénye:

$$P_{H_2} = \frac{m \cdot 35\text{ kWh/kg}}{1\text{ h}} = 143\,815\text{ kW} = 143,815\text{ MW} \quad (4)$$

A továbbiakban azt feltételeztem, hogy az év 8 760 órájában ez a hidrogénigény. Emellé pedig egy napnyi hidrogénszükséglet tárolására alkalmas kapacitást definiáltam a modellbe. A Hidrogénstratégiának megfelelően 240 MW elektrolizálót is elhelyeztem a modellben, mely Magyarország villamosenergia-piacáról vételezi az áramot.

3.4. A modell felépítése a PyPSA segítségével

A PyPSA majdnem minden funkciót tartalmazott, melyre az elemzés során szükség volt. Minimálisan kellett a kódot kiegészíteni: a határköltséget (marginal_cost [\$/MWh]) felbontottam üzemanyag (fuel_cost [\$/GJ]) és változó üzemeltetési és karbantartási (vom_cost [\$/MWh]) költségek összegére. Ugyanezt a módszertant alkalmaztam mind a távhő, mind a villamos energia esetében. Így ennek számítása a következőként történik:

$$\text{határkötség} = \frac{\text{üzemanyag költség} \cdot 3,6}{\eta} + \text{változó karbantatási költség} \quad (5)$$

, ahol η a hatásfokot jelöli, 3,6-os szorzó pedig a MWh és a GJ mértékegységek közötti átváltásból származik.

A modell Pythonban íródott, a programnyelv 3.11-es verzióját használtam. A kódolást a Visual Studio Code Jupyter Notebook-jában végeztem.

A bemeneti adatokat Python „dictionary” típusú változóként adtam meg. A hatásfokokat, üzemanyagköltségeket, változó üzemeltetési és karbantartási költségeket, minimális állási és futási időket, valamint a minimális stabil szintet megadtam minden technológiára vonatkozóan. Az egyes országok által alkalmazott technológiákat, azok beépített kapacitását, illetve az országok közötti határkeresztező kapacitásokat ugyanilyen módon definiáltam, illetve létrehoztam egy klasszikus Python listát, mely a vizsgált országokat tartalmazza.

A kis moduláris reaktor esetében a fentebb már definiált nukleáris technológia üzemanyagköltségével számoltam. A változó karbantartási költséget azonban a nukleáris technológiára jellemző költség másfélszereseként definiáltam, ugyanis a kisebb méretek mindig nagyon változó üzemeltetési és karbantartási költséget eredményeznek [47].

Ez után következett a korábbiakban már bemutatott idősoros adatok beolvasása a CSV fájlokból. Ezt a Python Pandas könyvtára segítségével tettem meg, a megújuló termelésre, a villamos energia- és távhőtermelésre, továbbá hidrogéntermelésre vonatkozó óras adatokat olvastam be a megfelelő változókba. Azonban a beolvasás „Dataframe” típusú változókba történik, melyekből az adatok kinyerése nem egyértelmű feladat, így ezeket átkonvertáltam listákká. Létrehoztam még egy listát, mely a vizsgált időpontokat tartalmazza.

A PyPSA modell felépítése egy „Hálózat” típusú változó létrehozásával történt, ennek átadtam a vizsgálandó időpontok listáját is. A „Hálózathoz” több különböző hálózati elem adható hozzá.

Definiáltam „Gyűjtősíneket” is: minden országnak egy külön villamosenergia-gyűjtőszint, Magyarország esetére pedig hő és hidrogén gyűjtőszint is definiáltam, továbbá a kapcsolt energiatermelés megvalósításának érdekében külön egy SMR gyűjtőszint is deklaráltam,

azonban ennek pontos megvalósítását a későbbiekben részletezem. A gyűjtősínekhez nevet és energiahordozót adtam meg paraméterként.

Az egyes gyűjtősíneket „Link”-ek kapcsolják össze. A linknek megadtam, hogy melyik két gyűjtősínt kell összekötnie, illetve a rajta átmenni képes névleges teljesítményt adtam át. Jelenleg a modell a bizonyos országok közötti határkeresztező import / export kapacitások különbözőségét nem tudja kezelni, így mindkét irányban ugyanaz a teljesítmény képes haladni. A hidrogéntermelés megvalósítása nem generátorral történik, ugyanis az elektrolizáló a hidrogént a hálózatról felvett elektromos árammal állítja elő. Így ezt is egy „Link”-ként kezeltem, melynek átadtam hatásfokot, változó üzemeltetési és karbantartási költséget és beépített teljesítményt.

A gyűjtősínhez kapcsolódnak a „Load” elemek, melyek a teljesítményigényeket határozzák meg és a következő bemeneti paraméterekkel rendelkeznek: a teljesítmény értéke, melyik gyűjtősínről veszi fel az energiát, illetve minden ilyen elemnek van egy neve. Ennek definiálása akár a hő, akár a villamos energia, akár a hidrogén esetében ugyanolyan módon történik.

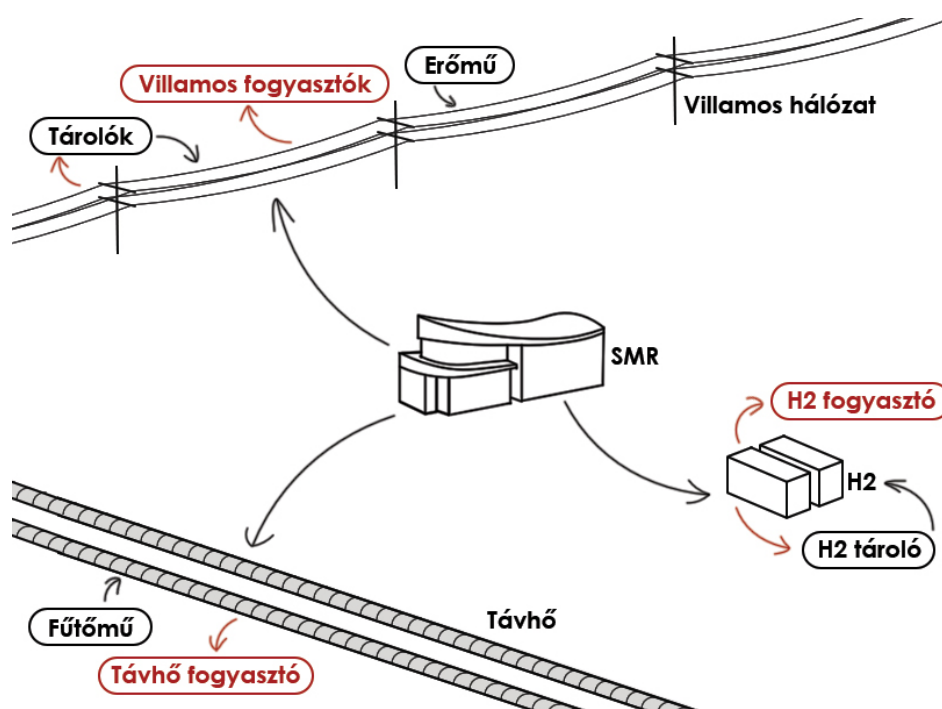
Szintén a gyűjtősínekhez kapcsolódnak a „Generátor” típusú elemek. A generátorok a gyűjtősínekre termelnek. A generátorként definiáltam az összes erőművet. A generátoroknak nevet kell adni és meg kell határozni, hogy mely gyűjtősínekre termelnek. Ezeken túl a futtatásokhoz át kell adni számukra a termelés határköltségét, a nominális teljesítményt, a minimális futási és állási időt, a minimális stabil szintet, továbbá azt is, hogy az aktuális időpillanatban a beépített kapacitásuk hányad részével képesek termelni.

A gyűjtősínekről teljesítmény felvételére és leadására is alkalmasak a „Tároló egységek”. A tárolók definiálására a szivattyús-tározós erőművekhez kapcsolódó víztározók, illetve az akkumulátorok modellezésének érdekében volt szükség. A tároló egységeknek meg kell adni újfent a nevét, illetve, hogy mely gyűjtősínről veszi fel és adja le a teljesítményt. Ezentúl definiáltam a tárolók kezdeti töltöttségi szintjét (ez a nulladik időpillanatban vett töltöttséget mutatja), a betárolás és kitárolás hatásfokát, illetve a maximális tárolókapacitást. A turbina és szivattyú teljesítménye külön nem adható meg, azonban a turbina teljesítményét a tároló egység névleges teljesítményeként definiálva megadhatjuk a szivattyú teljesítményét a névleges teljesítmény arányában, ezzel igazából teljesen kezelhetővé téve

a ki és betárolás teljesítményének kérdését. A tárolókra változó üzemeltetési és karbantartási költséget adtam csak meg.

A fentieket a PyPSA szintaktikájának megfelelően kódoltam le. Ennek ismertetése nem képezi a dolgozat részét, csak a modell felépítésének elméleti megfontolásai. A teljes kód a magyarázó kommentekkel a 7.2-es fejezetben tekinthető meg.

A kutatásom fő részét pedig a trigenerációs kis moduláris reaktor hálózatba történő beépítése jelentette. A trigenerációs technológiák egyszerre három termék előállítására képesek. Dolgozatomban olyan kis moduláris reaktorokat vizsgáltam, melyek egyszerre képesek hő, villamos energia és hidrogén termelésére is. A kis moduláris reaktort egy generátorként definiáltam, melynek ugyanazon paramétereket adtam át, mint bármely másik generátornak. A különbség azonban az volt, hogy a reaktor az általa termelt hőteljesítményt egy köztes gyűjtősinbe táplálta be. Ebből a köztes gyűjtősinből a „Linkek” segítségével kapcsoltam rá Magyarország villamos energia-, távhő- és hidrogén-hálózatára. Ebben az esetben a linkeknek megadtam az átalakítás hatásfokát, illetve annak költségét. Ezzel pedig teljessé vált Magyarország energiarendszere. Az SMR hő és hidrogéntermelését úgy korlátoztam, hogy az általa megtermelt hőenergia maximum 10-10%-a irányuljon a távhő- és hidrogéntermelésre. Az 1. ábrán látható az energiarendszer egyszerűsített modellje:



1. ábra A magyar energiarendszer modellje

Az így megalkotott rendszer pedig egy lineáris optimalizációs problémaként állt elő, melyet a HIGHS solver segítségével oldott meg a program. A solvert [48] kutatás alapján választottam, ugyanis a cikk alapján ez bizonyult a leggyorsabb szabad hozzáférésű solvernek. Azonban a nagyon komplex modell megoldása lineáris optimalizálással igen sok időt igényelt, egy évnyi adatokkal 20+ óra futási idővel kellett számolnom. A kutatás végeredményeit, vagyis az óránkénti termelési adatokat CSV fájlban kaptam meg, majd ezekből hőtérvépet gyártva vontam le a következtetésem.

4. EREDMÉNYEK BEMUTATÁSA

A dolgozatom fő eredménye egy olyan nyílt forráskódú energiapiaci modell megalkotása, mellyel lehetővé válik a trigenerációs kis moduláris reaktorok termelésének elemzése. A fejezetben bemutatom, hogy modellem ténylegesen működik és reális, előzetes elvárásoknak megfelelő eredményeket ad.

A PyPSA könyvtár forráskódjában végül kevés változtatást eszközöltem. A határköltséget felbontottam a változó karbantartási, illetve üzemanyagköltség összegére, emellett bizonyos funkcióbeli hiányosságokat a modell eszközeivel valósítottam meg (például kapcsolt energiatermelő egységet nem lehetett a modellben elhelyezni), melyeket az előző fejezetben már bemutattam.

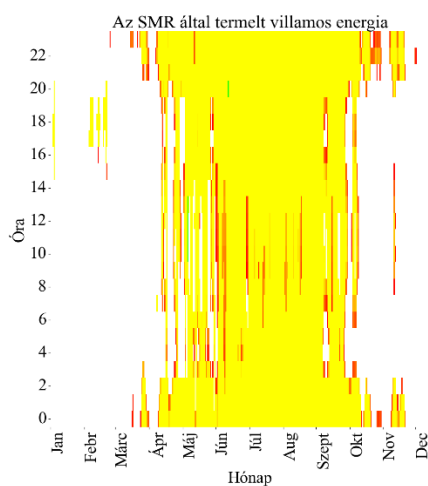
A munka végeredményét mutató hőtérképek pedig a 2, 3. és 4. ábrákon láthatóak. Az ábrákon egy teljes év minden időpillanatában ábrázoltam az aktuális hő-, villamosenergia- és hidrogéntermelést.

A futtatás végén idősoros adatokat kaptam, melyekből hőtérképet készítettem, melyek alapján nyilatkozhatunk az SMR-ek szerepéről a rendszerben.

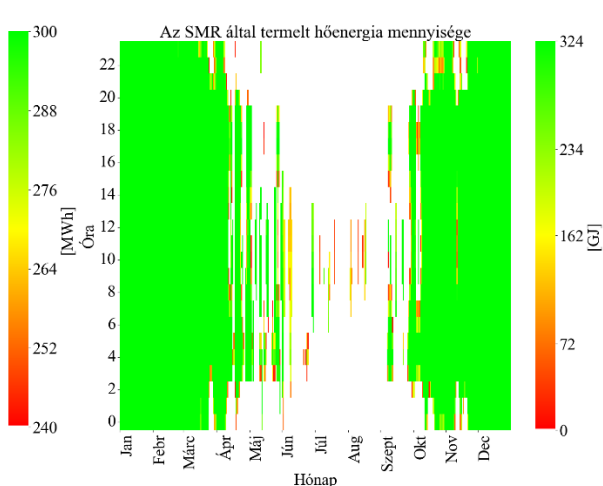
A 3. ábrán mutattam be az SMR hidrogéntermelését. Látható, hogy a reaktornak a hidrogénhálózatba gyakorlatilag minden időpontban megéri betermelnie, így néhány órát leszámítva mindig maximális teljesítményen termel. A hidrogénpiac azonban még egészen kezdetleges, csak a kis moduláris reaktor, illetve a 240 MW-os elektrolizáló termel 1 napos tároló kapacitással. Az SMR által azonban relatíve olcsón termelhető meg a hidrogén, így teljesen versenyképes a piaci áron dolgozó elektrolizálóval szemben.

A villamos energia termelése a 2. ábrán látható. A reaktor minden órában legalább 240 MWh áramot termel, ábrámon ez a mennyiség jelenti a fehér színt. A reaktor elméletileg képes lenne hőteltjesítményének 100%-ából is villamos energiát termelni, azonban ilyen szcenárió a vizsgált helyzetben csak néhány óra esetében állt fenn. Az viszont egyértelműen látszik, hogy a reaktoral a villamosenergia-piacba történő betermelés mindig megéri, még akkor is, amikor a hálózatban nagy részarányban termelnek a megújuló kapacitások.

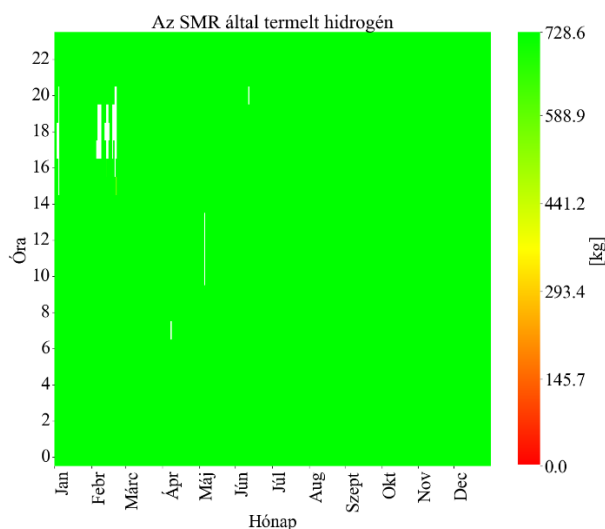
Az SMR hőtermelését a 3. ábrán tüntettem fel. Egyértelműen látszik, hogy azokban az időszakokban, amikor nagy a hőigény (vagyis főként a téli időszakban) akkor a kis moduláris reaktor a megengedhető maximális teljesítményén termel a távhőhálózatra. Ennek oka, hogy a távhő termelésében sok fosszilis technológiát alkalmaznak, melyeknek fajlagos költsége magasabb, mint a kis moduláris reaktorok által megtermelt hő esetében. A jelenleg elterjedő megújuló technológiák (szél és fotovoltaiikus erőművek) pedig nem feltétlenül alkalmasak a hő előállítására. Bár az általuk megtermelt villamos energia segítségével a hőszivattyús technológiák elméletben alkalmazhatóak lennének, azonban ezek relatíve alacsony hőmérsékleten képesek hőenergia termelésére, melyhez a jelenlegi hálózatok átalakítására lehet szükség, melynek megtörténte 2030-ig nem biztos, hogy bekövetkezik.



2. ábra Az SMR villamos energia termelése



3. ábra Az SMR hőenergia termelése



4. ábra Az SMR hidrogén termelése

5. ÖSSZEFOGLALÁS ÉS EREDMÉNYEK ÉRTÉKELÉSE

A jövő energiarendszerének felépítése még nagyon képlékeny lehet. Napjainkban egyértelműen a megújulók minél nagyobb részarányú alkalmazása a cél, azonban egy energiarendszer kizárólag megújuló kapacitásokra történő felépítése rengeteg potenciális veszélyt hordoz magában. Termelésük kiszámíthatatlan, velük a hőtermelés nehézkes, emellett a frekvenciatartás kérdése is sokkal nagyobb kihívást jelent, mint a konvencionális technológiák esetében. Ezen problémák mindegyikére megoldást jelenthetnek a kis moduláris reaktorok, a trigenerációs technológia alkalmazása pedig a terheléskövetéshez is hozzájárulhat. A kis moduláris reaktorok pedig nemcsak koncepció-szinten léteznek. Kínában a villamos hálózatra termel a maga 210 MW-os teljesítményével a HTR-PM [49] reaktortípus. Emellett Oroszországban a BREST 300 [50] ólomhűtésű reaktor konstrukciója is megkezdődött már, továbbá számtalan koncepció született már különböző egzotikus kis moduláris reaktorokra is.

A technológia tehát még egyértelműen a jövőé, de az már napjainkban is látszik, hogy mindenképpen megvalósítható lesz, így a kis moduláris reaktorok néhány évtizeden belül az energiarendszerek szerves részét képezhetik. Azonban ehhez még szükséges kis moduláris reaktorokkal, azon belül kogenerációs és trigenerációs reaktorokkal foglalkozni, termelésüket minél pontosabban modellezni.

A dolgozatban bemutatam 5 energiapiac modellezésre alkalmas szoftvert, kitérve, hogy milyen feladatok elvégzésére lehetnek alkalmasak, milyen kutatások során alkalmazták már őket. Ezek közül kiválasztottam azt a modellt, mely leginkább alkalmas lehet kapcsolt energiarendszerek tervezésére, modellezésére.

Bemutattam Magyarország és szomszédjai energiapiacának és rendszerének felépítését, valamint a PyPSA-ban épített modellben néhány sajátos megoldást. A kód leírása után értékeltem a kis moduláris reaktorok trigenerációs termelésének karakterisztikáját.

A 4. fejezetben rámutattam, hogy egy általam vizsgált villamosenergia-rendszerben milyen szerepe lehet egy trigenerációs kis moduláris reaktornak. Az eredményeim azt mutatták, hogy a technológia alkalmazása érdemes, a hidrogénpiacon és a távhőszektorban egyértelműen nagyon előnyös tulajdonságai vannak. Ezen túl a villamos energia esetében

zsinór villamosenergia-termelő erőműként működhet, azonban a hidrogén- és távhőtermelés nagyságának flexibilis változtatásával az SMR a villamosenergia-piac terheléskövetését is segítheti, ugyanis jelenleg a távhő és hidrogénszektorban a részesedése igen kismértékű.

A dolgozat keretein belül megkezdett kutatásban sok potenciál van még, melyekkel a jövőben szeretnék foglalkozni. Ezen továbbfejlesztési irányok a következők:

- Hidrogén és a távhőpiac termelői és fogyasztói oldalának kibővítése.
- Kis moduláris reaktor hidrogén és távhő termelésének továbbfejlesztése, pontosabb modellezése.
- Érzékenységi vizsgálat az SMR változó üzemeltetési és karbantartási költsége, valamint a különböző piacokon résztvevő erőművek költségére vonatkozóan.
- Algoritmus fejlesztése az SMR-rel termelt hőenergia optimális elosztására a villamosenergia-, hő- és hidrogéntermelés között.
- Szén-dioxid kibocsátás modellezése a fosszilis tüzelésű erőművekben.
- Határkeresztező kapacitások eltérő import / export kapacitásának modellezése
- Kapcsolt hő és villamosenergia-termelés modellezése.
- A 3 piac hosszú időtávú (20 – 50 éves) beruházásainak szimulációja.

6. FELHASZNÁLT FORRÁSOK

- [1] "A MAGYAR VILLAMOSENERGIA RENDSZER 2022. ÉVI ADATAI", MEKH, MAVIR, 2022.
- [2] "A MAGYAR TÁVHŐSZEKTOR 2021. ÉVI ADATAI DATA OF THE HUNGARIAN DISTRICT HEATING SECTOR 2021 Tiszta energia, fenntartható környezet."
- [3] "MAGYARORSZÁG NEMZETI HIDROGÉNSTRATÉGIÁJA STRATÉGIA A TISZTA HIDROGÉN ÉS HIDROGÉNTÉCHNOLÓGIÁK HAZAI BEVEZETÉSÉRE ÉS A HIDROGÉNIPAR HÁTTÉRBÁZISÁNAK MEGTEREMTÉSÉRE."
- [4] Intergovernmental Panel on Climate Change, *Climate change 2014: synthesis report: longer report*.
- [5] M. Sonnberger *et al.*, "Climate concerned but anti-nuclear: Exploring (dis)approval of nuclear energy in four European countries," *Energy Res Soc Sci*, vol. 75, p. 102008, May 2021, doi: 10.1016/J.ERSS.2021.102008.
- [6] M. J. Mayer, B. Biró, B. Szücs, and A. Aszódi, "Probabilistic modeling of future electricity systems with high renewable energy penetration using machine learning," *Appl Energy*, vol. 336, Apr. 2023, doi: 10.1016/j.apenergy.2023.120801.
- [7] U. K. Pata and A. Samour, "Do renewable and nuclear energy enhance environmental quality in France? A new EKC approach with the load capacity factor," *Progress in Nuclear Energy*, vol. 149, p. 104249, Jul. 2022, doi: 10.1016/J.PNUCENE.2022.104249.
- [8] K. Zawalińska, J. Kinnunen, P. Gradziuk, and D. Celińska-Janowicz, "To whom should we grant a power plant? Economic effects of investment in nuclear energy in Poland," *Energies (Basel)*, vol. 13, no. 11, Jun. 2020, doi: 10.3390/en13112687.
- [9] A. Aszódi *et al.*, "Comparative analysis of national energy strategies of 19 European countries in light of the green deal's objectives," *Energy Conversion and Management: X*, vol. 12, p. 100136, Dec. 2021, doi: 10.1016/J.ECMX.2021.100136.
- [10] A. Aszódi *et al.*, "The effect of the future of nuclear energy on the decarbonization pathways and continuous supply of electricity in the European Union," *Nuclear Engineering and Design*, vol. 415, p. 112688, Dec. 2023, doi: 10.1016/J.NUCENGDES.2023.112688.
- [11] Essi Lehto, "After 18 years, Europe's largest nuclear reactor starts regular output."

- [12] M. A. Sayegh *et al.*, "Trends of European research and development in district heating technologies," *Renewable and Sustainable Energy Reviews*, vol. 68. Elsevier Ltd, pp. 1183–1192, Feb. 01, 2017. doi: 10.1016/j.rser.2016.02.023.
- [13] B. Ildikó and V. Marianna, "Hidrogén előállításának lehetőségei Possibilities of hydrogen production." [Online]. Available: <http://www.kfgi.uni-miskolc.hu/>
- [14] M. Zsófia, "MISKOLCI EGYETEM GÉPÉSZMÉRNÖKI ÉS INFORMATIKAI KAR ENERGETIKAI ÉS VEGYIPARI GÉPÉSZETI INTÉZET HIDROGÉNTERMELÉS, JOGSZABÁLYOK, FENNTARTHATÓSÁG, AVAGY MILYEN ÚT VEZET EGY HIDROGÉN HAJTOTTA ZÖLD JÖVŐ FELÉ."
- [15] P. A. Pilavachi, "Power generation with gas turbine systems and combined heat and power." [Online]. Available: www.elsevier.com/locate/apthermeng
- [16] D. Burrin, S. Roy, A. P. Roskilly, and A. Smallbone, "A combined heat and green hydrogen (CHH) generator integrated with a heat network," *Energy Convers Manag*, vol. 246, Oct. 2021, doi: 10.1016/j.enconman.2021.114686.
- [17] F. Franzoni, M. Milani, L. Montorsi, and V. Golovitchev, "Combined hydrogen production and power generation from aluminum combustion with water: Analysis of the concept," *Int J Hydrogen Energy*, vol. 35, no. 4, pp. 1548–1559, Feb. 2010, doi: 10.1016/J.IJHYDENE.2009.11.107.
- [18] S. P. Cicconardi, A. Perna, and G. Spazzafumo, "Combined power and hydrogen production from coal. Part B: Comparison between the IGHP and CPH systems," *Int J Hydrogen Energy*, vol. 33, no. 16, pp. 4397–4404, Aug. 2008, doi: 10.1016/J.IJHYDENE.2008.05.066.
- [19] M. Minutillo, A. Perna, and A. Sorce, "Combined hydrogen, heat and electricity generation via biogas reforming: Energy and economic assessments," *Int J Hydrogen Energy*, vol. 44, no. 43, pp. 23880–23898, Sep. 2019, doi: 10.1016/J.IJHYDENE.2019.07.105.
- [20] G. Locatelli, S. Boarin, A. Fiordaliso, and M. E. Ricotti, "Load following of Small Modular Reactors (SMR) by cogeneration of hydrogen: A techno-economic analysis," *Energy*, vol. 148, pp. 494–505, Apr. 2018, doi: 10.1016/J.ENERGY.2018.01.041.
- [21] G. Spazzafumo and G. Raimondi, "Economic assessment of hydrogen production in a Renewable Energy Community in Italy," *e-Prime - Advances in Electrical Engineering, Electronics and Energy*, vol. 4, p. 100131, Jun. 2023, doi: 10.1016/J.PRIME.2023.100131.

- [22] M. Rämä, M. Leurent, and J. G. Devezieux de Lavergne, “Flexible nuclear co-generation plant combined with district heating and a large-scale heat storage,” *Energy*, vol. 193, p. 116728, Feb. 2020, doi: 10.1016/J.ENERGY.2019.116728.
- [23] J. Hampp, M. Düren, and T. Brown, “Import options for chemical energy carriers from renewable sources to Germany,” *PLoS One*, vol. 18, no. 2 February, Feb. 2023, doi: 10.1371/journal.pone.0281380.
- [24] D. Shropshire, “Economic viability of small to medium-sized reactors deployed in future European energy markets,” *Progress in Nuclear Energy*, vol. 53, no. 4, pp. 299–307, May 2011, doi: 10.1016/J.PNUCENE.2010.12.004.
- [25] H. K. Ringkjøb, P. M. Haugan, and I. M. Solbrekke, “A review of modelling tools for energy and electricity systems with large shares of variable renewables,” *Renewable and Sustainable Energy Reviews*, vol. 96, pp. 440–459, Nov. 2018, doi: 10.1016/J.RSER.2018.08.002.
- [26] T. Brown, J. Hörsch, and D. Schlachtberger, “PyPSA: Python for power system analysis,” *J Open Res Softw*, vol. 6, no. 1, 2018, doi: 10.5334/jors.188.
- [27] M. M. Frysztański, J. Hörsch, V. Hagenmeyer, and T. Brown, “The strong effect of network resolution on electricity system models with high shares of wind and solar,” *Appl Energy*, vol. 291, p. 116726, Jun. 2021, doi: 10.1016/J.APENERGY.2021.116726.
- [28] F. Neumann and T. Brown, “The near-optimal feasible space of a renewable power system model,” *Electric Power Systems Research*, vol. 190, p. 106690, Jan. 2021, doi: 10.1016/J.EPSR.2020.106690.
- [29] “calliope-readthedocs-io-en-stable”.
- [30] T. Tröndle, J. Lilliestam, S. Marelli, and S. Pfenninger, “Trade-Offs between Geographic Scale, Cost, and Infrastructure Requirements for Fully Renewable Electricity in Europe,” *Joule*, vol. 4, no. 9, pp. 1929–1948, Sep. 2020, doi: 10.1016/j.joule.2020.07.018.
- [31] P. Laha and B. Chakraborty, “Cost optimal combinations of storage technologies for maximizing renewable integration in Indian power system by 2040: Multi-region approach,” *Renew Energy*, vol. 179, pp. 233–247, Dec. 2021, doi: 10.1016/j.renene.2021.07.027.
- [32] S. Pfenninger and J. Keirstead, “Comparing concentrating solar and nuclear power as baseload providers using the example of South Africa,” *Energy*, vol. 87, pp. 303–314, Jul. 2015, doi: 10.1016/j.energy.2015.04.077.

- [33] T. Trondle, "Supply-side options to reduce land requirements of fully renewable electricity in Europe," *PLoS One*, vol. 15, no. 8 August, Aug. 2020, doi: 10.1371/journal.pone.0236958.
- [34] B. Elliston, I. Macgill, and M. Diesendorf, "Least cost 100% renewable electricity scenarios in the Australian National Electricity Market."
- [35] M. Fripp, "Switch: A planning tool for power systems with large shares of intermittent renewable energy," *Environ Sci Technol*, vol. 46, no. 11, pp. 6371–6378, Jun. 2012, doi: 10.1021/es204645c.
- [36] J. Johnston, B. Maluenda, R. Henríquez, and M. Fripp, "Switch 2.0: A Modern Platform for Planning High-Renewable Power Systems."
- [37] S. Nasirov, A. Girard, C. Peña, F. Salazar, and F. Simon, "Expansion of renewable energy in Chile: Analysis of the effects on employment," *Energy*, vol. 226, p. 120410, Jul. 2021, doi: 10.1016/J.ENERGY.2021.120410.
- [38] M. Fripp, "Making an Optimal Plan for 100% Renewable Power in Hawaii-Preliminary Results from the SWITCH Power System Planning Model Making an Optimal Plan for 100% Renewable Power in Hawaii," 2016.
- [39] "urbs Documentation Release 1.0.0 tum-ens," 2023.
- [40] M. Stüber and L. Odersky, "Uncertainty modeling with the open source framework urbs," *Energy Strategy Reviews*, vol. 29, p. 100486, May 2020, doi: 10.1016/J.ESR.2020.100486.
- [41] B. Hetterich, A. Vandersickel, J. Dorfner, and H. Spliethoff, "Optimal energy supply system and hourly operation plan for the TUM campus Garching using linear programming model URBS," 2016.
- [42] "https://www.energyexemplar.com/plexos."
- [43] Bence Biró and dr. Attila Aszódi, "Út az energiaválságból egy fenntartható energiagazdálkodásba – Magyarország és a szomszédos országok villamosenergia-ellátásának nagy felbontású szimulációja és elemzése," 2022.
- [44] "Magyarország 2021-es távhő adatai." Accessed: Oct. 28, 2023. [Online]. Available: <https://statinfo.ksh.hu/Statinfo/themeSelector.jsp?lang=hu>
- [45] O. Ruhnau, L. Hirth, and A. Praktiknjo, "Time series of heat demand and heat pump efficiency for energy system modeling," *Sci Data*, vol. 6, no. 1, Dec. 2019, doi: 10.1038/s41597-019-0199-y.
- [46] A. Züttel, A. Remhof, A. Borgschulte, and O. Friedrichs, "Hydrogen: The future energy carrier," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 368, no.

1923. Royal Society, pp. 3329–3342, Jul. 28, 2010. doi: 10.1098/rsta.2010.0113.
- [47] Nea, “Small Modular Reactors: Nuclear Energy Market Potential for Near-term Deployment District heating or desalination applications 2 x KLT-40S 25 Gcal/h Low-temperature process heat 70 MWe,” 2016.
- [48] M. Parzen, “Optimization solvers: the missing link for a fully open-source energy system modelling ecosystem”, doi: 10.5281/zenodo.6409432.
- [49] “Status report 96-High Temperature Gas Cooled Reactor-Pebble-Bed Module (HTR-PM).”
- [50] “BREST-OD-300 (RDIPE, Russian Federation).”

7. MELLÉKLETEK

7.1. A vizsgált erőmű jellemzők

Általános erőmű jellemzők:

- *Nettó beépített kapacitás (BT)*: A blokk bruttó beépített kapacitásának és önfogyasztásának különbsége
- *Minimális stabil szint (MSS)*: Az egyes blokkok minimális stabil termelési szintje. E szint alatt a blokk valamilyen műszaki okból tartósan nem üzemeltethető, így a blokk termelését nem lehet rendszerszabályozás miatt ezen szint alá csökkenteni
- *Hatásfok (η)*: Az adott blokk elektromos és hőteljesítményének hányadosa
- *Felterhelési sebesség (FS)*: Meghatározza azt a sebességet, amellyel az erőmű nulláról a minimális stabil szintre történő felterhelése megtörténik
- *Leterhelési sebesség (LS)*: Meghatározza azt a sebességet, amellyel az erőmű minimális stabil szintről a nullára való leterhelése megtörténik
- *Maximális felszabályozási sebesség (MFS)*: Meghatározza azt a maximális sebességet, amellyel az erőmű a minimális stabil szint és a nettó beépített kapacitás között felfele irányú teljesítményváltoztatást tud végezni
- *Maximális leszabályozási sebesség (MLS)*: Meghatározza azt a maximális sebességet, amellyel az erőmű a minimális stabil szint és a nettó beépített kapacitás között lefele irányú teljesítményváltoztatást tud végezni
- *Minimális futási idő (MFI)*: Az a minimális óraszám, ameddig a blokknak üzemelnie kell a bekapcsolás után
- *Minimális állási idő (MÁI)*: Az a minimális óraszám, ameddig a blokknak leállított állapotban kell lennie a leállítást követően
- *Indítási profil* (opcionális): Meghatározza a megawattban mért teljesítményfokozatok sorozatát, amelyeken a blokk indítás közben áthalad
- *Leállítási profil* (opcionális): Meghatározza a megawattban mért teljesítményfokozatok sorozatát, amelyeken a blokk leállítást követően áthalad
- *Üzemanyag költség (ÜK)*: A blokkok által felhasznált üzemanyag egységnyi tüzelőhőre vonatkozó költsége
- *Változó üzemeltetési és karbantartási költség (VÜéKK)*: A blokk változó üzemeltetési és karbantartási költsége

- *Állandó üzemeltetési és karbantartási költség (ÁÜéKK)*: A blokk fix éves üzemeltetési és karbantartási költsége

Atomerőmű specifikus jellemzők:

- *Üzemanyag-átrakási ráta (ÜÁR)*: Az üzemanyag átrakás miatt a blokk rendelkezésre nem állásának várható aránya éves szinten
- *Minimum idő két üzemanyag-átrakás között (MIKÜÁK)*: Az üzemanyag átrakási események között szükséges minimális óraszám, azaz egy kampány hossza
- *Üzemanyag-átrakás átlagos ideje (ÜÁÁI)*: Az üzemanyag átrakás átlagos ideje
- *Üzemanyag-átrakás minimum ideje (ÜÁMiI)*: Az üzemanyag átrakás minimális ideje
- *Üzemanyag-átrakás maximum ideje (ÜÁMaI)*: Az üzemanyag átrakás maximum ideje
- *Nem várható kiesés aránya (NVKA)*: Meghatározza a nem tervezett kiesések várható mértékét, amelyek a termelőkapacitás részleges vagy teljes kiesését eredményezik egy bizonyos időtartamra
- *Javítás átlagos ideje (JÁI)*: A javítás átlagos ideje
- *Javítás minimum ideje (JMiI)*: A javítás minimális ideje
- *Javítás maximum ideje (JMaI)*: A javítás maximum ideje

7.2. A Pythonban megírt kód

```
#imports

import warnings
import pypsa
import pandas as pd
import matplotlib as plt
import linopy
import numpy as np
from pyomo.environ import Constraint

print("ok")
#initial datas

#Efficiencies
efficiencies = {"Nuclear": 0.337, "Lignite": 0.416, "Gas": 0.544, "Oil": 0.368, "Hydro-RoR": 1, "Hydro-Reservoir": 1, "Wind": 1, "Solar": 1, "Geothermal": 0.218, "Biomass": 0.342, "Waste": 0.33, "Biogas": 0.390, "Unserved": 1}

#Fuel price in $/GJ
fuel_prices = {"Nuclear": 2.095, "Lignite": 4.675, "Gas": 14.462, "Oil": 23.878, "Hydro-RoR": 0, "Hydro-Reservoir": 0, "Wind": 0, "Solar": 0, "Geothermal": 0, "Biomass": 11.1, "Waste": 0, "Biogas": 11.568, "Unserved": 400}

#VO&M Charge in $/MWh
vom_charges = {"Nuclear": 6.847, "Lignite": 4.246, "Gas": 5.753, "Oil": 2.959, "Hydro-RoR": 0, "Hydro-Reservoir": 0, "Wind": 4.052, "Solar": 0, "Geothermal": 6.271, "Biomass": 6.979, "Waste": 18.241, "Biogas": 10.517, "Unserved": 1}

#CO2 emissions kg/GJ -->now is not used
co_emissions = {"Nuclear": 0, "Lignite": 760*3.6, "Gas": 370*3.6, "Oil": 710*3.6, "Hydro-RoR": 0, "Hydro-Reservoir": 0, "Wind": 0, "Solar": 0, "Geothermal": 0, "Biomass": 0, "Waste": 0, "Biogas": 0, "Unserved": 0}

#Price of CO2 $/tons
co_price = 0

#countries
countries = ["Hungary", "Austria", "Slovenia", "Slovakia", "Croatia", "Serbia"]

#power plant capacities in Hungary and neighbouring countries
p_nomm = {
```

```

    "Hungary": {"Nuclear": 4400, "Gas": 2400, "Hydro-RoR": 57.8,
"Wind":329, "Solar": 6400, "Geothermal": 2.71, "Biomass": 796, "Waste":
100, "Unserved": 2000},
    "Austria": {"Gas": 3301, "Oil": 164, "Hydro-RoR": 5543, "Hydro-
Reservoir": 2965,"Wind": 7211, "Solar": 7044, "Biomass": 696, "Waste":
103, "Unserved": 10000 },
    "Croatia": {"Nuclear": 348, "Lignite": 192, "Gas": 1048,"Hydro-RoR":
549, "Hydro-Reservoir": 1856, "Wind": 1364, "Solar": 768, "Geothermal":
17, "Biomass": 148, "Unserved": 2000},
    "Romania": {"Nuclear": 1975, "Lignite": 1980, "Gas": 2958, "Oil":100,
"Hydro-RoR": 3440, "Hydro-Reservoir": 4153, "Wind": 5255, "Solar": 5054,
"Biomass": 137, "Unserved": 2000},
    "Slovenia":{"Nuclear": 712, "Lignite": 689, "Gas": 1555,"Hydro-RoR":
1199, "Wind": 150, "Solar": 1650, "Biomass": 90, "Biogas": 34, "Unser-
ved": 2000},
    "Slovakia":{"Nuclear": 2880, "Gas": 1153,"Hydro-RoR": 1304, "Hydro-
Reservoir": 451, "Wind": 500, "Solar": 1200, "Geothermal": 4, "Biomass":
200, "Biogas": 200, "Unserved": 2000},
    "Serbia": { "Lignite": 4000, "Hydro-RoR": 2729, "Hydro-Reservoir":
629, "Wind": 3510, "Solar": 1540, "Unserved": 2000}
}

#transmissions
transmission = {
    "Hungary": {"Austria":800, "Slovakia": 2000, "Slovenia": 1200, "Roma-
nia": 1300, "Serbia": 600, "Croatia": 2000},
    "Austria": {"Slovenia": 1200 },
    "Croatia":{"Serbia": 600, "Slovenia": 2000},
    "Serbia":{"Romania": 1300}
}

min_downs = {"Nuclear": 18.000, "Lignite": 8.667, "Gas": 4.200, "Oil":
5.200, "Hydro-RoR": 0, "Hydro-Reservoir": 0, "Wind": 0, "Solar": 0,
"Geothermal": 0, "Biomass": 0, "Waste": 0, "Biogas": 0, "Unserved": 0}
p_mins = {"Nuclear": 0.45, "Lignite": 0.433, "Gas": 0.430, "Oil": 0.5,
"Hydro-RoR": 0, "Hydro-Reservoir": 0, "Wind": 0, "Solar": 0, "Geother-
mal": 0.5, "Biomass": 0.3, "Waste": 0, "Biogas": 0.3, "Unserved": 0}
min_ups = {"Nuclear": 10.000, "Lignite": 8.667, "Gas": 3.200, "Oil":
5.200, "Hydro-RoR": 0, "Hydro-Reservoir": 0, "Wind": 0, "Solar": 0,
"Geothermal": 0, "Biomass": 0, "Waste": 0, "Biogas": 0, "Unserved": 0}

#n = 2000
print("ok")
#calculating datas to PyPSA
#fuel prices in $/MWh, with effeciencies
for price in fuel_prices:

```

```

    fuel_prices[price] *= 3.6
    fuel_prices[price] /= efficiencies[price]

#the full variable cost of the electricity in $/MWh (marginal cost)
for tech in fuel_prices:
    fuel_prices[tech] += vom_charges[tech]
marginal_costs = fuel_prices

print("ok")
#price of SMR electricity/ MWh
SMR_electricity_price = 2.09*3.6/0.3+vom_charges["Nuclear"]*1.5
SMR_heat_price = 2.09*3.6/0.9+vom_charges["Nuclear"]*1.5
SMR_hydrogen_price = 2.09*3.6/0.3/0.75+vom_charges["Nuclear"] + 1.04

#price of heat
gas_heat_price = 14.462*3.6/0.89+0.44
renewable_heat_price = 1.14 #this means geothermal
renewable_heat_price2 = 1.41+11.1*3.6/0.82 #biomass
waste_heat_price = 1.2
#Read loads and renewable production

#Austria
df = pd.read_csv("AT_Load_1980-2019.csv")
atload = df.tail(8760)
df = pd.read_csv("AT_Reserv_1980-2019.csv")
atreserv = df.tail(8760)
df = pd.read_csv("AT_RoR_1980-2019.csv")
atror = df.tail(8760)
df = pd.read_csv("AT_Solar_1980-2019.csv")
atsolar = df.tail(8760)
df = pd.read_csv("AT_Wind_1980-2019.csv")
atwind = df.tail(8760)

#Croatia
df = pd.read_csv("HR_Load_1980-2019.csv")
hrload = df.tail(8760)
df = pd.read_csv("HR_Reserv_1980-2019.csv")
hrreserv = df.tail(8760)
df = pd.read_csv("HR_RoR_1980-2019.csv")
hrror = df.tail(8760)
df = pd.read_csv("HR_Solar_1980-2019.csv")
hrsolar = df.tail(8760)
df = pd.read_csv("HR_Wind_1980-2019.csv")
hrwind = df.tail(8760)

```

```

#Hungary
df = pd.read_csv("HU_Load_1980-2019.csv")
huload = df.tail(8760)
df = pd.read_csv("HU_RoR_1980-2019.csv")
huror = df.tail(8760)
df = pd.read_csv("HU_Solar_1980-2019.csv")
husolar = df.tail(8760)
df = pd.read_csv("HU_Wind_1980-2019.csv")
huwind = df.tail(8760)

#Romania
df = pd.read_csv("RO_Load_1980-2019.csv")
roload = df.tail(8760)
df = pd.read_csv("RO_Reserv_1980-2019.csv")
roreserv = df.tail(8760)
df = pd.read_csv("RO_RoR_1980-2019.csv")
roror = df.tail(8760)
df = pd.read_csv("RO_Solar_1980-2019.csv")
rosolar = df.tail(8760)
df = pd.read_csv("RO_Wind_1980-2019.csv")
rowind = df.tail(8760)

#Serbia
df = pd.read_csv("RS_Load_1980-2019.csv")
rsload = df.tail(8760)
df = pd.read_csv("RS_Reserv_1980-2019.csv")
rsreserv = df.tail(8760)
df = pd.read_csv("RS_RoR_1980-2019.csv")
rsror = df.tail(8760)
df = pd.read_csv("RS_Solar_1980-2019.csv")
rssolar = df.tail(8760)

#Slovenia
df = pd.read_csv("SI_Load_1980-2019.csv")
slload = df.tail(8760)
df = pd.read_csv("SI_RoR_1980-2019.csv")
slror = df.tail(8760)
df = pd.read_csv("SI_Solar_1980-2019.csv")
slsolar = df.tail(8760)
df = pd.read_csv("SI_Wind_1980-2019.csv")
slwind = df.tail(8760)

#Slovakia
df = pd.read_csv("SK_Load_1980-2019.csv")
skload = df.tail(8760)
df = pd.read_csv("SK_Reserv_1980-2019.csv")
skreserv = df.tail(8760)

```

```

df = pd.read_csv("SK_RoR_1980-2019.csv")
skror = df.tail(8760)
df = pd.read_csv("SK_Solar_1980-2019.csv")
sksolar = df.tail(8760)
df = pd.read_csv("SK_Wind_1980-2019.csv")
skwind = df.tail(8760)

#Hungary heat and hydrogen
df = pd.read_csv("heat_demand_HU.csv", sep = ";")
huheat = df.tail(8760)
df = pd.read_csv("hu_hydrogen_load.csv", sep=";")
huhydrogen = df.tail(8760)

print("ok")
#making snapshots
snapshots = huload["Datetime"].tolist()

#converting the loads of countries from dataframe to list
huloads = huload["Load (2030)"].tolist()
atloads = atload["Load (2030)"].tolist()
slloads = slload["Load (2030)"].tolist()
skloads = skload["Load (2030)"].tolist()
roloads = roload["Load (2030)"].tolist()
hrloads = hrload["Load (2030)"].tolist()
rsloads = rsload["Load (2030)"].tolist()
huheats = huheat["Power"].tolist()
huhydrogens = huhydrogen["Hydrogen production"].tolist()

#renewable datas in %, so we have to convert them into fractions

#hungary renewables
hurors = huror["Hydro Run-of-river and poundage [%]"].tolist()
hurors[:] = [x / 100 for x in hurors]
huwinds = huwind["Wind"].tolist()
huwinds[:] = [x / 100 for x in huwinds]
husolars = husolar["Solar"].tolist()
husolars[:] = [x / 100 for x in husolars]

#Croatia
hrreservs = hrreserv["Hydro Water Reservoir [%]"].tolist()
hrreservs[:] = [x / 100 for x in hrreservs]
hrrors = hrror["Hydro Run-of-river and poundage [%]"].tolist()
hrrors[:] = [x / 100 for x in hrrors]
hrsolars = hrsolar["Solar"].tolist()
hrsolars[:] = [x / 100 for x in hrsolars]

```

```

hrwinds = hrwind["Wind"].tolist()
hrwinds[:] = [x / 100 for x in hrwinds]

#Austria
atreservs = atreserv["Hydro Water Reservoir [%]"].tolist()
atreservs[:] = [x / 100 for x in atreservs]
atrors = atror["Hydro Run-of-river and poundage [%]"].tolist()
atrors[:] = [x / 100 for x in atrors]
atsolars = atsolar["Solar"].tolist()
atsolars[:] = [x / 100 for x in atsolars]
atwinds = atwind["Wind"].tolist()
atwinds[:] = [x / 100 for x in atwinds]

#Romania
roreservs = roreserv["Hydro Water Reservoir [%]"].tolist()
roreservs[:] = [x / 100 for x in roreservs]
rorors = roror["Hydro Run-of-river and poundage [%]"].tolist()
rorors[:] = [x / 100 for x in rorors]
rosolars = rosolar["Solar"].tolist()
rosolars[:] = [x / 100 for x in rosolars]
rowinds = rowind["Wind"].tolist()
rowinds[:] = [x / 100 for x in rowinds]

#Serbia
rsreservs = rsreserv["Hydro Water Reservoir [%]"].tolist()
rsreservs[:] = [x / 100 for x in rsreservs]
rsrors = rsror["Hydro Run-of-river and poundage [%]"].tolist()
rsrors[:] = [x / 100 for x in rsrors]
rssolars = rssolar["Solar"].tolist()
rssolars[:] = [x / 100 for x in rssolars]

#Slovenia
slrors = slror["Hydro Run-of-river and poundage [%]"].tolist()
slrors[:] = [x / 100 for x in slrors]
slsolars = slsolar["Solar"].tolist()
slsolars[:] = [x / 100 for x in slsolars]
slwinds = slwind["Wind"].tolist()
slwinds[:] = [x / 100 for x in slwinds]

#Slovakia
skreservs = skreserv["Hydro Water Reservoir [%]"].tolist()
skreservs[:] = [x / 100 for x in skreservs]
skrors = skror["Hydro Run-of-river and poundage [%]"].tolist()
skrors[:] = [x / 100 for x in skrors]
sksolars = sksolar["Solar"].tolist()

```

```

sksolars[:] = [x / 100 for x in sksolars]
skwinds = skwind["Wind"].tolist()
skwinds[:] = [x / 100 for x in skwinds]

print("ok")
#making the model-->network

network = pypsa.Network()
network.set_snapshots(snapshots)

"""
#defining carriers with their CO2 emissions-->they are in the model but
we do not use them
network.add("Carrier", "gas", co2_emissions=co_emissions["Gas"]/1000)
network.add("Carrier", "uranium", co2_emissions=co_emissions["Nuclear"]/1000)
network.add("Carrier", "lignite", co2_emissions=co_emissions["Lignite"]/1000)
network.add("Carrier", "oil", co2_emissions=co_emissions["Oil"]/1000)
network.add("Carrier", "wind", co2_emissions=co_emissions["Wind"]/1000)
network.add("Carrier", "sun", co2_emissions=co_emissions["Solar"]/1000)
network.add("Carrier", "water", co2_emissions=co_emissions["Hydro-Ror"]/1000)
network.add("Carrier", "biomass", co2_emissions=co_emissions["Biomass"]/1000)
network.add("Carrier", "biogas", co2_emissions=co_emissions["Biogas"]/1000)
network.add("Carrier", "waste", co2_emissions=co_emissions["Waste"]/1000)
network.add("Carrier", "heat")
network.add("Carrier", "hydrogen")
"""

for country in countries:
    if country == "Hungary":

        #adding the buses, we have an extra bus: Nheat2--> it is needed
        #to calculate correct with the trigeneration technology
        network.add("Bus", country)
        network.add("Bus", "HungaryHeat")
        network.add("Bus", "HungaryHydrogen")
        network.add("Bus", "Nheat")
        network.add("Bus", "Nheat2")

        #adding the trigeneration SMR with the nominal heat capacity of
        900 MW
        network.add(
            "Generator",

```

```

        "SMR",
        bus = "Nheat",
        p_nom = 1000,
        efficiency = 1.0,
    )

    #connecting the SMR's heat with another heat Bus
    network.add(
        "Link",
        "SMR2Heat0",
        bus0="Nheat",
        bus1="Nheat2",
        efficiency = 1.0,
        p_nom_extendable=True,
    )

    #adding the hydrogen generator link
    network.add(
        "Link",
        "SMR2Hydrogen",
        bus0="Nheat2",
        bus1="HungaryHydrogen",
        efficiency = 0.3*0.85,
        marginal_cost = SMR_hydrogen_price,
        p_nom=100,
        p_nom_extendable=False,
    )

    network.add(
        "Link",
        "Electrolyzer", #from the Hungary Grid to
        bus0="Hungary",
        bus1="HungaryHydrogen",
        efficiency = 0.68,
        marginal_cost = 1.04,
        p_nom=240,
        p_nom_extendable=False,
    )

    #adding the electricity generation link
    network.add(
        "Link",
        "SMR2AC",
        bus0="Nheat2",
        bus1="Hungary",
        marginal_cost = SMR_electricity_price,

```

```

        efficiency = 0.3, #kb kisebb, mint egy nagy reaktor
        p_nom=1000,
        p_nom_extendable=False,
    )

    #adding the heat generation link
    network.add(
        "Link",
        "SMR2Heat",
        bus0="Nheat2",
        bus1="HungaryHeat",
        efficiency = 0.9,
        marginal_cost= SMR_heat_price,
        p_nom=100,
        p_nom_extendable=False,
    )

    #adding the conventional technologies
    for tech in p_nomm[country]:
        network.add(
            "Generator",
            "{} {}".format(country, tech),
            bus=country,
            p_nom=p_nomm[country][tech],
            marginal_cost=marginal_costs[tech],
            efficiency = efficiencies[tech],
            p_max_pu=(huwinds if tech == "Wind" else husolars if tech
== "Solar" else hurors if tech == "Hydro-RoR" else 1),
            p_min_pu=p_mins[tech],
            committable = True,
            min_down_time = min_downs[tech],
            min_up_time = min_ups[tech],
        )

    #adding storage units
    network.add(
        "StorageUnit",
        "Battery",
        bus = country,
        p_nom = 100,
        marginal_cost=2.0,
        efficiency_store = 0.930,
        efficiency_dispatch = 0.9346,
        max_hours = 2,
        state_of_charge_initial = 100,
    )

```

```

#adding the electricity load
network.add(
    "Load",
    "{} load".format(country),
    bus=country,
    p_set=huloads,
)
#adding Hungary's heat load
network.add("Load", "Hungary Heat", bus="HungaryHeat", p_set =
huheats)

#adding Hungary's Gas Heating Plant
network.add(
    "Generator",
    "Gas Heating Plant",
    bus="HungaryHeat",
    marginal_cost=gas_heat_price,
    efficiency = 0.89,
    p_nom = 2500*0.70,
)

#adding Hungary's Waste Heating Plant
network.add(
    "Generator",
    "Waste Heating Plant",
    bus="HungaryHeat",
    marginal_cost=waste_heat_price,
    efficiency = 0.79,
    p_nom = 2500*0.06,
)

#adding Hungary's Geo Heating Plant
network.add(
    "Generator",
    "Geo Heating Plant",
    bus="HungaryHeat",
    marginal_cost=renewable_heat_price,
    efficiency = 0.1,
    p_nom = 2500*0.11,
)

#adding Hungary's Biomass Heating Plant
network.add(
    "Generator",
    "Biomass Heating Plant",

```

```

        bus="HungaryHeat",
        marginal_cost=renewable_heat_price2,
        efficiency = 0.89,
        p_nom = 2500*0.13,
    )

    #Adding Hungary's Hydrogen Load
    network.add("Load", "Hungary H2", bus="HungaryHydrogen", p_set =
huhydrogens)

    #Adding Hydrogen Storage Unit--> it can store one day Hydrogen
demand
    network.add(
        "StorageUnit",
        "Hydrogen Storage",
        bus = "HungaryHydrogen",
        p_nom = 150,
        marginal_cost = 0.51,
        efficiency_store = 0.95,
        efficiency_dispatch = 0.95,
        max_hours = 24,
        state_of_charge_initial = 50*24,
    )

    #we add the generators, buses and links of other countries in the
same way we did in the case of Hungary
    elif country == "Austria":
        network.add("Bus", country)

    for tech in p_nomm[country]:
        network.add(
            "Generator",
            "{} {}".format(country, tech),
            bus=country,
            p_nom=p_nomm[country][tech],
            marginal_cost=marginal_costs[tech],
            efficiency = efficiencies[tech],
            p_max_pu=(atwinds if tech == "Wind" else atsolars if tech
== "Solar" else atrors if tech == "Hydro-RoR" else atreservs if tech
=="Hydro-Reservoir" else 1),
            p_min_pu= p_mins[tech],
            committable = True,
            min_down_time = min_downs[tech],
            min_up_time = min_ups[tech],
        )

```

```

network.add(
    "StorageUnit",
    "Hydro-Pump Storage At",
    bus = country,
    p_nom = 3459.28,
    p_min_pu = -0.739959,
    efficiency_store = 0.8802,
    efficiency_dispatch = 0.8802,
    max_hours = 1722178/3459.28,
    marginal_cost = 0.51,
    state_of_charge_initial = 861089,
)

network.add(
    "Load",
    "{} load".format(country),
    bus=country,
    p_set=atloads,
)

elif country == "Slovakia":
    network.add("Bus", country)

    for tech in p_nomm[country]:
        network.add(
            "Generator",
            "{} {}".format(country, tech),
            bus=country,
            p_nom=p_nomm[country][tech],
            efficiency = efficiencies[tech],
            marginal_cost=marginal_costs[tech],
            p_max_pu=(skwinds if tech == "Wind" else sksolars if tech
== "Solar" else skrors if tech == "Hydro-RoR" else skreservs if tech
=="Hydro-Reservoir" else 1),
            p_min_pu= p_mins[tech],
            committable = True,
            min_down_time = min_downs[tech],
            min_up_time = min_ups[tech],
        )

network.add(
    "StorageUnit",

```

```

        "Hydro-Pump Storage Sk",
        bus = country,
        p_nom = 925.86,
        p_min_pu = -0.894455,
        efficiency_store = 0.8802,
        efficiency_dispatch = 0.8802,
        marginal_cost = 0.51,
        max_hours = 49650/925.86,
        state_of_charge_initial = 24825,
    )

    network.add(
        "Load",
        "{} load".format(country),
        bus=country,
        p_set=skloads,
    )

    elif country == "Serbia":
        network.add("Bus", country)

    for tech in p_nomm[country]:
        network.add(
            "Generator",
            "{} {}".format(country, tech),
            bus=country,
            p_nom=p_nomm[country][tech],
            efficiency = efficiencies[tech],
            marginal_cost=marginal_costs[tech],
            p_max_pu=(rssolars if tech == "Solar" else rsrors if tech
== "Hydro-RoR" else rsreservs if tech == "Hydro-Reservoir" else 1),
            p_min_pu = p_mins[tech],
            committable = True,
            min_down_time = min_downs[tech],
            min_up_time = min_ups[tech],
        )

    network.add(
        "StorageUnit",
        "Hydro-Pump Storage Rs",
        bus = country,
        marginal_cost = 0.51,
        p_nom = 614,
        p_min_pu = -0.91205,
        efficiency_store = 0.8802,

```

```

        efficiency_dispatch = 0.8802,
        max_hours = 194000/614,
        state_of_charge_initial = 97000,
    )

    network.add(
        "Load",
        "{} load".format(country),
        bus=country,
        p_set=rsloads,
    )

    elif country == "Slovenia":
        network.add("Bus", country)

        for tech in p_nomm[country]:
            network.add(
                "Generator",
                "{} {}".format(country, tech),
                bus=country,
                p_nom=p_nomm[country][tech],
                efficiency = efficiencies[tech],
                marginal_cost=marginal_costs[tech],
                p_max_pu=(slwinds if tech == "Wind" else slsolars if tech
== "Solar" else slrors if tech == "Hydro-RoR" else 1),
                p_min_pu=p_mins[tech],
                committable = True,
                min_down_time = min_downs[tech],
                min_up_time = min_ups[tech],
            )

        network.add(
            "StorageUnit",
            "Hydro-Pump Storage S1",
            bus = country,
            marginal_cost = 0.51,
            p_nom = 185,
            p_min_pu = -0.972973,
            efficiency_store = 0.8802,
            efficiency_dispatch = 0.8802,
            max_hours = 2600/185,
            state_of_charge_initial = 1300,
        )

```

```

        network.add(
            "Load",
            "{} load".format(country),
            bus=country,
            p_set=slloads,
        )

    elif country == "Croatia":
        network.add("Bus", country)

    for tech in p_nomm[country]:
        network.add(
            "Generator",
            "{} {}".format(country, tech),
            bus=country,
            p_nom=p_nomm[country][tech],
            efficiency = efficiencies[tech],
            marginal_cost=marginal_costs[tech],
            p_max_pu=(hrwinds if tech == "Wind" else hrsolars if tech
== "Solar" else hrrors if tech == "Hydro-RoR" else hrreservs if tech
=="Hydro-Reservoir" else 1),
            p_min_pu=p_mins[tech],
            committable = True,
            min_down_time = min_downs[tech],
            min_up_time = min_ups[tech]
        )

    network.add(
        "StorageUnit",
        "Hydro-Pump Storage Cr",
        bus = country,
        marginal_cost = 0.51,
        p_nom = 300,
        p_min_pu = -0.6666666667,
        efficiency_store = 0.8802,
        efficiency_dispatch = 0.8802,
        max_hours = 18300/300,
        state_of_charge_initial = 9150,
    )

    network.add(
        "Load",
        "{} load".format(country),
        bus=country,
        p_set=hrloads,

```

```

    )

    elif country == "Romania":
        network.add("Bus", country)

        for tech in p_nomm[country]:
            network.add(
                "Generator",
                "{} {}".format(country, tech),
                bus=country,
                p_nom=p_nomm[country][tech],
                marginal_cost=marginal_costs[tech],
                efficiency = efficiencies[tech],
                p_max_pu=(rowinds if tech == "Wind" else rosolars if tech
== "Solar" else rorors if tech == "Hydro-RoR" else roreservs if tech
=="Hydro-Reservoir" else 1),
                p_min_pu=p_mins[tech],
                committable = True,
                min_down_time = min_downs[tech],
                min_up_time = min_ups[tech],
            )

        network.add(
            "StorageUnit",
            "Hydro-Pump Storage Ro",
            bus = country,
            p_nom = 746,
            marginal_cost = 0.51,
            p_min_pu = -1.384718,
            efficiency_store = 0.8802,
            efficiency_dispatch = 0.8802,
            max_hours = 92000/746,
            state_of_charge_initial = 46000,
        )

        network.add(
            "Load",
            "{} load".format(country),
            bus=country,
            p_set=roloads,
        )

#adding links between countries
if country not in transmission:

```

```

        continue

    for other_country in countries:
        if other_country not in transmission[country]:
            continue
        # NB: Link is by default unidirectional, so have to set p_min_pu
        = -1
        # to allow bidirectional (i.e. also negative) flow
        network.add(
            "Link",
            "{} - {} link".format(country, other_country),
            bus0=country,
            bus1=other_country,
            p_nom=transmission[country][other_country],
            p_min_pu=-1,
        )

print("ok")
network.optimize(solver_name="highs")
df = network.generators_t.p
df
df.to_csv("kesz")
df2 = network.links_t.p1
df2.to_csv("links")
df3 = network.storage_units_t.p_store
df4 = network.storage_units_t.p_dispatch
df5 = network.storage_units_t.state_of_charge

df3.to_csv("p_store")
df4.to_csv("p_dispatch")
df5.to_csv("SOC")

```