

Károsanyag kibocsátási modul továbbfejlesztése SUMO közlekedési szimulációs programhoz

Írta:

Fekete Levente István

SZE – ÉÉKK – közlekedésmérnök Bsc

2. évfolyam

Konzulens:

Nagy Viktor

Egyetemi tanársegéd

Tartalomjegyzék

1. Bevezetés	4
1.1 A kutatás célja	4
2. Környezeti tényezők	5
2.1 Közlekedés hatásai	6
2.2 Lépések a tisztább légkör érdekében	7
2.2.1 Low-emission zones (LEZ): Alacsony kibocsátási zónák:	7
2.2.2 Alacsony kibocsátási zónák lehetőségei szimulált környezetben	7
2.3 CO ₂ és a károsanyag kibocsátási rendszer.....	8
2.3.1 Kibocsátás besorolás rendszere:.....	8
2.4 Mérőműszerek	11
2.4.1 Az infravörös mérés elve	11
3. SUMO használata	12
4. Python, mint programozási nyelv	14
4.1 A programhoz használt modulok.....	14
4.1.1 Fájlkészítő modul - Pandas	15
4.1.2 Matematikai modul - Matplotlib - Adatvizualizációs célokra	15
4.1.3 Numpy – matematikai és adatkezelési modul.....	17
5. Az általam írt program felépítése és működése	17
5.1 Modulok, csomagok importálása.....	17
5.1.1 A csv fájl beolvasása	17
5.1.2 A raszter háló méretének meghatározása	18
5.1.3 Mátrix feltöltése és a mátrix kezelése.....	18
5.1.4 Megjelenítés.....	18
6. Eredmények	19
6.1 Csomóponti vizsgálat	19
6.2 Város szintű károsanyag kibocsátás vizsgálata	20
7. A program fejleszthetősége, jövőbeli tervek.....	22
8. Konklúzió.....	23
Irodalomjegyzék.....	25
Mellékletek.....	27
1. Melléklet:.....	27
2. Melléklet.....	29

Ábrajegyzék

Ábra 1: Atmoszférikus szén-dioxid mennyiség alakulása az évek során [2]	5
Ábra 2: Közúti közlekedés CO ₂ kibocsátása [3]	6
Ábra 3: Közúti közlekedés jármű szerinti CO ₂ kibocsátása [3]	6
Ábra 4: Alacsony kibocsátási zóna táblája – London [8].....	7
Ábra 5: HBEFA 3 és HBEFA 4 közötti eltérés és hiba szemléltetése [13]	10
Ábra 6: Mérőműszerrel mért motor kibocsátási adatai [15]	11
Ábra 7: SUMO szimulációs program működése.....	12
Ábra 8: SUMO konfigurációs fájl tartalma	13
Ábra 9: Szimuláció által készített kibocsátási adatokat tartalmazó fájl	13
Ábra 10: Parancssor által végzett xml-ről csv-re történő átalakítás.....	14
Ábra 11: Output kibocsátási adatok a csv fájlban	15
Ábra 12: Általam készített kezdetleges verzió	16

Ábra 13: Általam írt program modulok importálása és a kimeneti fájl beolvasása	17
Ábra 14: Az általam készített program ábra megjelenítő kódrészlete.....	18
Ábra 15: Négyágú körforgalom	19
Ábra 16: Négyágú csomópont.....	19
Ábra 17: Győr térképe [19]	20
Ábra 18: Hálózat ráközelítve.....	20
Ábra 19: győri hálózat CO ₂ kibocsátás térképe.....	21
Ábra 20: CO kibocsátás szemléltetése a győri hálózati	22
Ábra 21: A hálózat terheltsége, CO ₂ koncentrációja.....	23

Táblázatjegyzék

Táblázat 1: AIR járművek által kibocsátott CO ₂ besorolása [10]	8
Táblázat 2: Euro járműkategóriák [11]	9
Táblázat 3: Euro 4 járműbesorolás összehasonlítása benzin és dízelmotoros járművek esetén [11]	9
Táblázat 4: HBEFA4 és HBEFA3 összehasonlítása [12]	10
Táblázat 5: HBEFA oldal ország szerinti CO ₂ kibocsátás lekérdezése [14]	10
Táblázat 6: Eloszlási függvény működése	18

1. Bevezetés

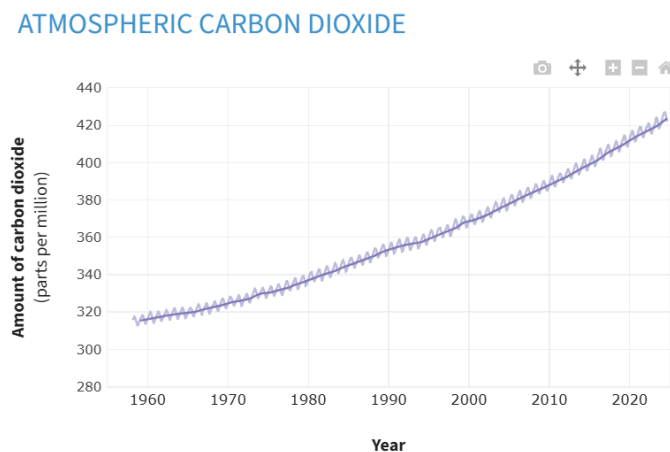
A Tudományos Diákköri dolgozatomban a SUMO nevezetű, open-source szoftverrel készült szimulációt elemzek. E szoftver futtatása által készült kimeneti adatokat felhasználva egy programot készítettem, mely a különböző adatokat rendezi olvasható formátumba. Ezek a fájlok könnyen értelmezhetőek és könnyedén adatelemzések alapjául szolgálhatnak. Célja, hogy egy hálózat vagy város szintű kimutatást és térképet mutasson az adott károsanyag kibocsátásról. A térkép letölthető, nagyítható, színezhető és formázható. A kimeneti fájlban feltüntetett adatokat felhasználva x és y koordinátákat ábrázoltam, mely megmutatja a kibocsátás helyét és mennyiségét az adott gázból. Tanulmányomban szeretném bemutatni, hogy egyes közúti csomópontok, hálózati típusok és elrendezések hogyan befolyásolják az utca vagy a város légkörét. Én a CO₂ és a CO kibocsátásra fogok koncentrálni a dolgozatomban, viszont jóval több mindennel is lehetne foglalkozni a későbbiekben. A gázok és porok légköri viselkedését és ezen anyagok mozgását befolyásoló tényezők, mint például a szél, a fák és az épületeket is számításba vehetőek az adatok ábrázolásánál. Később szeretnék létrehozni egy olyan ablakot, ahol gombok segítségével lehet választani különböző kibocsátási fajták közül, melyek mind-mind új térképen jelennének meg. Továbbá, egy mozgatható kapcsolóval irányítani lehetne az idő múlását, így látni lehetne, hogy hogyan alakult a károsanyag kibocsátás mennyisége a program haladtával.

1.1 A kutatás célja

A program által futtatott szimuláció kimutatásokat ad a károsanyag kibocsátásokról. Ezekkel az adatokkal vizsgálható, hogy melyik utak, csomópontok és városrészek szennyezettek. Károsanyag, szállópor és zajszennyezés kimutatással egy adott városrész, vagy terület élhetősége is vizsgálható. A SUMO kibocsátási fájlokat egy általam írt python program segítségével használtam fel.

2. Környezeti tényezők

A környezeti helyzet egyre súlyosbodik, az EU nagy energia és tőkeforrásokat fordít az üvegházhatású gázok kibocsátásának csökkentésére. [1] „2019-ben 122,3 gramm volt az új autók kilométerenkénti átlagos szén-dioxid-kibocsátása, ami alacsonyabb, mint az EU 2015-2019-es időszakra vonatkozó 130 grammos célértéke, de így is magasabb, mint a 2021-re kitűzött 95 g/km.” írja az Európai Parlament oldalán lévő „Új limit az autók és kisteherautók szén-dioxid-kibocsátására” című cikkben a jelenlegi helyzet alcím alatt.



Ábra 1: Atmoszférikus szén-dioxid mennyiség alakulása az évek során [2]

A diagramot a NOAA klímáért felelős állami szervezet állította ki, mely szervezet a Nemzeti Óceán- és Légkörkutató Hivatal az amerikai kormány tudományos ügynöksége. A méréseket a Mauna Loa megfigyelőállomáson készítették Hawaiiin 1958-ra visszavezetve. Az ábra a havi szén-dioxid kibocsátást mutatja ppm (parts per million) mennyiségben, mely a vizsgált részecske számát mutatja ki egy millió részecskében. [2] A NOAA Nemzeti Környezetvédelmi Központja a következőt írja a szén-dioxid és a légkör helyzetéről. „A szén-dioxid koncentráció emelkedik a fosszilis üzemanyagok miatt, melyet az emberek energiaként égetnek el. A fosszilis tüzelőanyagok, mint például a szén és olaj korbont tartalmaznak, melyet a növényzet kivont az atmoszférából fotoszintézis által sok millió év alatt, melyet mi pár évszázad alatt mind visszajuttatunk az atmoszférába. A 20. század közepe óta az éves fosszilis tüzelőanyagok égetése folyamatosan emelkedett. Az 1960-as években a közel 11 milliárd tonna per évről közel 36.6 milliárd tonnára növekedett a szén-dioxid kibocsátása, a Global Carbon Budget 2023 adatai szerint.” A szén-dioxid napjainkban egyik legtöbbet foglalkozott gázfajtája. E gáz az üvegházhatást okozó gázok közé tartozik, mely hőt vesz fel és csak lassan engedi ki azt, mely felmelegíti a Földet. A nagy szén-dioxid mennyiség káros hatással van az emberekre és a környezetre.

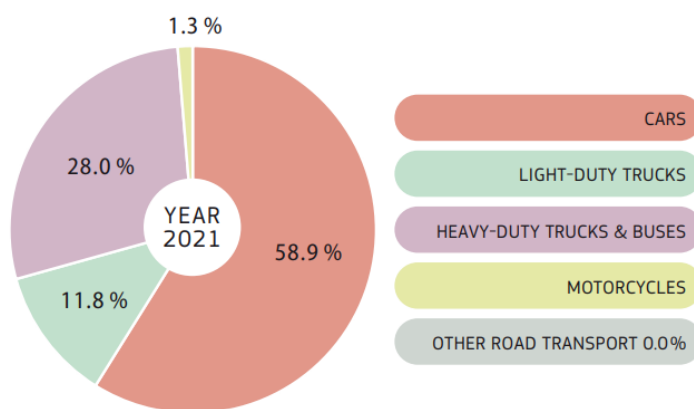
2.1 Közlekedés hatásai

A közlekedés az egyik nagy része a CO₂, szállópor és egyéb gázok kibocsátásában. Egy forgalmi csomópont is befolyásolhatja, hogy mennyi gáz kerül a levegőbe. Egy tudatos és zöld várostervezésnél a csomópontok kialakítása és elhelyezése is befolyásolhatja a levegő minőségét. A házak és a zöld környezet, mint például fás, bokros parkok, terek és fásított utcák elősegíthetik a levegő és a környezet tisztulását.

[3] A közúti közlekedés által kibocsátott CO₂ károsanyagok részesedése az EU-27-ben:

	ROAD TRANSPORTATION				TOTAL TRANSPORT (***)	TOTAL EMISSIONS (**)
	Cars	Light-duty trucks	Heavy-duty trucks and buses			
1990	74.8	46.9	7.1	19.7	16.5	100
1995	75.6	47.2	7.6	19.7	18.9	100
2000	73.9	45.2	8.1	19.4	21.7	100
2005	72.6	43.8	8.4	19.4	22.9	100
2010	72.1	44.0	8.2	18.9	24.1	100
2015	73.3	45.3	8.0	19.0	25.5	100
2020	76.6	45.3	8.6	21.7	25.8	100
2021	76.2	44.9	9.0	21.3	26.7	100

Ábra 2: Közúti közlekedés CO₂ kibocsátása [3]



Ábra 3: Közúti közlekedés jármű szerinti CO₂ kibocsátása [3]

A diagramokon látszik, hogy az évtizedek alatt a közlekedés egyre nagyobb részt foglal el a CO₂ kibocsátás terén. A közúti közlekedés során, a legnagyobb mennyiségű üvegházhatású gázok kibocsátásának a személygépjárműveknek és a kisteherautóknak köszönhető, mivel ezek a járművek teszik ki az Európai Unió szén-dioxid-kibocsátásának mintegy 15%-át. [3] A bal oldali táblázaton látható a közúti közlekedés kibocsátási része az EU ágazataihoz való viszonyulásának az aránya évekre bontva. A jobb oldali kördiagrammon látható a 2021-es év közúti közlekedés által kibocsátott CO₂ érték járműtípusokra bontva. Ez az ábra is mutatja, hogy a személygépjárművek a legtöbb károsanyagot kibocsátó járművek, viszont utánuk a nehéz tehergépjárművek követik a sort.

Az EU káros üvegházhatású gázok kibocsátása elleni tervei nagyban befolyásolják a városok tisztaságát, élhetőségét. Ezek a lépések a jövő generációit fogják szolgálni, hiszen ezek a változtatások nem láthatóak pár év alatt. Az Európai Parlament szén-dioxid kibocsátásról szóló cikk első sora: [4] „A közúti közlekedési ágazat klímasemlegessége érdekében az EU 2035-től betiltja a belső égésű motorral hajtott új személygépkocsik és kisteherautók értékesítését.”

2.2 Lépések a tisztább légkör érdekében

[5] Angilában elsőként Londonban, majd később Birminghamban, Oxfordban, Bristolban, Bath-ban, Glasgow-ban, Edinburgh-ban, Manchester-ben és Newcastle-ben vezettek be hasonló „tisztá levegős” zónákat. Ezeket a zónákat a levegő minősége és a város élhetősége érdekében hozták létre. A bírsági díjak további elriasztás a szennyező gépjárművezetőknek, hogy kerüljék el az érintett tartományt és használjanak alternatív útvonalakat. Mikor egy jármű áthalad ezen a zónán, a rendszer, mely általában egy automatikus rendszámolvasó kamera, díjat szab ki, mely mértéke a zóna besorolásától és a jármű besorolásától függ. Ezekben a körzetekben a város meg is tilthatja az alacsonyabb környezeti kategóriában lévő gépjárművek behaladását. A városokban négy típusú körzet található meg (A, B, C, D). A zónák ösztönzik a járművezetőket a kevésbé szennyező járművek és alternatív útvonalak használatára. [6] A város légterének tisztulását is elősegítheti, mely már a 2022-ben bevezetett övezetekben is megfigyelhető. Az NO₂ koncentrációja éves adatok alapján 2019-hez képest 22%-os csökkenést mutatott és várhatóan csökkenő tendencia figyelhető majd meg.

2.2.1 Low-emission zones (LEZ): Alacsony kibocsátási zónák:

[7] Elsők között London városa vezette be ezen területek szabályozását, mely korlátozza vagy kizárja a szennyező járműveket. Több változata van, például az ULEZ, mely a Szélsőségesen alacsony kibocsátási zóna, mely még szigorúbban korlátozza a járműveket. A legkomolyabb megszorítás a Zéró kibocsátási zónák (ZEZ), ahol a belsőégésű motorral rendelkező járművek kitiltásra kerültek, beleértve a Plug-in hybrid járműveket, melyek képtelenek kibocsátás mentes üzemelésre. Ebben a kerületben csak akkumulátoros elektromos járművek és hidrogénhajtású járművek engedélyezettek. Ezek a korlátozások vonatkoznak a közösségi közlekedésre is.



ábra 4: Alacsony kibocsátási zóna táblája – London [8]

2.2.2 Alacsony kibocsátási zónák lehetőségei szimulált környezetben

Ezek a körzetek egyre nagyobb hangsúlyt kapnak a napjainkban és előkészítik a tisztább és zöldebb városokat a jövőben. E zónák kijelölésével és bevezetésével is lehetne foglalkozni a szimulációban, mely betekintést adhat a virtuális zöldebb várostervezés felhasználásához.

2.3 CO₂ és a károsanyag kibocsátási rendszer

A napjaink klímahelyzetének egyik kulcsfontosságú gáza a szén-dioxid. E gáz az üvegházhatást okozó gázok közé tartozik, mely hőt vesz fel és csak lassan engedi ki azt, mely felmelegíti a Földet. A nagy szén-dioxid mennyiség káros hatással van az emberekre és a környezetre. Hogy mennyi szén-dioxidot termel egy személygépjármű? [9] Az SMMT New Car CO₂ Report 2019 szerint, egy átlagos új személygépjármű szén-dioxid kibocsátása 124.5g/km az Egyesült Királyságban. A dízel motorok általában 15% - 20%-kal kevesebb szén-dioxidot bocsátanak ki a benzinmotorokhoz képest.

2.3.1 Kibocsátás besorolás rendszere:

[10] Az AIR index kibocsátási besorolása szerint 5 különböző fokozatot különít el. Az AIR Alliance egy nemzetközi nonprofit szervezet, mely személygépjárművezetőknek segít a jármű választásban és karbantartásban. 2017-ben tudósok és egészségügyi szakemberek alapították és céljuk, hogy a CO₂ és a károsanyag kibocsátások káros hatásait ismertessék az emberiséggel. AIR jármű kibocsátási index:

Kibocsátási tartomány	AIR index CO ₂ besorolás	Jelentése
0-95 g/km	A	Alacsony kibocsátás
95-130 g/km	B	Mérséklet kibocsátás
130-180 g/km	C	Magas kibocsátás
180-230 g/km	D	Rendkívül magas kibocsátás
230+ g/km	E	Szélsőséges kibocsátás

Táblázat 1: AIR járművek által kibocsátott CO₂ besorolása [10]

[11] Az EU járműbesorolási rendszere:

Euro Standard	Introduction date		Emission limits		
	New approvals	All new registrations	Petrol NOx	Diesel NOx	Diesel PM
Euro-1	1 July 1992	31 December 1992	0.97g/km*	0.97g/km*	0.14g/km
Euro-2	1 January 1996	1 January 1997	0.5g/km*	0.9g/km* (direct injection)	0.1g/km
Euro-3	1 January 2000	1 January 2001	0.15g/km	0.5g/km	0.5g/km
Euro-4	1 January 2005	1 January 2006	0.08g/km	0.25g/km	0.025g/km
Euro-5	1 September 2009	1 January 2011	0.06g/km	0.08g/km	0.005g/km
Euro-6	1 September 2014	1 September 2015	0.06g/km	0.08g/km	0.0045g/km

*includes NOx and HC

Táblázat 2: Euro járműkategóriák [11]

A vizsgálat során használt gépjárművek besorolása - EU4 kategória (EC2005)

A bevezetés dátuma 2005 január 1 volt, mely magával hozta, hogy a 2006 január 1-től nyilvántartásba vett autók ebbe a besorolásba tartozzanak és ezen határértékeknek feleljenek meg. [11] A MyVehicle.ie által írt 2021-es cikk bemutatja az Euro 4-es kategóriát és a megszorításait, amelyeknek meg kell felelnie a gépjárműnek:

Euro 4 kibocsátás határa (benzin)	Euro 4 kibocsátás határa (dízel)
CO – 1.0 g/km	CO – 0.50 g/km
HC – 0.10 g/km	HC + NOx – 0.30 g/km
NOx – 0.08	NOx – 0.25 g/km
PM – nincs határ	PM – 0.025 g/km

Táblázat 3: Euro 4 járműbesorolás összehasonlítása benzin és dízelmotoros járművek esetén [11]

Mi az a HBEFA: the HandBook of Emission FActors for road transport, magyarul a közúti közlekedés kibocsátási faktorának kézikönyve. A kézikönyveket folyamatosan frissítik és újabbnál újabb verziókat bocsátanak ki. Ez a program a járművek osztálya szerint határozza meg a kibocsátási faktort, például személygépjárművekre és motorkerékpárokra is. A HBEFA-t felhasználják környezeti elemzésekhez, légszennyezéssel és klímával foglalkozó elemzésekhez, közlekedési és logisztikai programok bemeneti adataihoz. A HBEFA3/PC_G_EU4 (PC_gasoline_Euro-4 vagy PC_petrol_Euro-4, amely elnevezése a

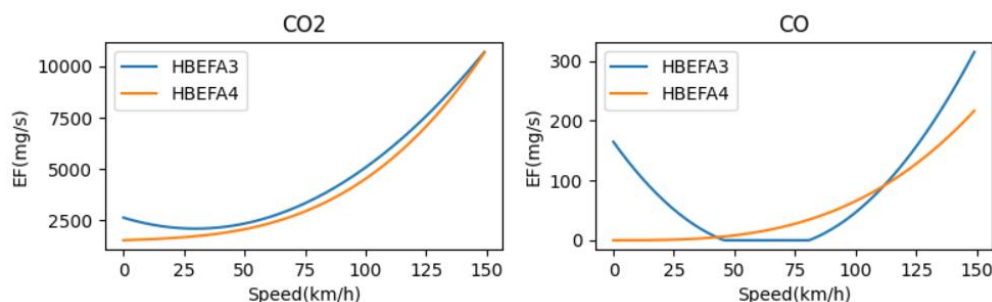
használt verziótól függ) járműtípussal dolgoztam, mely a benzines belsőégésű motoros személygépjármű Euro 4-es besorolással.

[12] HBEFA4 és HBEFA3 alapján kibocsátás faktorának vizsgálata: mérési hiba:

	HBEFA4		HBEFA3	
	CO ₂ hiba	CO hiba	CO ₂ hiba	CO hiba
Dízel	7.46%	20.81%	6.1%	41%
Benzin	4.32%	89.31%	4%	26%

Táblázat 4: HBEFA4 és HBEFA3 összehasonlítása [12]

A HBEFA4-es verzió a legújabb, így érdemes lenne ezt a verziót használni, viszont a dízelmotorral rendelkező járművek esetén a szimuláció a CO₂ érték esetén 7,46%-os és CO esetén pedig 20,81%-os hibával dolgozik. Benzinmotorral rendelkező jármű esetén a program a CO₂ 4,32%-os és a 89,31%-os hibával dolgozik, mely jelentősen nagy. A mérési hiba a HBEFA4-es verzióban meglehetősen nagy, így a HBEFA3 került használatra. [13] Egy github-on talált programrészlettel vizualizálni lehet a két HBEFA közötti eltéréseket.



Ábra 5: HBEFA 3 és HBEFA 4 közötti eltérés és hiba szemléltetése [13]

[14] Az alábbi táblázat az európai országok átlagos személygépjárművek CO₂ szennyezését mutatja, mely a HBEFA hivatalos kimutatása alapján került lekérdezésre. Ez alapján is látszik a kibocsátás csökkenő tendenciája az évek alapján. (Vehkm - Vehicle km – jármű km)

	CO ₂ kibocsátás [g/Vehkm]		
Év	2015	2020	2025
Svájc	197,853	177,746	161,636
Ausztria	169,148	169,318	158,112
Németország	177,814	174,004	163,59

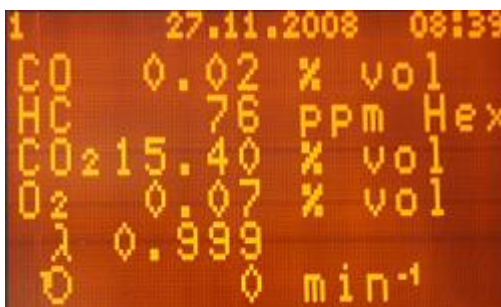
Táblázat 5: HBEFA oldal ország szerinti CO₂ kibocsátás lekérdezése [14]

2.4 Mérőműszerek

A gépjárművek és személygépjárművek kibocsátási adataink méréséhez, a KTI Közlekedéstudományi Intézet Közlekedési légszennyezés weboldalán írja, hogy az [15] „Otto-motorok motordiagnosztikában a kipufogógáz-elemző alapműszernek számít. Egy igazán használható készüléknek az alábbi égéstermék komponenseket kell tudni mérni, ráadásul pontosan:

- CO, szénmonoxid, térfogat százalékban mérjük
- HC, szénhidrogén, ppm-ben mérjük, /parts per million/
- CO₂, széndioxid, térfogatszázalékban kerül kijelzésre
- O₂, oxigén, szintén térfogatszázalékban mérhető
- Lambda-érték (légviszony tényező), ez nem mért, hanem számított érték.”

Az alábbi képen (Ábra 6) egy mérőműszerrel készült elemzést láthatunk egy hibátlanul működő motor adatairól. Az értékek megfelelnek a követelményeknek és a lambda értéke is közel áll az egyhez, mely az ideális érték.



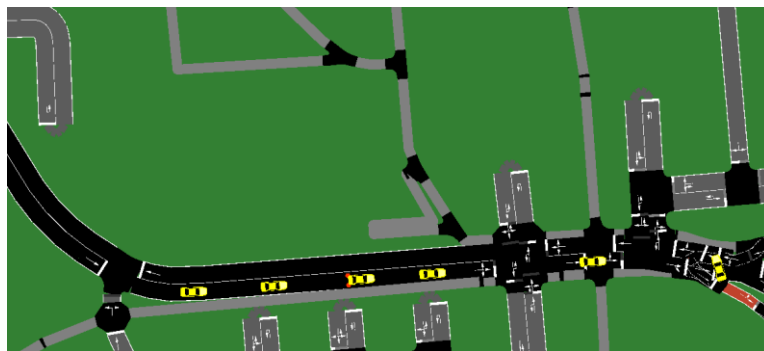
Ábra 6: Mérőműszerrel mért motor kibocsátási adatai [15]

2.4.1 Az infravörös mérés elve

[16] „A kipufogógázok (CO, CO₂, CH) vizsgálata esetén minden gáz más hullámhosszúságú sugárzást nyel el. A készülék alapelve, a gázok különböző energia elnyelésének értékén alapul.”

3. SUMO használata

A használt program, a SUMO, angolul, Simulation Of Urban MObility, vagyis Városi környezet szimuláció, mely egy nyílt forráskódú szoftver. 2001-ben lett kiadva, a Institut für Verkehrssystemtechnik, vagyis a Közlekedési Rendszertechnikai Intézet által, mely a German Aerospace Center alá tartozik. A program egy mikroszkopikus közlekedési szimuláció, mely képes nagyobb hálózatokkal is dolgozni. A program rendelkezik egy hálózat építő és szerkesztő felülettel (netedit.exe), egy internet alapú OpenStreetMap modullal, mellyel internetes térképről kivágott részlet importálható a hálózatba a „netconvert” paranccsal (webwiz.exe). Ez a funkció a hálózatot tökéletesen felépíti, viszont a kanyarodási lehetőségeket, melyekkel a szimulációs járművek navigálnak, nem tudja legenerálni valósághoz közelinek sem. Gyakran nem létező és lehetetlen kanyarodási irány jelekkel látja el az utakat. Az elkészült modell a SUMO (sumogui.exe) alkalmazással indítható. Az elkészült fájl egy „.sumocfg” formátumban lesz, melyet a sumogui.exe programmal lehet futtatni. A szoftver előre elkészített fájlokat importál, ezek által módosítható a hálózat, a járművek és útvonalaik is. A következő képen egy OpenStreetMap által generált térkép került importálásra, amelyen látszik, hogy az OSM a hálózatot még nagyjából helyesen veszi fel, viszont az kanyarodási jeleket, amelyek alapján a járművek navigálnak és haladnak, azokat lehetetlen irányokban veszi fel.



Ábra 7: SUMO szimulációs program működése

Olyan útburkolati jeleket használ és olyan helyeken, ami közúti viszonyok között nem megfelelő és nem megengedett. Ez a gyors hálózat importáló az egyik hibája a szoftvernek. Ha ez a funkció tökéletesen működne, akkor rendkívül sok időt spórolhatna meg az embereknek, akik rendszeresen használják e szoftvert.

A SUMO a „net” fájlt használja a szimuláció futtatása során. Az úthálózat létrehozható és módosítható a netedit.exe programmal, vagy letölthető az OSM oldalán található térkép. Az OSM vagyis az OpenStreetMap nevezetű oldalon egy térképet láthatunk, melyből egy kiválasztott területen lévő hálózatot le is tudunk tölteni. Sajnos, a leképezett térkép elég pontatlan, az utakat elméretezi, rosszul köti össze és a kanyarodási lehetőségeket lehetetlen és valótlan irányokban helyezi, így nem használtam ezt a funkciót. A hálózatunkat a „netedit.exe” programmal tervezhetjük meg, melyhez a járműállományt és az útvonalakat is elkészíthetjük, amelyeket külön fájlokban tárol, mint például „net”, „view”, „add” és „rou”. Az elkészült hálózatunkhoz nem muszáj előre definiált útvonalakat használni, hanem helyette egy előre elkészített szkript segítségével a „python randomTrips.py -n gyor.net.xml -e 20000” parancs beírásával a parancssorban legenerálja az adott hálózatunkra a véletlenszerű útvonalakat. A „sumocfg” kiterjesztésű fájlt módosítva beleírjuk, hogy milyen értékekkel, mennyi ideig fusson a szimuláció, mekkora időléptekkel, én egy másodperces léptekkel dolgoztam. Az output mező megadásával képezhetjük le a kibocsátási adatokat.

```

<?xml version="1.0" encoding="UTF-8"?>
<configuration>
  <input>
    <net-file value="gyor.net.xml" />
    <route-files value="trips.trips.xml" />
  </input>
  <output>
    <emission-output value="emission_out_gyor.xml"/>
  </output>
  <time>
    <begin value="00000" />
    <end value="03600" />
    <step-length value="1" />
  </time>
</configuration>

```

Ábra 8: SUMO konfigurációs fájl tartalma

A SUMO program futtatása után legenerált kibocsátás fájl, az Emission Output, mely az alábbi képen látható.

```

<emission-export>

<timestep time="<TIME_STEP>">

  <vehicle id="<VEHICLE_ID>" eclass="<VEHICLE_EMISSION_CLASS>" CO2="<VEHICLE_CO2>" CO="<VEHICLE_CO>"
  HC="<VEHICLE_HC>" NOx="<VEHICLE_NOX>" PMx="<VEHICLE_PMX>" fuel="<VEHICLE_FUEL>" electricity="<VEHICLE_ELECTRICITY>" noise="<VEHICLE_NOISE>" \
  route="<VEHICLE_ROUTE>" type="<VEHICLE_TYPE>" waiting="<VEHICLE_WAITING>" lane="<VEHICLE_LANE>" \
  pos="<VEHICLE_POS>" speed="<VEHICLE_SPEED>" angle="<VEHICLE_ANGLE>" x="<VEHICLE_POS_X>" y="<VEHICLE_POS_Y>"/>

  ... more vehicles ...

</timestep>

... the next timestep ...

</emission-export>

```

Ábra 9: Szimuláció által készített kibocsátási adatokat tartalmazó fájl

A konfigurációs fájlban egy soros kóddal módosítható, hogy milyen állományokat hozzon létre a program. Ezekből az egyik, a kibocsátás, mely tartalmazza a károsanyag kibocsátás adatait.

A kimeneti adat kétféleképpen generálható le. A konfigurációs fájlba egy sor hozzáadásával, vagy az adott konfigurációs fájlt tartalmazó almappa parancssorába írandó kód által: „sumo -c run.sumocfg --emission-output emissions.xml”.

További kibocsátási lehetőségek beállításához szükséges parancsok:

--emission-output.geo: bekapcsolható a kimeneti adatok WGS84 georeferált koordinátái a hálózatra.

--emission-output.attributes ATTR1,ATTR2,...: korlátozza a kimeneti adatok tulajdonságait. Alapértelmezettnek minden be van kapcsolva, mindent kiír.

--emission-output.precision: a kiírt adatok tizedes pontosságát állítja.

--emission-output.step-scaled: nem másodpercben írja ki a kibocsátási adatokat, hanem szimulált ütemenként.

4. Python, mint programozási nyelv

A nyelvet a holland Guido van Rossum fejlesztette ki, ami 1991-ben jelent meg. Hamar nagy népszerűsége lett és szerte ágazó területeken használják. Szintaxisa rendkívül hasonlít az angol nyelvhez, így könnyen értelmezhető és tanulható. A nyelv hozzáférhető Windows, Linux, macOS és még sok más operációs rendszerben. A python programozási nyelvnek színes felhasználhatósága van. Egészen a webfejlesztéstől, a matematikai egyenleteken és szoftverfejlesztésen át egészen a robotikáig, melyeket modulok segítségével használhatunk.

4.1 A programhoz használt modulok

A kimeneti adatokat a pandas nevezetű modullal kezeltem. A SUMO szimulációból kinyert adatokat egy szövegesen olvasható formátumban kaptuk meg, viszont ezzel a szöveges xml kiterjesztésű fájlal nehéz dolgozni, így erre a problémára először egy parancs kellett, mely átalakítja az eredeti fájlt, és a nekünk szükséges adatokat egy új fájlba másolja. E lépés után csak a lényeges adatok maradnak számunkra.

```
\SUMO\sumo_gyor>python xml2csv.py emission_out_gyor.xml
```

Ábra 10: Parancssor által végzett xml-ről csv-re történő átalakítás

A mappa parancssorába írva „python xml2csv.py 'fájl_neve.xml'” az xml fájlból csv kiterjesztésű fájlt készít. Ez a programrészlet egy előre elkészített szkript, melyet a SUMO

eszközök mappájában találtam meg. Természetesen a szkriptnek és az átalakítandó xml-nek egy almappán belül kell elhelyezkedniük, különben nem tud belőle csv fájlt készíteni.

Az elkészült kibocsátási csv fájl egyik sora, mellyel a későbbiekben dolgoztam:

6.00;125.17;7790.36;0.76;3.34;0.16;233.79;HBEFA3/PC_G_EU4;0.00;2484.75;1;-
699652934#2_0;69.41;40.16;!

```
timestep_time;vehicle_co;vehicle_co2;vehicle_HC;vehicle_NOx;vehicle_Pm;vehicle_angle;vehicle_eClass;vehicle_electricity;vehicle_fuel;vehicle_id;vehicle_lane;vehicle_route;vehicle_speed;vehicle_type;vehicle_waiting;vehicle_x;vehicle_y  
0.000;164.778;2624.722;0.812;1.204;0.066;61.827;HBEFA3/PC_G_EU4;0.000;837.222;0;121040310_0;55.940;5.100;0!var#1;0.000;DEFAULT_VEHTYPE;0.000;3706.696;2077.898  
0.025;164.362;2660.007;0.811;1.219;0.067;61.827;HBEFA3/PC_G_EU4;0.000;848.476;0;121040310_0;67.351;5.102;0!var#1;0.065;DEFAULT_VEHTYPE;0.000;3706.697;2077.899  
0.050;163.954;2695.529;0.810;1.234;0.067;61.827;HBEFA3/PC_G_EU4;0.000;859.805;0;121040310_0;67.391;5.105;0!var#1;0.129;DEFAULT_VEHTYPE;0.000;3706.700;2077.900  
0.075;163.505;2727.886;0.809;1.248;0.068;61.827;HBEFA3/PC_G_EU4;0.000;870.125;0;121040310_0;67.197;5.110;0!var#1;0.193;DEFAULT_VEHTYPE;0.000;3706.704;2077.902
```

Ábra 11: Output kibocsátási adatok a csv fájlban

A képen is látszik, hogy a későbbiekben ezt sem mi, sem a program nem fogja tudni kezelni. Szerencsére a Python sok területen alkalmazható és rengeteg sok különböző modullal rendelkezik az adatkezelés és matematika területén is.

4.1.1 Fájelkezelő modul - Pandas

A Python sokszínű moduljai közül a pandas DataFrame nevezetű részét használtam, mely a meglévő adatokból egy táblázatot csinált, melynek sorai és oszlopai vannak. Ezek után az adatokra már lehet sor és oszlop alapján hivatkozni. A DataFrame táblázatok rendkívül flexibilisek, sokféleképpen lehet az adatokra hivatkozni, mint például oszlop neve szerint, vagy oszlop száma szerint. Én az utóbbit használtam, mivel így csak számokat kell beírni az oszlopok váltogatásához. A sok adatformálás után kezdődhet meg az egyéni felhasználáshoz szükséges adatok kigyűjtése.

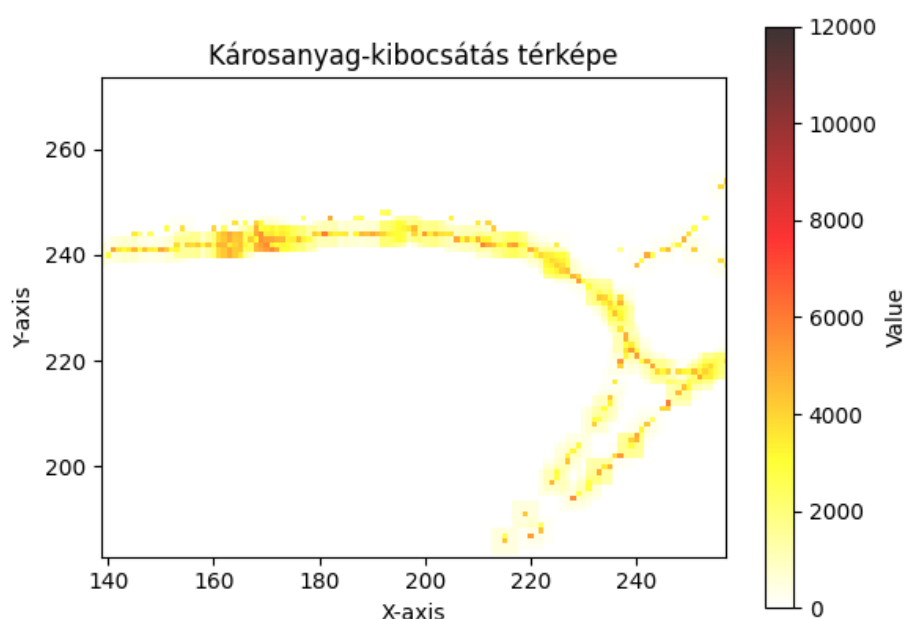
4.1.2 Matematikai modul - Matplotlib - Adatvizualizációs célokra

Az adatok a fájlunkban a következőképpen helyezkednek el a listában: [CO₂_érték, x pozíció, y pozíció]. [17] A lista egy beépített függvény, amivel több elemet tudunk tárolni egyetlen változóban. A programunk még „for” és „if” függvényeket használ. A „for” függvénnyel iterálni lehet egy adott elemet, vagyis ismételten végre lehet hajtani, listát és még sok minden mást, mivel ez az egyik legrugalmasabb függvény mely használata a legegyszerűbb programokhoz is elengedhetetlen.

Az „if” vagyis a „ha” logikai feltételes elágazás függvény, melyet a következőképpen lehet használni [18]:

```
Ha <feltétel> akkor
    <utasítások>
Elágazás vége
```

A matPLOtlib modul nevében benne is van a használt parancs, a PLOT, mely magyarul telek, parcella vagy grafikus helyzetet jelent. Egy egyszerű „for” ciklusba helyezve a listát, a ciklus végigmegy a lista elemein, azokat egyesével megvizsgálva. „If” függvényekkel behatárolva a program megvizsgálja a lista első elemét, a CO₂ értéket, mely, ha egy adott tartományba tartozik, annak egy szint tulajdonít. A színekkel később vizualizálni lehet majd az adatok mértékét és ránézésre is meg lehet mondani, hogy hol mennyi volt a kibocsátás mértéke. Ha az „if” feltétel teljesül, akkor a program plot-olja, vagyis belehelyezi az x pozíció és az y pozíció alapján a koordináta rendszerbe az értéket. A megjelenítés tulajdonságai az „imshow” paranccsal módosíthatók (szín, megjelenített forma, mérete és áttetszősége is állítható). A megjelenítéshez pedig a „matplotlib.show()” parancsot kell használni. Az adatok vizualizációját egy külön megnyíló ablakban látjuk, melyet akár méretezhetünk is. Az adatok meghatározását az x és y tengelyek segítségével szemmel is elvégezhetjük. A tengelycímek és a kép címe is módosítható. A következő ábrát az általam készített program kezdetleges verziójával készítettem, mely Győr, Egyetem tér 1. cím mellett megtalálható körforgalom másához készült.



Ábra 12: Általam készített kezdetleges verzió

4.1.3 Numpy – matematikai és adatkezelési modul

A csomagot 2005-ben fejlesztették ki és neve a „Numerical Python” elnevezésből ered. Ez felelős a mátrixok és tömbök kezeléséért. Megkönnyíti a mátrixokban a keresést, összeadást és a legnagyobb vagy legkisebb elem számának megtalálását. A modul gyorsítja a tömbök használatát, majdnem ötvenszeresére a normális python listánál. E féle csomagok és tömbök az adattudományok területén nagy szerepet kapnak, hisz a gyorsaság és a memóriahasználat rendkívül fontos, ha a programunk hatalmas adatmennyiségekkel dolgozik.

5. Az általam írt program felépítése és működése

5.1 Modulok, csomagok importálása

A Pandas modul az adatok módosítására és elemzésére használt csomag, mellyel a beolvasást és a csv fájl kezelését végeztem. A Matplotlib.pyplot a parcellák feltöltését és az adatok megjelenítését végzi, a Numpy pedig a mátrixok kezeléséért és módosításáért felel.

5.1.1 A csv fájl beolvasása

```
import numpy as np
import pandas as pd
from matplotlib import pyplot as plt

#CSV beolvasása
df = pd.read_csv("C:\\EGYETEM\\SZE\\TDK\\Emission-Sumo\\emissions.csv", sep=';', header=0, usecols=[2, 18, 19])
df.dropna(inplace=True)

# Listába rakjuk a dataframe-et
data = df.values.tolist()
```

Ábra 13: Általam írt program modulok importálása és a kimeneti fájl beolvasása

pd.read_csv: CSV fájlként olvassa be egy dataframe-be, ahol pontosvesszővel vannak elválasztva a cellák. A header pedig az adatok első sorát kiveszi, ez lesz a fejléc, ezekkel a nevekkkel is lehet az oszlopokra hivatkozni. A „usecols” paranccsal a használni kívánt oszlopok indexét kell megadni. Ezután a „dropna” az üres és hibás adatokat kidobálja a fájlból, majd a dataframe elemeit listába rakjuk, így a dataframe egy sora egy lista lesz, így egy nagy listát kapunk, melynek minden eleme egy lista.

5.1.2 A raszter háló méretének meghatározása

„df.iloc[:, 1].max()” Integer féle adattípus helymeghatározása, mely megkeresi a második oszlopban a maximális x koordináta értékét, ez lesz az x tengely határa, majd ezekhez +1-et adunk, hogy biztosan beleférjen az ábrába az összes cella.

5.1.3 Mátrix feltöltése és a mátrix kezelése

A mátrixot először 0-s értékkel töltöttem fel, majd az adott x, y koordinátaához hozzáadtam az értéket. Az eloszlás függvény pedig a következőképpen működik, mely akkor lép életbe, ha az eredeti középső cella értéke meghaladott egy előre megadott értéket, majd az eredeti érték negyede egy kereszt alakba oszlik szét s ez reprezentálja a gáz terjedését.

0	0	0	0	0
0	0	5	0	0
0	5	20	5	0
0	0	5	0	0
0	0	0	0	0

Táblázat 6: Eloszlási függvény működése

E függvény optimalizálásához a CoPilot mesterséges intelligencia segítségét használtam fel. A programkód a 2. függelékben látható.

5.1.4 Megjelenítés

Az ábra megjelenítését a matplotlib csomag „imshow” parancsával végeztem, melynek tulajdonságai módosíthatók. A mátrix beolvasása után a színezés „turbo”-ra van állítva, így a megjelenítés egy hőtérképhez hasonló színvilágot utánoz. Az interpolációhoz a Gauss féle típust használtam, ez végzi a színek elmosását. Az alpha tulajdonsággal módosítható az áttetszőség, az origin pedig az origó helyzetét határozza meg, amely lower esetén a bal alsó sarok. A színskála értékeit a vmin és vmax határozzák meg. A title az ábra címét, míg a xlabel és ylabel pedig a tengelycímket, a colourbar label pedig a színskála címét határozzák meg. A végső megjelenítést, kiírást a „plt.show” parancs végzi.

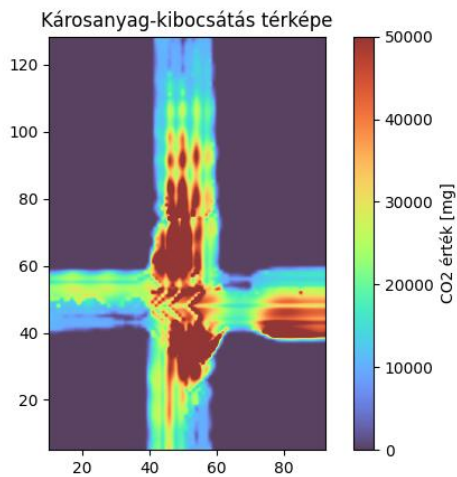
```
plt.imshow(matrix, cmap='turbo', interpolation='gaussian', alpha=0.8, vmin=0, vmax=100000, origin='lower')
plt.colorbar(label='CO2 érték [mg]')
plt.xlabel('')
plt.ylabel('')
plt.title('Károsanyag-kibocsátás térképe')
plt.show()
```

Ábra 14: Az általam készített program ábra megjelenítő kódrészlete

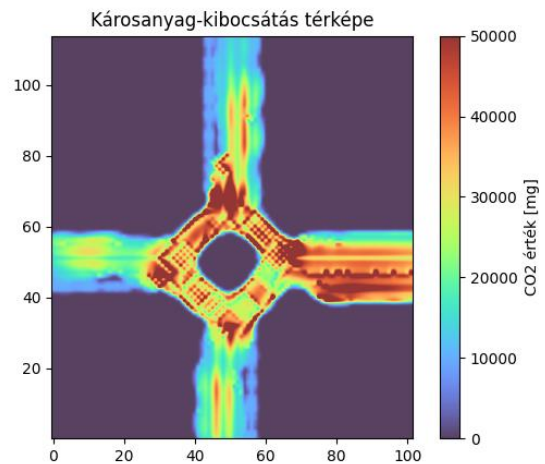
6. Eredmények

6.1 Csomóponti vizsgálat

Két csomóponton végeztem elemzést, a bal oldali kép (Ábra 16) egy egyszerű négyágú kereszteződést mutat, a jobb oldali kép (Ábra 15) pedig egy négy bemeneti ággal rendelkező körforgalmat mutat.



Ábra 16: Négyágú csomópont



Ábra 15: Négyágú körforgalom

A két generált ábra ugyanazzal a járműtípussal és járműadatokkal került importálásra. A szimuláció azonos időmennyiséggel futott le, így minden paramétere megegyezik. A képeket összehasonlítva látható, hogy egy egyszerű négyágú csomópont négy bemeneti ággal környezeti hatása teljesen különböző a négy bemenő sávval rendelkező körforgalmi csomóponttól. Az eredmény szemmel láthatóan eltérő, a négyágú csomópont esetében már a csomópont előtt is torlódás alakult ki, ezáltal magasabb a CO₂ koncentráció is. A körforgalom esetén csak a csomópont bemeneti ágai elején alakult ki nagyobb koncentráció, így ebben az esetben a körforgalom azonos forgalmi adatok esetén kedvezőbb CO₂ kibocsátást eredményezett, így környezeti elemzés szempontjából érdemes ezt választani. Persze egy hálózat szintű elemzés során nem lehet eldönteni csupán egyetlen csomópont vizsgálatából az eredményt, így későbbiekben egy nagyobb mátrix által regionális és városrész szerinti kibocsátást is képes lesz vizsgálni, melyre egy hasonló árnyalatú kibocsátási térkép fog rákerülni, amit az applikációban ki és be lehet majd kapcsolni, attól függően, hogy éppen milyen területet és milyen cél szerint vizsgálunk.

6.2 Város szintű károsanyag kibocsátás vizsgálata

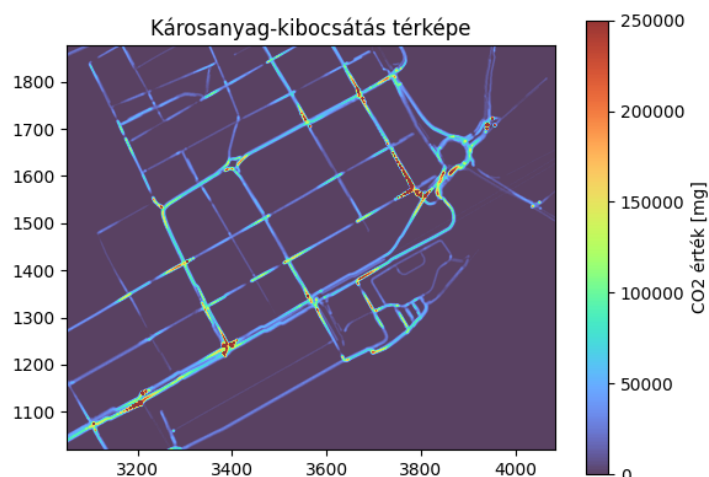
A következő elemzésemhez Győr városának belvárosi részét használtam fel. A szimuláció egy óráig futott, véletlenszerűen generált útvonalakkal, melyet a „randomTrips.py” szkripttel futtattam a hálózaton. A kimeneti adatok fájlmérete rendkívül nagy, még másodperces időléptekben is, viszont a program le tudja futtatni az ekkora méretű fájlokat is. Az ábrán használható közelítés, mely nem torzítja el a kapott eredményt, így nem csak távolról, hanem közelről is meg lehet vizsgálni egyes útszakaszokat. A képről látható, hogy a piros zónák és csomópontok károsanyagkibocsátási szinteket nézve leterheltek. Rendkívül sok útszakasz van, amely fő útvonalként szolgált, így ezek felett keletkezett a legtöbb CO₂. Sok helyen pirosan jelzi a program, vagyis a szén-dioxid kibocsátás több mint 100000 mg volt, mely 100 grammnak felel meg. Ha felemeljük a színskála értékét, megkapjuk, hogy hol volt a legtöbb a kibocsátás. Az alábbi ábrán (Ábra 18) látható a Győri Árkádnál lévő körforgalom, ahol rendkívül nagy mennyiségű koncentráció gyűlt össze, itt 500 grammra tettem a skála tetejét. Ez a kép akár lehet valóságghű is, viszont mivel nem egyesével vittem fel a közúti adatokat, így a véletlenszerűen elhelyezett járművek egy valós életben kevesebbet használt útra helyezett több járművet. Ettől még a program nem hibás, mivel ez csak egy feltevés volt egy valós hálózaton, fiktív adatokkal.



Ábra 17: Győr térképe [19]



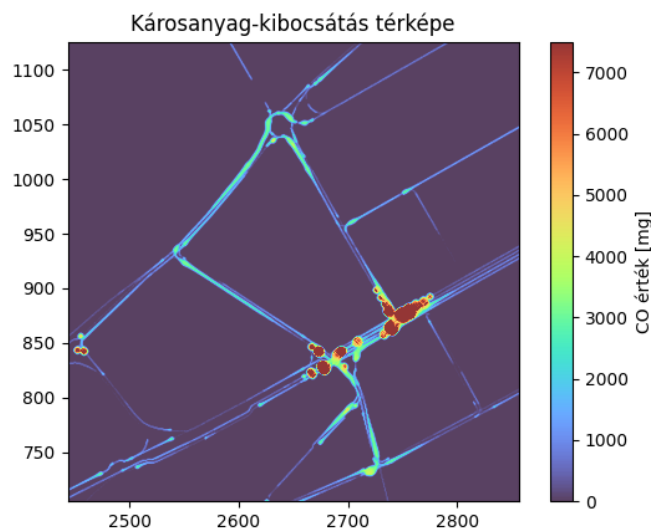
Ábra 18: Hálózat ráközelítve



Ábra 19: győri hálózat CO₂ kibocsátás térképe

A ráközelített képen látható (Ábra 19), hogy a csomópontok körül rendkívül nagy a gázkoncentráció. A tervezett csomópont befolyásolni fogja a kibocsátás mennyiségét, azonban mindenképpen több lesz egy adott csomópont körül, mint egy átlagos útszakaszon.

A szimuláció lefuttatása után a kimeneti fájl akárhányszor újra felhasználható, melyből azonos adatokat kapunk minden egyes alkalommal, mindaddig míg le nem futtatjuk újra a szimulációt. Ha a konfigurációs fájlban nem írjuk át az új kimeneti adatok fájlnevét, az új szimuláció felül fogja írni az előző kibocsátási adatokat. Szerencsére a programomat úgy írtam meg, hogy egy szám megváltoztatásával lehet változtatni a CO és a CO₂ közötti kibocsátási adatokat, viszont az értékek teljesen eltérnek. Bár mindkét érték milligrammban van megadva, viszont egy személygépjármű CO kibocsátása eltér a CO₂ kibocsátásától (CO₂=" 2624.72 " CO=" 164.78"). E paramétereket tudva át kell állítani az értékelosztást és a skála maximális értékét. Ezeket beállítva, a program egy gyönyörű képet ad, melyen rendkívül jól látszanak a CO gócpontok, ahol a gázkoncentráció kellően nagy volt. A képen látható a CO kibocsátás a győri hálózaton (Ábra 20).



Ábra 20: CO kibocsátás szemléltetése a győri hálózati

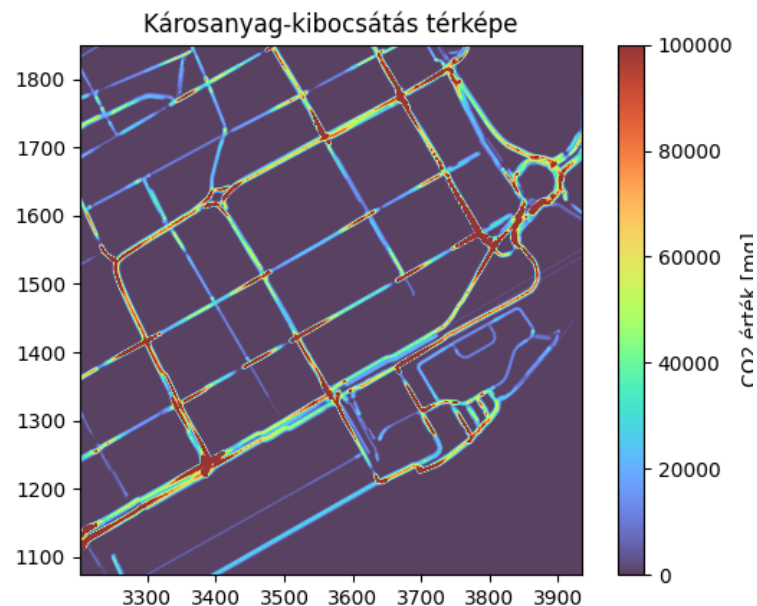
7. A program fejlesztetősége, jövőbeli tervek

A jelenlegi program egy „prototípus”, mely működik viszont nem olyan kifinomultsággal, amelyet el szerettem volna érni. Számos funkciót terveztem bele, melyek közül a legtöbb csak a jövőben tud megvalósulni. Az elsődleges célom a további fejlesztés során egy valóságghű és pontosabb gázkoncentráció eloszlás, amellyel pontosabb elemzéseket lehetne készíteni. Gázok terjedését befolyásoló elemeket vezetnék be, például kezdve egy szél funkcióval, amely segítségével meg lehetne adni, hogy milyen irányból és milyen erősségű szél fújjon. A projekt haladtával fák és épületek elhelyezését is lehetővé tenném, mely szintúgy hatással lenne a gázkoncentrációban. Ezek alapján, a programmal könnyebb lehet környezeti elemzéseket készíteni SUMO-val, ami sok otthoni és vállalati elemző tevékenységét könnyíthetné meg. A felhasználhatósága sokszínű lenne, mint például a szükséges zöld és alacsony kibocsátási területek meghatározásánál, hogy hogyan változna a károsanyagterkép. Egy időskálát is terveznék, amit grafikusán lehetne szabályozni, így a szimuláció futtatása során minden egyes időpillanat visszanézhető lenne, így láthatóvá válna, hogy hogyan alakult melyik területen a kibocsátás. Ezt a későbbiekben mind a grafikus felületről lehetne kezelni. A végleges verzióban egy programcsomagban lehetne futtatni, ami lehetővé tenné, hogy csupán egy ablakból változtatható legyen a gáz típusa, a szél iránya és erőssége, az esetleges környezeti elemek helye és nagysága és az időskála.

8. Konklúzió

A programcsomagot, melyet terveztem a SUMO szimulációs programhoz, egy prototípusa elkészült. A gázterjedés függvénye még nem tökéletes és a jövőben szeretnék még többet foglalkozni egy jobban működő és nagyobb tudományos háttérre fektetett gázterjedési törvény kifejlesztésével. A következő nagy lépést egy kész alkalmazás jelentené, mely a felhasználó számára egy könnyen kezelhető és egyszerűen módosítható felület lenne. Ezen ablakba be lehetne importálni a szimuláció futtatása után létrehozott kibocsátási fájlt, majd a program paramétereit meg lehetne változtatni innen, mint például a skála maximum mennyiségét, színét, a háttér képét, a szétoszlási függvény részleteit és akár még a szimulált időintervallumot is. Az input adataim kitaláltak voltak, mivel egy város szintű szimulációhoz egyszerűbb volt kísérleti fázisban a „randomTrips.py” szkript által legenerált adatokat használni, így ezek az értékek nem közelítik a valóságot, viszont kísérleteimhez tökéletesen működtek.

A későbbiekben szeretnék többet foglalkozni a témával és tovább fejleszteni az általam



Ábra 21: A hálózat terheltsége, CO2 koncentrációja

elkészített programot. Az üvegházhatású gázok és a járművek károsanyagkibocsátása egyre súlyosabb és fontosabb probléma, amivel foglalkoznunk kell. Minden évben erősebbnél erősebb megkorlátozásokat hoznak és egyre szigorúbb követelményeknek kell megfelelniük a gyártóknak és a felhasználóknak. Az általam készített tanulmány és program segíthet a SUMO szimulációs szoftver felhasználóinak, mivel hasonló környezeti kimutatást készítő kódrészlet még nem létezik. [20] Az SUMO hivatalos dokumentációs oldalán hivatkoznak pár darab előre elkészített szkript használatára, melyek használata valamely szinten kötődik a kibocsátáshoz,

viszont ezek közül egy sem foglalkozott még eddig a kibocsátás helyének és mennyiségének vizsgálatával. E problémát is megoldaná a kódrészletem, melyet a jövőben akár a SUMO szoftver által fejlesztett cég számára is továbbíthatnám. Ebben az esetben akár rengeteg ember munkáját könnyíthetném meg, mivel egy egyszerűen elkészíthető vizualizáció hasznos információkat és szemléletes ábrákat tud biztosítani a felhasználónak.

Köszönetet mondok konzulensemnek, Nagy Viktornak a számos javaslatért és szakmai tanácsaiért.

Irodalomjegyzék

- [1] Európai Parlament: Új limit az autók és kisteherautók szén-dioxid-kibocsátására – 2023.08.22. <https://www.europarl.europa.eu/topics/hu/article/20180920STO14027/uj-limit-az-autok-es-kisteherautok-szen-dioxid-kibocsatasara> (Felhasználva: 2024.10.24.)
- [2] NOAA klímáért felelős ügynökség oldala – atmoszferikus szén-dioxid (Írta: Rebecca Lindsey, 2024.04.09.) (Felhasználva: 2024.10.24.) <https://www.climate.gov/news-features/understanding-climate/climate-change-atmospheric-carbon-dioxide> Diagram
- [3] EU TRANSPORT IN FIGURES – STATISTICAL POCKETBOOK 2023 <https://op.europa.eu/en/publication-detail/-/publication/493b2403-7157-11ee-9220-01aa75ed71a1> 139. oldal (Felhasználva: 2024.10.22.)
- [4] Európai parlament honlapja - Új limit az autók és kisteherautók szén-dioxid-kibocsátására (Közzétéve: 2018.09.28.) (Felhasználva: 2024.10.22.) <https://www.europarl.europa.eu/topics/hu/article/20180920STO14027/uj-limit-az-autok-es-kisteherautok-szen-dioxid-kibocsatasara>
- [5] RAC weboldal, járművek kibocsátási besorolása (2024.10.07.) (Letöltve: 2024.10.26.) (Felhasználva: 2024.10.26.) <https://www.rac.co.uk/drive/advice/emissions/euro-emissions-standards/>
- [6] Bath városának a Tiszta levegő zónákról szóló jelentés (2021) (Letöltve: 2024.10.22.) <https://web.archive.org/web/20230827125918/https://beta.bathnes.gov.uk/sites/default/files/Summary%20of%20Bath%27s%20Clean%20Air%20Zone%20Annual%20Report%202021.pdf> f - 3-4. oldal
- [7] Tiszta levegő zónák és alacsony kibocsátási zónák leírása (Felhasználva: 2024.10.27.) <https://www.gov.uk/guidance/driving-in-a-clean-air-zone>
https://en.wikipedia.org/wiki/Clean_Air_Zone
https://en.wikipedia.org/wiki/Low-emission_zone#

[8] ULEZ tábla - https://en.wikipedia.org/wiki/London_low_emission_zone

[9] Society of Motor Manufacturers and Traders kimutatása 2019 (Felhasználva: 2024.10.28.)

<https://www.smmmt.co.uk/wp-content/uploads/SMMT-CO2-Summary-2019.pdf>

[10] AIR index weboldal (Felhasználva: 2024.10.28.)

<https://airindex.com/emissions-ratings/co2-emissions/co2-test-method/>

[11] MyVehicle.ie – EU kibocsátási szabvány (Felhasználva: 2024.10.28.)

<https://www.myvehicle.ie/car-news/euro-1-to-euro-6-emission-standards--a-comprehensive-guide>

[12] SUMO dokumentáció HBEFA4 kibocsátási hiba (Felhasználva: 2024.10.29.)

<https://sumo.dlr.de/docs/Models/Emissions/HBEFA4-based.html>

Emission Factors from the Model PHEM for the HBEFA Version 3

[https://cdn.prod.website-](https://cdn.prod.website-files.com/6207922a2acc01004530a67e/625e7fbf0ee7ac6fc3e12fbd_HBEFA_31_Docu_hot_emissionfactors_PC_LCV_HDV.pdf)

[files.com/6207922a2acc01004530a67e/625e7fbf0ee7ac6fc3e12fbd_HBEFA_31_Docu_hot_emissionfactors_PC_LCV_HDV.pdf](https://cdn.prod.website-files.com/6207922a2acc01004530a67e/625e7fbf0ee7ac6fc3e12fbd_HBEFA_31_Docu_hot_emissionfactors_PC_LCV_HDV.pdf) (Felhasználva: 2024.10.29.)

[13] HBEFA3 és HBEFA4 közötti hiba (Felhasználva: 2024.10.28.)

<https://github.com/eclipse-sumo/sumo/issues/15206> - DIAGRAM

[14] <https://www.hbefa.net/en/software#online-version> (Felhasználva: 2024.11.02.)

[15] Motordiagnosztika - <http://www.kozlekedesilegszennyez.es.hu/tudastar/altalanos-tudnivalok/a-kipufogo-gazok-vizsgalata>

[16] Auto-doktor – emisszió mérés - <https://www.auto-doktor.hu/hatosagi-vizsga/emisszio-meres> (Felhasználva: 2024.11.02.)

[17] W3School weboldala – Python (Felhasználva: 2024.10.27.)

https://www.w3schools.com/python/python_lists.asp

[18] Szent Norbert Gimnázium – programozás alapjai (Letöltve: 2024.10.27)

http://www.szent-norbert.hu/regi/kolmank/12_python_programozas/12_python_10.pdf

[19] Snazzy maps weboldal - Győr térképe – greyscale szín (Felhasználva: 2024.10.29.)

<https://snazzymaps.com/style/287755/grayscale-style>

[20] SUMO kibocsátási adatok dokumentáció

<https://sumo.dlr.de/docs/Simulation/Output/EmissionOutput.html> (Felhasználva: 2024.10.29.)

Mellékletek

1. Melléklet:

A SUMO Output kimeneti adatai

Név	Mértékegység, típus	Leírás
time_step	(szimuláció) s	A megadott érték a „timestep element” alatt
id	Id	A jármű azonosítója
Eclass	ID	Jármű kibocsátásának osztálya
CO ₂	Mg/s	A jármű által kibocsátott CO ₂ mennyisége az adott szimulációs időmértékben
CO	Mg/s	A jármű által kibocsátott CO mennyisége az adott szimulációs időmértékben
HC	Mg/s	A jármű által kibocsátott HC mennyisége az adott szimulációs időmértékben
NO _x	Mg/s	A jármű által kibocsátott NO _x mennyisége az adott szimulációs időmértékben
PM _x	Mg/s	A jármű által kibocsátott PM _x mennyisége az adott szimulációs időmértékben

Fuel	Mg/s	A jármű üzemanyag fogyasztása az adott szimulációs időmértékben
Electricity	Wh/s	A jármű áram fogyasztása az adott szimulációs időmértékben
Noise	dB	A jármű által kiadott zaj az adott szimulációs időmértékben
Route	Id	Útvonal neve
Type	Id	Jármű osztályának neve
Waiting	s	Az idő, melyet a jármű várakozással tölt el
Lane	Id	A sáv neve, ahová a jármű tart
Pos	Méter	A jármű helyzete az adott sávon, melyet a sáv elejétől mér
Speed	m/s	Jármű sebessége
Angle	Fok, szög	Jármű szöge
Pos_x	-	A jármű abszolút x koordinátája
Pos_y	-	A jármű abszolút y koordinátája

2. Melléklet

```
1. import numpy as np
2. import pandas as pd
3. from matplotlib import pyplot as plt
4. from collections import deque
5. import sys
6.
7. # Rekurziós határ beállítása
8. sys.setrecursionlimit(200000)
9.
10. #CSV beolvasása
11. df = pd.read_csv("C:\\EGYETEM\\SZE\\TDK\\Emission-Sumo\\sumo_gyor\\emission_out.csv", sep=';', header=0, usecols=[2,
12. 18, 19])
13.
14. df.dropna(inplace=True)
15.
16. #Listába rakjuk a dataframe-et
17. data = df.values.tolist()
18.
19. #Meghatározzuk a mátrix méretét
20. x_max = int(df.iloc[:, 1].max()) + 1
21. y_max = int(df.iloc[:, 2].max()) + 1
22.
23. #Betöltjük a mátrixot nullákkal
24. matrix = np.zeros((y_max, x_max))
25.
26. #Mátrix feltöltése
27. for value, x, y in data:
28.     x_idx = int(x)
29.     y_idx = int(y)
30.     matrix[y_idx, x_idx] += value
31.
32. #Minimum érték meghatározása
33. min_x = int(df.iloc[:, 1].min())
34. min_y = int(df.iloc[:, 2].min())
35.
36. # Shift coordinates and determine new matrix size
37. shifted_data = [(value, int(x) - min_x, int(y) - min_y) for value, x, y in data]
38. x_max = int(df.iloc[:, 1].max()) - min_x + 1
39. y_max = int(df.iloc[:, 2].max()) - min_y + 1
40.
41. # Függvény a felesleges értékek elosztására
42. def distribute_excess(matrix, threshold, distribute_factor, max_iterations=200000, log_interval=100):
43.     queue = deque()
44.     for y in range(matrix.shape[0]):
45.         for x in range(matrix.shape[1]):
46.             if matrix[y, x] > threshold:
```

```

47. while queue and iterations < max_iterations:
48.     y, x = queue.popleft()
49.     if matrix[y, x] > threshold:
50.         excess = matrix[y, x] - threshold
51.         matrix[y, x] = threshold
52.         for dy, dx in [(-1, 0), (1, 0), (0, -1), (0, 1)]:
53.             ny, nx = y + dy, x + dx
54.             if 0 <= ny < matrix.shape[0] and 0 <= nx < matrix.shape[1]:
55.                 matrix[ny, nx] += excess * distribute_factor
56.                 if matrix[ny, nx] > threshold:
57.                     queue.append((ny, nx))
58.
59.     # Iterációk számának kiírása
60.     if iterations % log_interval == 0:
61.         print(f"Iteration: {iterations}, Queue size: {len(queue)}")
62.         iterations += 1
63.     if iterations == max_iterations:
64.         print("Warning: Maximum iterations reached, distribution may be incomplete.")
65.     else:
66.         print("Distribution completed successfully.")
67.
68. # További felesleges értékek elosztása 10000 felett
69. distribute_excess(matrix, 10000, 0.25)
70.
71. plt.imshow(matrix, cmap='turbo', interpolation='gaussian', alpha=0.8, vmin=0, vmax=100000, origin='lower')
72. plt.colorbar(label='CO2 érték [mg]')
73. plt.xlabel("")
74. plt.ylabel("")
75. plt.title('Károsanyag-kibocsátás térképe')
76. plt.show()

```