Kerékpárút-hálózat gráf alapú elemzései Python modulok alkalmazásával

Digitális Föld c. tárgy projektfeladat  
Szántai Péter Zsolt

# Bevezető

Budapesten az utóbbi időszakban egyre nagyobb figyelmet kap a kerékpáros közlekedés, mint a túlterhelt és a fenntarthatóság szempontjából megkérdőjelezhetőbb gépjármű közlekedés alternatívája. A város történelmi szerkezete a mára tapasztalható gépjárművek sokaságát az utakon nem, vagy csak egyre komolyabb kompromisszumokkal képes kiszolgálni. A közlekedés ezen módja több szempontól sem hatékony és az utak terheltségén túl számos más tényező is közre játszhat a városi közlekedés infrastruktúrájának kialakításában.

Ezen feladat során azonban csak a XI. kerületben jelenleg meglévő kerékpárutak elhelyezkedésére, geometriájára és az egyes rész-hálózatok egymáshoz képesti viszonyára végeztem alap statisztikákat és elemzéseket a Python programozási nyelv és az ehhez szabadon felhasználható nyílt-forráskódú modulok felhasználásával.

# Terület alapú OSM gráf lekérdezés

## Open Street Map

A feladat alapjául szolgáló adatokat az OSM szabad felhasználású térkép-adatbázisából származtattam. Az OSM több felhasználó által közösen szerkesztett térképi alapú téradatbázis, így az általam végzett lekérdezések és elemzések ennek a térképadatbázisnak a minőségével és teljességével erős függésben állnak.

## OSMnx lekérdezés

Erre a térképi adatbázisra alapul a Geoff Boeing által fejlesztett OSMnx modul, melyet lekérdezéseimhez jól tudtam alkalmazni. A modullal OSM alapú leképezéseket végezhetünk el többek között területi alapon. A paraméterként megadható területről, adott hálózattípusú utakat a modul letölti és gráffá alakítja.

place\_name = "XI. kerület, Budapest, Közép-Magyarország, Magyarország" (1)

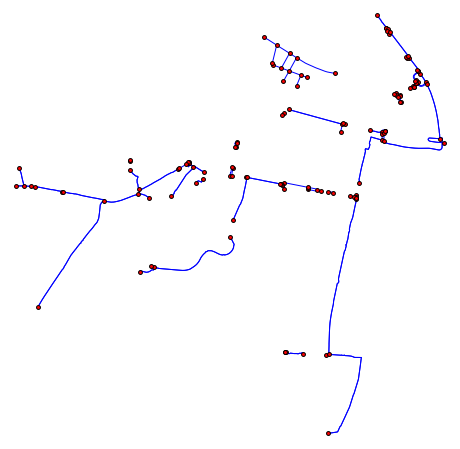
graph = ox.graph\_from\_place(place\_name,network\_type ='bike',simplify=True) (2)

A (2) kódban szereplő graph\_from\_place paramétereinek jelentése összefoglalva: olyan biciklizhető utak lekérdezése a képződő gráf egyszerűsítésével, melyek a megadott területen (1) találhatóak.

## A lekérdezés pontosítása szűréssel

Önmagában (2) lekérdezés nem vezetett a várt eredményre. Mint említettem ez a lekérdezés minden olyan utat tartalmaz, ami az adatok alapjául szolgáló OSM adatbázisban úgy van definiálva, hogy azon kerékpárral közlekedni lehet. Ez a meghatározás nem feltétlenül csak a valódi kerékpárutakra vonatkozhat. Az OSM-en található térképi elemek több különböző taggel lehetnek ellátva, ezekkel azonosíthatók az adott elemek tulajdonságai. Számos különböző tulajdonságot kapcsolhatunk így a térképi elemekhez, utak esetén például azok típusát (pl. „highway”, „cycleway”), a maximális sebességet, az út állapotát stb.

Az OSM értelmezésében a valósághoz hasonlóan többféle kerékpárút létezik (pl. buszsávban haladó, úttestre felfestet, különálló kerékpárút stb.). A valóságot megközelítő önálló kerékpárhálózat előállításához először is minden kerékpározható utat egyszerűsítés (simplify) nélkül kérdeztem le. Az így előálló gráfhálózat éleit azok attribútum adatai alapján leválogattam. Egy változóba kerültek azok a kerékpárutak (gráfélek) melyek nem cycleway vagy highway=cycleway attribútummal voltak ellátva. Az így szűrt éleket töröltem, ezáltal elvben minden olyan él maradt meg amely önálló kerékpárút. A szabadon maradt gráfcsúcsokat szintén eltávolítottam a hálózatból, majd elvégeztem a hálózat egyszerűsítését.



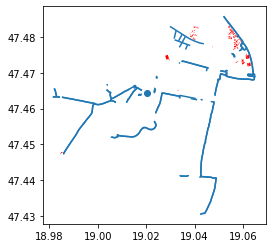
a. ábra XI. kerület kerékpárhálózata gráfként

A hálózatok grafikus megjelenítésére is alkalmas az OSMnx modul (a. ábra). Az ábrán az átláthatóság érdekében a gráfhálózat élei kékkel míg a csúcsok pirossal szerepelnek.

Nagyon szembetűnő a hálózat töredezettsége. A hálózatról kijelenthető, hogy sok kisebb különálló egységből tevődik össze, melyek között helyenként igen jelentős arányban hiányoznak összekötő elemek. Az eredmény nem feltétlenül egyezik a természetben tapasztalható állapottal. Könnyen lehet, hogy már a kiinduló adatforrás (OSM) hiányos, vagy csupán a tagek nincsenek minden esetben helyesen felvételezve. Minden esetre a továbbiakban ezen a rendkívül tagolt hálózaton végeztem el az elemzéseimet.

## Point of Interest

A hálózatunk területéről lekérdezhetünk ún. POI-kat is. Ezek lehetnek bármilyen a felhasználás szempontjából lényeges objektumok (pl. egyetemek, vasútállomások, éttermek). Ezeket a POI-kat a hálózattal való közös ábrázolásához az éleket GeoDataframe (GDF) formátumba kell konvertálni, ami az OSMnx saját függvényével könnyen elvégezhető.



b. ábra XI. kerületi kerékpárutak (kék) és egyetemei épületek (piros)

Ezeket példaként útvonal optimalizáláshoz, legrövidebb vagy leggyorsabb utak szelekciójához használhatjuk.

## Gráf vetítése metrikus rendszerbe

A későbbi statisztikák és számítások helyes elvégzéséhez szükséges volt az egész hálózatot egy metrikus vetületi rendszerbe vetíteni. Célszerűen a Magyarországon gyakran alkalmazott HD72/EOV rendszerbe vetítettem (EPSG:23700).

graph\_proj = ox.project\_graph(graph, crs=23700) (3)

A (3) kódsor függvénye az EPSG szám alapján felismeri a kiválasztott vetületi rendszert (alapértelmezése az UTM zónarendszerbe vetít). A vetítés után – mint korábban is – elvégeztem az élek és a csúcsok leválogatását. A helyesen beállított projekció ellenőrizhető az edges\_proj.crs utasítással.

<Projected CRS: EPSG:23700>

Name: HD72 / EOV

Axis Info [cartesian]:

- Y[east]: Easting (metre)

- X[north]: Northing (metre)

Area of Use:

- name: Hungary

- bounds: (16.11, 45.74, 22.9, 48.58)

Coordinate Operation:

- name: Egyseges Orszagos Vetuleti

- method: Hotine Oblique Mercator (variant B)

Datum: Hungarian Datum 1972

- Ellipsoid: GRS 1967

- Prime Meridian: Greenwich

# Hálózati elemzések

## OSMnx statisztikák

A lekérdezett és létrehozott gráfon az OSMnx függvényeivel is lehet alapvető vagy akár részletes statisztikákat készíteni.

### Alapstatisztikák

A hálózat mindennemű átalakítása nélkül egy egyszerű kódsorral elkészíthetünk egy alapstatisztikát. Ez a statisztika a legalapvetőbb információkat tartalmazza és egy könyvtár (dictionary) típusú változóba rögzíthető. Ilyen módon információkat nyerünk pl. a csúcsok és élek számáról, a metszések számáról, az élek összesített hosszáról stb.

{'n': 136,

'm': 190,

'k\_avg': 2.7941176470588234,

'intersection\_count': 6,

'streets\_per\_node\_avg': 0.19117647058823528,

'streets\_per\_node\_counts': {0: 0, 1: 7, 2: 0, 3: 5, 4: 1},

'streets\_per\_node\_proportion': {0: 0.0,

1: 0.051470588235294115,

2: 0.0,

3: 0.03676470588235294,

4: 0.007352941176470588},

'edge\_length\_total': 46107.51599999996,

'edge\_length\_avg': 242.67113684210506,

'street\_length\_total': 26067.309,

'street\_length\_avg': 208.538472,

'street\_segments\_count': 125,

'node\_density\_km': None,

'intersection\_density\_km': None,

'edge\_density\_km': None,

'street\_density\_km': None,

'circuity\_avg': 1.1172671676152197,

'self\_loop\_proportion': 0.0,

'clean\_intersection\_count': None,

'clean\_intersection\_density\_km': None}

### Kibővített statisztikák

Ezek az alapok kibővíthetőek, de ehhez definiálnunk kell a befoglaló területet. A befoglaló terület minden lekérdezett gráfélt (és csúcsot) tartalmazó konvex, zárt poligon.

Egyes statisztikák értelmezéséhez (pl. utak sűrűsége) elengedhetetlen ennek a befoglaló területnek az ismerete. Az így kapott statisztika nagyos sok információt ad át ömlesztve a hálózatról.

A képen szöveg, boríték, névjegykártya látható

Automatikusan generált leírás

c. ábra XI. kerületi kerékpárutak befoglaló területe

Ezek értelmezése és felhasználása további feldolgozást és gráfelméleti ismereteket igényel, viszont akadnak egyszerűen kiolvasható értékek is mint például az élek és csúcsok száma, egy kilométerre vonatkoztatott sűrűségégek (csúcsok, élek, metszések, utak).

n 136

m 190

k\_avg 2.79412

intersection\_count 6

streets\_per\_node\_avg 0.191176

streets\_per\_node\_counts {0: 0, 1: 7, 2: 0, 3: 5, 4: 1}

streets\_per\_node\_proportion {0: 0.0, 1: 0.051470588235294115, 2: 0.0, 3: 0...

edge\_length\_total 46107.5

edge\_length\_avg 242.671

street\_length\_total 26067.3

street\_length\_avg 208.538

street\_segments\_count 125

node\_density\_km 5.47119

intersection\_density\_km 0.241376

edge\_density\_km 1854.88

street\_density\_km 1048.67

circuity\_avg 4.07377e-05

self\_loop\_proportion 0

clean\_intersection\_count None

clean\_intersection\_density\_km None

avg\_neighbor\_degree {84663800: 0.0, 215402640: 3.0, 223746850: 0.0...

avg\_neighbor\_degree\_avg 1.64583

avg\_weighted\_neighbor\_degree {84663800: 0.0, 215402640: 0.01748200809999708...

avg\_weighted\_neighbor\_degree\_avg 0.0443386

degree\_centrality {84663800: 0.007407407407407408, 215402640: 0....

degree\_centrality\_avg 0.0206972

clustering\_coefficient {84663800: 0, 215402640: 0, 223746850: 0, 2626...

clustering\_coefficient\_avg 0.00612745

clustering\_coefficient\_weighted {84663800: 0, 215402640: 0, 223746850: 0, 2626...

clustering\_coefficient\_weighted\_avg 8.39872e-06

pagerank {84663800: 0.0015173085446425216, 215402640: 0...

pagerank\_max\_node 1178769878

pagerank\_max 0.030679

pagerank\_min\_node 84663800

pagerank\_min 0.00151731

eccentricity {2281223169: 3049.254, 3007250306: 4124.106, 4...

diameter 4124.11

radius 2091.15

center [1178769878]

periphery [3007250306, 3325126123]

closeness\_centrality {84663800: 0.0, 215402640: 7.032158090592992e-...

closeness\_centrality\_avg 0.00013043

dtype: object

Az ilyen részletes adatok későbbi felhasználásra és kimutatásokra is alkalmasak lehetnek azonban a Python programnyelvhez számos céleszköz található, melyek direktebb elemzéseket tesznek lehetővé (pl. NetworkX).

# A projekthez tartozó fájlok

Mivel az OSMnx modul graph\_from\_place függvénye letölti a szükséges kiinduló adatokat külső forrásra nem volt szükség.

A projekt teljes kódsora az alábbi linken érhető el: <https://github.com/szpeterzs/digitalearth>

# Források és felhasznált modulok

Open Street Map – <https://overpass-turbo.eu/>

Nominatim – <https://nominatim.openstreetmap.org/ui/search.html>

Overpass Turbo – <https://overpass-turbo.eu/>

Geoff Boeing – <https://geoffboeing.com/>

Automating GIS-processes – <https://autogis-site.readthedocs.io/en/latest/>

## Python modulok

OSMnx – <https://github.com/gboeing/osmnx>

NetworkX – <https://networkx.org/>

Matplotlib – <https://matplotlib.org/>