

Épületgépészet Kiadó Kft.

1111 Budapest, Műgyetem rkp. 3. III. em. 81.

Telefon/fax: 201- 2562

www.epgeponline.hu e-mail: info@epgeplap.hu

Befogadó nyilatkozat

Budapest, 2023. június 30.

Kocsis Kende

gépészmérnök MSc hallgató úrnak

Ezúton igazoljuk, hogy az alábbi cikket szerkesztőségünk a közlésre átvette:

Kocsis Kende, Kövendi József:

Budapest második hulladékhasznosító művének logisztikai értékelése

A cikket lektorálásra megküldtük dr. Szűcs Botond egyetemi adjunktus úrnak.

A lektori vélemény szerint a cikk közölhető.

A lektori vélemény alapján a cikk nyomdai előkészítését megkezdjük.

A cikk a Magyar Épületgépészet (HU ISSN 1215 9913) 2023 július-augusztusi összevont számában jelenik meg.

Budapest, 2023. június 30.



Dr. Barna Lajos

Főszerkesztő

Budapest második hulladékhasznosító művének logisztikai értékelése

Kocsis Kende, Kövendi József

Absztrakt

Budapesten évente mintegy 680-700 000 tonna települési szilárd hulladék keletkezik, amelynek kevesebb mint két-harmadát hasznosítják újra vagy hasznosítják energetikai célokra, a többi a hulladéklerakóban végzi. Ennek a környezeti és logisztikai problémának a leküzdésére javasolták egy második hulladékhasznosító üzem telepítését a város déli részén. A hulladékból energiát előállító erőművek (WtE-erőművek) esetében a tüzelőanyag-fogyasztáshoz kapcsolódó egyetlen költség a szállítási költség, mivel a városi vezetés gazdasági támogatást nyújt a hulladék ártalmatlanításához. Mivel a szállítás nagyban befolyásolja az üzemeltetési költségeket, a szállítási útvonalak logisztikai optimalizálása közvetlen költségmegtakarítást ígér. Ebben a tanulmányban a Dél-Budapesti terület logisztikai optimalizálását képelemzés és logikai alapú algoritmus-programozás segítségével végeztük el. Ennek eredményeként 230 000 kilogramm települési szilárd hulladék (MSW) optimális szállítását oldottuk meg, ami havi 4 835,2 km utazási távolság csökkenést eredményezett. Ez az érték 2 894 351 Ft költség, 9 459,6 kg CO₂ és 45,3 kg NO_x kibocsátás csökkenést jelent a városi területeken.

Abstract

The city of Budapest produces approximately 680-700 000 tonnes of municipal solid waste every year, of which less than 2/3 is being recycled or used for energetic purposes, the rest ends up in a landfill. To combat this environmental and logistical problem the installation of a second waste incineration plant has been proposed in the south of the city. The only cost associated with fuel consumption in the case of waste to energy powerplants (WtE plants) is the transport cost, as the city council provides economic support for waste disposal. Since the transportation has a huge influence on the cost of operation, logistical optimisation of the transport routes promises a direct impact on cost savings.

In this study the logistical optimisation of the Southern Budapest area was carried out using pixel analysis and logic based algorithm programming.



Kocsis Kende

Budapesti Műszaki és Gazdaságtudományi Egyetem, Gépészmérnöki Kar, Épületgépészeti és Gépészeti Eljárástechnika Tanszék
kende.kocsis@gmail.com



Kövendi József

Pázmány Péter Katolikus Egyetem
Információs Technológiai és Bionikai Kar

As a result the optimal transport of 230 000 kilograms of municipal solid waste (MSW) was solved resulting in a 4 835,2 km monthly travel distance reduction. This value can be translated to 7 823 € cost, 9 459,6 kg CO₂ and 45,3 kg NO_x emissions reduction in the urban areas.

Kulcsszavak: települési hulladékgyűjtés, útvonaloptimalizálás, hulladékból energia előállítása, fenntarthatóság, jármű útvonal optimalizáció programozás

1. Bevezetés

Jelen tanulmányban az előzetes tervek alapján az Ócsai út és az M0-as autópálya kereszteződéséhez telepítendő HUHA 2 hulladékhasznosító erőmű hulladékbeszállításának optimalizálásával foglalkozunk. A hulladékhasznosító erőművek esetében kiemelkedő jelentősége van a hulladékszállítás költségének, mivel a hulladék megsemmisítéséért a budapesti önkormányzattól 2017-es adatok szerint 8 446 Ft/tonna díjat kapnak. Ezek alapján a tüzelőanyag egyetlen költsége maga a beszállítási költség, amelynek csökkentésével direkt módon lehet befolyásolni az üzemelési költséget és ezáltal a hulladékhasznosító erőmű versenyképességét növelni a földgáztüzelésű energiatermeléssel szemben.

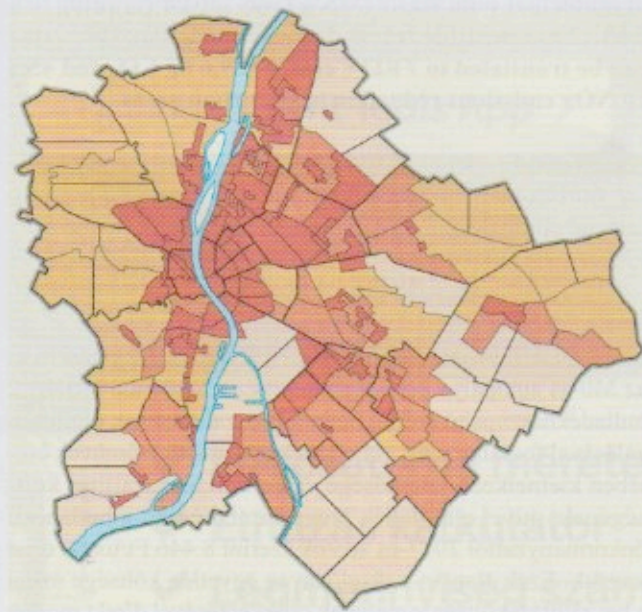
A beszállításhoz szükséges anyagi költségek és a hozzá rendelhető kibocsátási értékek csökkentésével versenyképesebbé válhat a hulladékhasznosítás, mint energiatermelő erőforrás és lecsökkenthetjük Budapest távhőrendszerének és elektromos ellátásának földgáz függését, valamint nem utolsósorban redukálhatjuk a beszállítási tevékenység során keletkező környezeti szennyezést.

2. Módszertan

2.1. A keletkező hulladékmennyiség meghatározása

A szállítási útvonalak optimalizálása érdekében egy egyszerűsített megközelítést alkalmaztunk, amely során Budapestet zónákra bontottuk fel, majd minden egyes zóna számára egy központi pontot határoztunk meg, ahonnan a hulladék begyűjtésre kerül. A modell pontosságának növelése érdekében a meghatározott gyűjtési zónákat a kerületekhez képest jóval kisebb méretűvé tettük, így összesen 182 zónát kaptunk. A modell szerint a hulladékgyűjtő járművek a javasolt üzemtől indulnak, a lehető legrövidebb úton haladva a célzóna kijelölt középpontjáig. A középpont elérését követően a kapacitásuknak megfelelő mennyiségű hulladékot összegyűjtik és visszaszállítják a WtE-üzembe. Ez az eseménylánc addig folytatódik, amíg a WtE-üzem szilárdhulladék-kapacitása el nem éri a célpontot.

A begyűjtési útvonalak meghatározásához helyalapú TSZH termelőési adatbázisra van szükség. A tanulmányhoz szükséges adatok beszerzése érdekében felvettük a kapcsolatot a budapesti FKF Zrt-vel, sajnos eredménytelenül, mivel a tanulmány készítésének időpontjában ilyen adatbázissal nem rendelkeztek. Ezen információs hiányosság leküzdésére alternatív módszert választottunk: képi alapú információ kinyerést végeztünk a népsűrűségi térképekből képfeldolgozással.

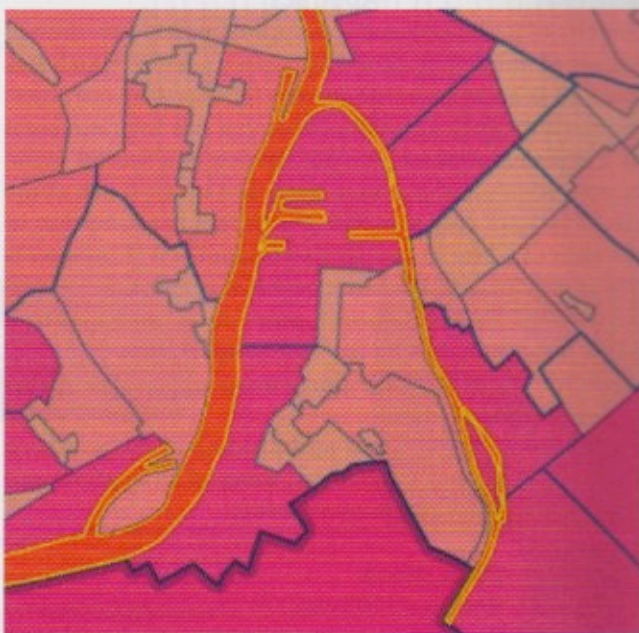


1. ábra. Budapest népsűrűségi adatai finomított zónákkal [1]

Az 1. ábra az elemzési módszerben alkalmazott népsűrűségi térképet mutatja. A képfeldolgozásra azért volt szükség, mert a térkép nem tartalmazott az egyes zónák méretét részletező információkat. Az ábrák szín- és térbeli adatait használtuk fel a terület, a népsűrűség és így a jelzett zónák népességének meghatározására. Magyarországon az Európai Parlament által közzétett adatok szerint évente 381 kg hulladék keletkezik egy főre vetítve [2]. A népsűrűség és a számított zóna területadatok alapján meghatározható az adott zóna lakosság száma, amely alapján meghatározható az övezetekben keletkező települési szilárd hulladék mennyisége.

Az elemzésért felelős algoritmus Python nyelven íródott. Ezt a programozási nyelvet használtuk a tanulmány során minden algoritmus esetében. A program alapvető feladata a zónák körvonalainak felismerése és elkülönítése volt a képen, területük mérése, e területek elkülönítése önálló grafikus fájlakká, majd az egyes területek geometriai középpontjainak meghatározása. Az első lépés az, hogy az elemzendő képet betápláljuk a programba. Ez a kép azonban még mindig sok olyan információt tartalmaz, amely zajnak vagy zavaró hatásnak minősül és megakadályozza, hogy az algoritmus pontosan meghatározza a zónák körvonalait. Ennek elkerülése érdekében a következő módszert alkalmaztuk: először a kép színezési sémáját RGB színmodellről HSV színmodellre alakítottuk át. Ezzel a módszerrel az R (Red), G (Green) és B (Blue) helyett a színesatornák H (Hue), S (Saturation) és V (Value) értéket vesznek fel.

A HSV (hue, saturation, value; más néven HSB, hue, saturation, brightness) az RGB színmodell alternatív ábrázolása, amelyet az 1970-es években terveztek számítógépes grafikusok, hogy jobban megfeleljen az emberi látás színérzékelési tulajdonságainak. Lehetővé teszi, hogy jobban meg tudjuk különböztetni a kép képpontjainak intenzitását, az eredmények az 2. ábrán láthatók. Az átalakítás nagyban leegyszerűsíti a szűrési folyamatot, mivel a HSV színösszetételben a színadatokat csak egy csatorna, a V tartalmazza, nem pedig három csatorna, mint az RGB esetében.



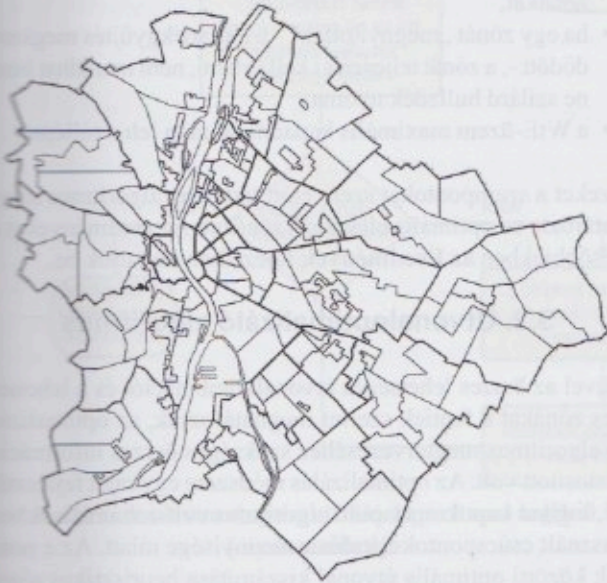
2. ábra. A HSV színesatornába konvertált térkép (felül) és egy közeli kép a Dunával (alul)

Az intenzitáskülönbségek alapján a megfelelő határértékek megválasztásával a Duna szakaszai elkülöníthetők a kép többi részétől, így a folyó nem jelenik meg a térkép zónális felismerésében. A felső kinagyított képen megfigyelhetők a

Duna és a környező területek közötti éles kontúrok. A kiválasztott Duna-képet egy maszk létrehozásával különítjük el. Ez a maszk később a folyó területének elkülönítésére szolgál.

A bemeneti képet szürkeárnyaltos képpé, azaz színes képből fekete-fehér képpé alakítottuk át, hogy az algoritmust ne zavarják a színek. A Duna területének korábban létrehozott, szintén fekete-fehér maszkját erre a képre helyeztük rá, megakadályozva a folyó zónaként való besorolását. A kontúrvonalak további megkülönböztetése érdekében a kép hisztogramját megnyújtottuk, megváltoztattuk oly módon, hogy a sötét tónusok tovább sötétedjenek, a világosak pedig tovább világosodjanak.

Ezt követően a képet küszöbértékelésnek vetettük alá, amelynek eredményeképpen a szürke minden árnyalata eltűnt, kivéve a legsötétebb színt, amely a zónák kontúrvonalait jelentette. Az eredmény a 3. ábrán látható.



3. ábra. A zónák körvonalai a kiküszöbölést követően

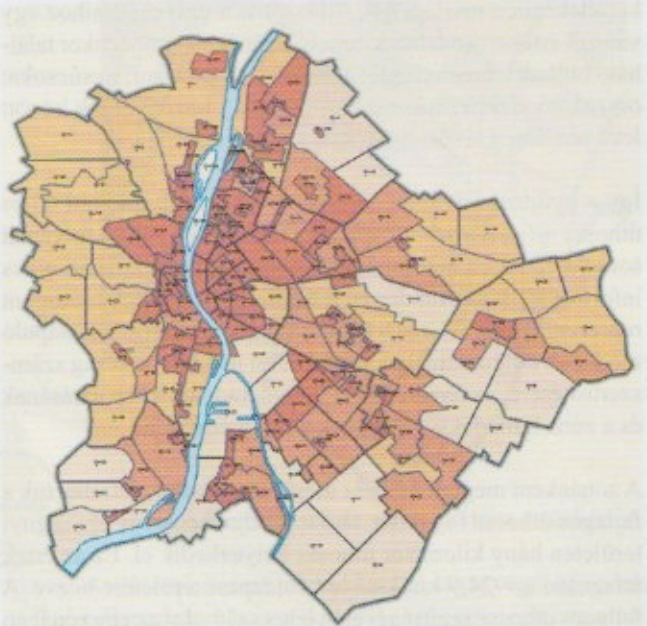
Ezután a kontúrok meghatározásával elkülöníthetjük az általuk körülvevett zónákat. Minden egyes zónát a pontos terület és geometriai középpont adatokkal azonosítottunk, illetve a zóna területének nagysága alapján ID azonosítószámot rendelünk hozzá. A területet a képen található összes pixelszám és a zónákon belül elhelyezkedő pixelek száma közötti összefüggés alapján határoztuk meg a kifejezetten Budapestre számított pixel/km² arányossági tényező segítségével az (1) összefüggés szerint.

$$A_{\text{zóna}} = \frac{A_{\text{zóna,pixel}} \cdot A_{\text{Bp}}}{A_{\text{Bp,pixel}}}, \quad (1)$$

ahol

- $A_{\text{zóna}}$ – a keresett zóna területe [km²]
- $A_{\text{kerület,pixel}}$ – a keresett terület pixelszáma [-]
- A_{Bp} – Budapest területe [km²]
- $A_{\text{Bp,pixel}}$ – Budapest területén lévő pixelek száma [-]

Az azonosított zónákat az eredeti térképre visszamaszkoltuk, így kaptuk meg a 4. ábrát.



4. ábra. A képelemzés eredménye: Budapest az összes azonosított zónával és geometriai középponttal

A városon belül 182 övezetet azonosítottunk. Minden egyes zónára kiszámítható a keletkező hulladék mennyisége a népsűrűség, a számított zónaterületek és a magyarországi fajlagos hulladéktermelési index alapján, amely 381 kg (szilárd hulladék)/fő/év. A számítási módszert a (2) összefüggés mutatja.

$$m_{\text{TSZH}} = f_{\text{nép}} \cdot A_{\text{kerület}} \cdot \dot{m}_{\text{TSZH, fajlagos}}, \quad (2)$$

ahol

- m_{TSZH} – heti TSZH termelés [kg/hét]
- $f_{\text{nép}}$ – népsűrűség [fő/km²]
- $A_{\text{kerület}}$ – a zóna területe [km²]
- $\dot{m}_{\text{TSZH, fajlagos}}$ – fajlagos TSZH termelés [kg/fő/év]

2.2. A TSZH begyűjtési útvonalak meghatározása

A középpontokhoz földrajzi koordinátákat társítottunk, hogy a későbbiekben lehetővé tegyük az útvonal-szimulációkat. Ezeket a középpontokat hipotetikus gyűjtőpontokként határoztuk meg az egyes zónák számára. Ezt követően a 182 rendelkezésre álló középpontból csak 100-at használtunk fel a szimulációkban, mivel a többi 82 zónát a már meglévő, Észak-Budapesten található HUHA 1 hulladékhasznosító erőmű látja el.

A koordinátákhoz tartozó 100 középpontból egy teljes gráfot építettünk fel, hogy az útvonaloptimalizáló algoritmus számára lehetővé tegyük a szabad útvonalválasztást. A gráf 100 csúccsal és így 9900 éllel rendelkezett, mivel a valós életben az A → B útvonal a legtöbbszőr nem egyenlő a B → A útvonallal az útlezárások, az egyirányú utak és az esetleges forgalmi dugók miatt, ami jobb alternatív útvonalakat eredményez. Egy algoritmust fejlesztettünk ki a középpontok között lévő 9900 útvonal szimulálására egy térképészeti adatbázis-program segítségével. Ezen eredmények alapján a grafikont

kétféleképpen mérlegeltük. Először is a gráf csúcsaihoz egy változó értéket rendeltünk, amely a zónákban mindenkor található hulladék mennyiségét ábrázolja. Másodsor, a csúcsokat összekötő élekhez a korábban szimulált középpontok között lévő tényleges távolságértékeket rendeltük.

Így a gyűjtési folyamat során a zónák között megtett teljes úthossz meghatározható volt, de a tényleges gyűjtési folyamat során a zónákban belül megtett úthosszra vonatkozóan nincs információ, mivel minden egyes zónát csak egy-egy csúcspont reprezentál. E probléma leküzdésére egy arányszámon alapuló módszert dolgoztunk ki a zónán belül megtett távolság számszerűsítésére, a szeméteskocsi fennmaradó üres kapacitásának és a zóna hulladéksűrűségének a függvényében.

A zónánként megtett úthossz meghatározásához létrehoztuk a fajlagos úthossz fogalmát, ami megadja, hogy egy egységnyi területen hány kilométer úthossz helyezkedik el. Ez az érték átlagosan $a = 24,9 \text{ km/km}^2$ lett Budapest területére nézve. A fajlagos úthossz segítségével ki lehet számolni az egy zónában található úthossz értékét. Ezen érték segítségével fogjuk tudni kiszámolni az egyes zónákban megtett út hosszát a begyűjtött hulladékmennyiség függvényében:

$$l = A_{\text{kerület}} \cdot a, \quad (3)$$

ahol

$$l - \text{úthossz [km]} \\ a - \text{fajlagos úthossz [km/km}^2\text{]}$$

Ezt a számítást azért alkalmazhattuk, mert feltételeztük, hogy az összes keletkező hulladék begyűjtéséhez a járműveknek végig kell haladnia az összes utca teljes hosszán. Ezzel a számítási móddal a sűrűn lakott területek esetén jóval kevesebb úthosszt kell bejárni a szállító járműnek egy egységnyi hulladék felvételéhez, mint egy gyéren lakott területen, ami megfelel a való életben tapasztalt helyzetnek.

Ennek vizualizálására létrehoztuk a lineáris TSZH sűrűséget, ami megadja, hogy mekkora úthosszt kell bejárni a szállító járműnek egy egységnyi hulladékmennyiség felvételéhez:

$$f_{\text{TSZH}} = \frac{m_{\text{TSZH}}}{l}, \quad (4)$$

ahol

$$f_{\text{TSZH}} - \text{lineáris TSZH sűrűség [kg/km]}$$

Az úthosszak azonosításával meg tudtuk határozni, hogy egy zóna hulladék-kiürítéséhez hány liter üzemanyagra van szükségünk, figyelembe véve, hogy a szemétszállító járművek átlagos fogyasztása 73 l/100 km . [3]

Az esetek túlnyomó részében egyetlen jármű azonban nem képes egy zóna kiürítésére, emiatt a 8 tonnányi hulladék szállítóképességre kell normálni a fogyasztást.

$$v_{\text{kocsi}} = \frac{k}{f_{\text{TSZH}} \cdot l} \cdot \dot{v} \cdot \frac{l}{100}, \quad (5)$$

ahol

$$v_{\text{kocsi}} - \text{egy jármű fogyasztása 8 tonna kapacitás esetén [l]} \\ k - \text{szállítóképesség [kg], üres állapotban 8000 kg} \\ \dot{v} - \text{fajlagos üzemanyag fogyasztás [l/100 km]}$$

3. Az optimalizálás logikája

3.1. Az ellátott zónák meghatározása

Az elemzett WtE-erőmű maximális kapacitása évi 230 000 tonna szilárd hulladék. Ezt az értéket nagymértékben meghaladja a grafikus rendszerben jelenleg használt 100 meghatározott zónában található hulladék mennyisége. Az erőmű nem tudja ellátni a teljes meghatározott területet, mivel korlátozott kapacitással rendelkezik, amelyet nem lehet túllépni, ezért egy másik algoritmus került kidolgozásra, amelynek célja a legelőnyösebb, ténylegesen ellátható zónák meghatározásának segítése. Ez az algoritmus három egyszerű elven alapult:

- előnyben kell részesítenie a WtE-üzemhez közelebb eső zónákat,
- ha egy zónát „megnyitottak” - a hulladékgyűjtés megkezdődött -, a zónát teljesen ki kell üríteni, nem maradhat benne szilárd hulladék nyoma,
- a WtE-üzem maximális kapacitását nem lehet túllépni.

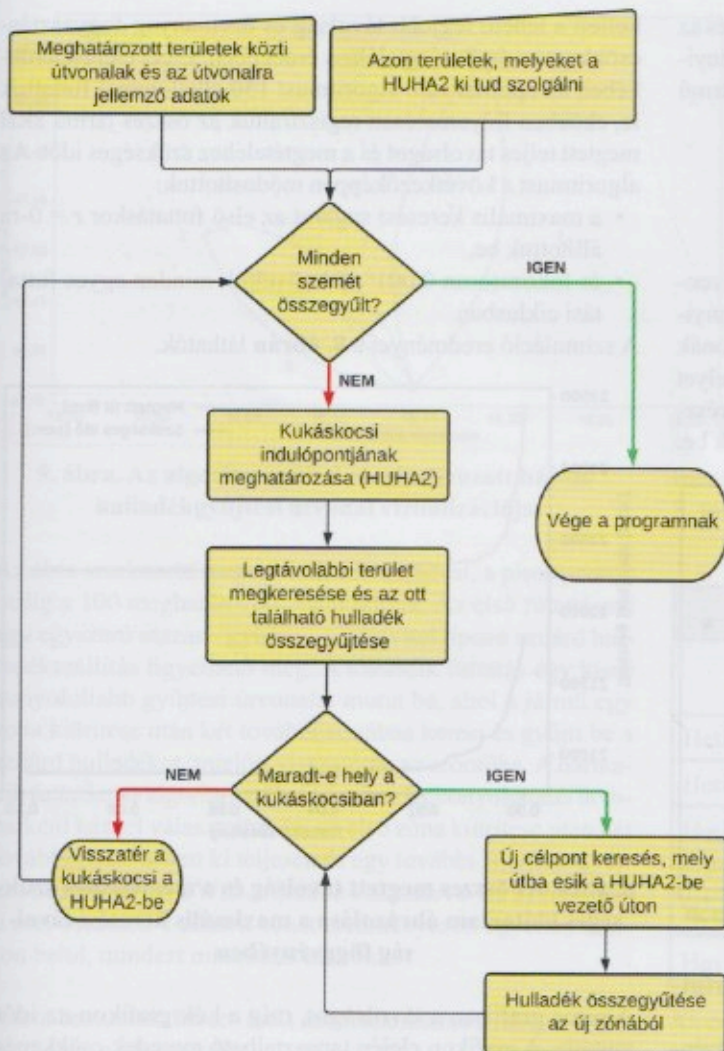
Ezeket a szempontokat szem előtt tartva az algoritmus meghatározta az optimális ellátandó zónákat. Az eredményeket a későbbiekben az **Eredmények** fejezetben mutatjuk be.

3.2. Útvonaloptimalizáló algoritmus

Mivel az összes lehetséges útvonalkombinációt és a lehetséges zónákat a fentiek szerint meghatároztuk, az optimalizáló algoritmus megtervezéséhez szükséges összes információ biztosított volt. Az optimalizálás módszere egy saját fejlesztésű, logikai kapukon alapuló algoritmus volt a számításokhoz használt csúcspontok hatalmas mennyisége miatt. Az e pontok közötti optimális útvonal kiszámítása heurisztikus algoritmusokkal vagy lineáris egészértékű programozással óriási számítási erőforrásokat igényelne, ami az optimalizálást rendkívül hatékonytalanná tenné, ezért a saját fejlesztésű algoritmus e probléma leküzdésére készült. Az algoritmus logikáját a következő oldalon látható **5. ábra** szemlélteti.

Az algoritmus először azt elemzi, hogy az összes hulladékot begyűjtötték-e a megadott zónákban vagy sem. Ha bármelyik zónában maradt hulladék, a települési hulladékszállító jármű elhagyja a hulladékhasznosító üzemet. Az első érdekes pont az erőműtől legtávolabbi csúcspont lesz. Ennek a döntésnek az oka a következő:

1. Az összes korábban azonosított zónát kivétel nélkül ki kell üríteni, így a legtávolabbi csúcsot mindenképpen meglátogatjuk.
2. Ha a jármű első célpontja a legtávolabbi pont, a szállítás legköltséghatékonyabb módja az, ha üresen teszi meg a jármű a zónához vezető utat, így minimalizálva a jármű súlyát és fogyasztását.
3. Ha a legtávolabbi zóna kiürült és a járműnek még van némi tárolóképessége, akkor a zónából a WtE-üzembe való visszaúton más zónákat is elláthat. Mivel a visszaútvonalat



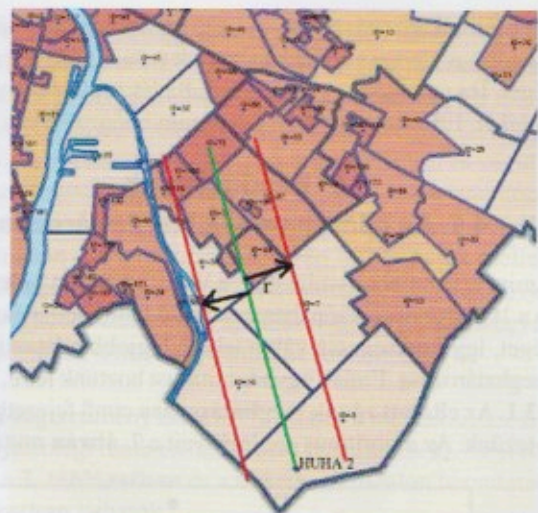
5. ábra. Az útvonaloptimalizáló algoritmus logikai folyamatábrája

mindenképpen meg kell tenni, a második zóna ellátása által okozott kisebb kitérő csak kismértékben növeli a teljes utazási távolságot. Ezt a módszert nevezhetjük „trickle back ellátásnak”.

4. A trickle back ellátás csak akkor életképes, ha a meglátogatott zónák külső peremét őrítik ki először a belső zónákhoz képest, logikai okokból: egyszerűen nem lehet hulladékot gyűjteni a visszaúton, ha az erőműhöz közelebbi zónákat már elsőként ellátták.

Miután a hulladékot a legtávolabbi zónában összegyűjtötték, az algoritmus elemzi, hogy maradt-e még tárolókapacitás a járműben. Ha a válasz igen, akkor a visszautazás során keres egy másik zónát, ahová szállíthat. Az optimális ellátandó zóna megtalálása érdekében másodszor egy keresési sugár módszert dolgoztak ki. A sugármechanika szemléltetése a 6. ábrán látható.

Először is, a jármű jelenlegi helye és a fenti ábrán zöld színrel jelölt erőmű helye között egy légvonalbeli egyenest határozzunk meg. Másodszor, az egyenesre merőleges vektorpár kerül meghatározásra.



6. ábra. A keresési sugár vizualizálása a térképen

A vektorok hossza jelzi a keresési sugár végét egy adott pillanatban. Maga a keresési sugár egy téglalap, amelyet a légvonal és a vektorok hossza határoz meg. A vektorok hossza ciklikusan növekszik $0,01^\circ$ -kal Budapest szélességi fokán mérve, ami $75,124 \text{ km}/1^\circ$ -nak felel meg. Ennek az értéknek a számítási módszere a képlet sorszámával fejezhető ki.

$$F_{\text{fok}} = \frac{2 \cdot r_{\text{Föld, egyenlítő}} \cdot \cos \alpha \cdot \pi}{360^\circ}, \quad (6)$$

$$F_{\text{fok}} = 75,124 \frac{\text{km}}{1^\circ}, \quad (7)$$

ahol

$r_{\text{Föld, egyenlítő}}$ – a Föld egyenlítői sugara [km]

α – szélességi fok Budapest magasságában $[\circ]$

F_{fok} – egy fok kilométerben megadott értéke Budapest ívén $[\text{km}/1^\circ]$

A keresési vektorok maximális hosszát az aktuális jármű és az erőmű közötti távolság harmadára állítottuk be. Ez csak egy kezdeti érték volt, amelyet később optimalizáltunk, az eredmények az **Eredmények** fejezetben láthatók. Minden egyes ciklussal a vektorok hossza a maximális határértékig növekedett, és e ciklusok során az algoritmus a következő szempontok alapján elemezte a csúcspontokat:

- amelyik zóna csúcspontja a sugáron belül helyezkedik el,
- a zóna MSW-tartalma,
- a második zóna ellátása miatt szükséges többlet távolság nagysága.

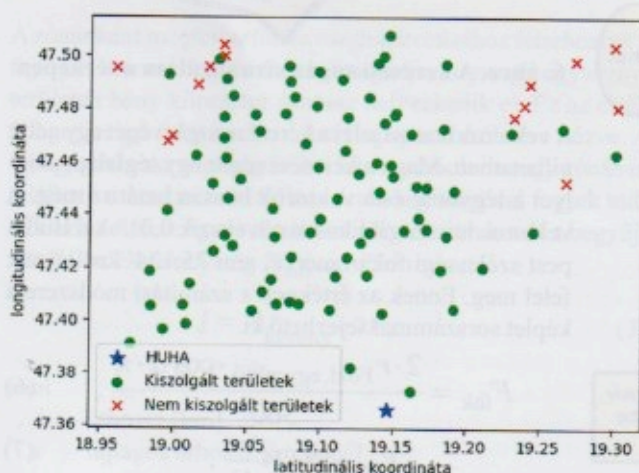
Ezen megfontolások alapján az algoritmus meghatározta az optimális csúcspontot, mint másodlagos célpontot. E lépést követően a hulladékgyűjtés a második célállomáson elkezdődik, és az algoritmus ismét elemzi, hogy a járműnek van-e szabad tárolókapacitása vagy sem. Ha a válasz igen, akkor a fent bemutatott módszerek alapján új keresési sugarat határoz meg és új zónát lát el. Ez mindaddig folytatódik, amíg a jármű tárolókapacitása meg nem telik, vagy amíg a keresési körzeten belüli zónákban az összes szilárd hulladékot be nem gyűjtötték.

Ha a tárolókapacitás betelt, a jármű visszatér az erőműbe és az algoritmus elemzi a zónákban maradt szilárd hulladék mennyiségét. Ha van még begyűjtetlen hulladék, akkor egy új jármű indul el. Ellenkező esetben a program befejeződik.

4. Eredmények

4.1. Az ellátott zónák meghatározása

Amint azt korábban említettük, a WtE-üzem nem tudta felvenni a 100 azonosított zónában keletkező teljes hulladékmennyiséget, így szükségessé vált a lehető legjobb ellátandó zónák meghatározása. Emiatt egy algoritmust hoztunk létre, amelyet a 3.1. Az ellátott zónák meghatározása című fejezetben részletezünk. Az algoritmus eredményeit a 7. ábrán mutatjuk be.



7. ábra. Az ellátott zónák ábrázolása koordináta szerint

A 7. ábra a földrajzi koordináták alapján mutatja a zónaközpontok (csücspontok) helyét. Az algoritmus által megfelelőnek ítélt zónákat zöld pontokkal jelöltük, míg a nem megfelelő zónákat piros kereszttel. A 100 zónából összesen 85 zónát sikerült ellátni. Megfigyelhető, hogy egyes ellátatlanul maradt zónák fizikailag közelebb helyezkednek el a WtE-erőműhöz (kék csillaggal jelölve), mint más ellátandó zónák. Ez a korábban, az Az ellátott zónák meghatározása c. fejezetben részletezett három feltétellel magyarázható, főként a következő kritériumokkal:

- ha egy zónát „megnyitottak” - a hulladékgyűjtés megkezdődött -, a zónát teljesen ki kell üríteni, nem maradhat benne szilárd hulladék nyoma,
- a WtE-üzem maximális kapacitását nem lehet túllépni.

Az érintetlenül hagyott zónákat nem látták el, mivel azok szilárdhulladék-tartalma túl magas volt ahhoz, hogy az erőmű a számítás adott állapotában befogadja. Ennek eredményeképpen az algoritmus a távolabbi, kisebb MSW-tartalmú zónák ellátását választotta.

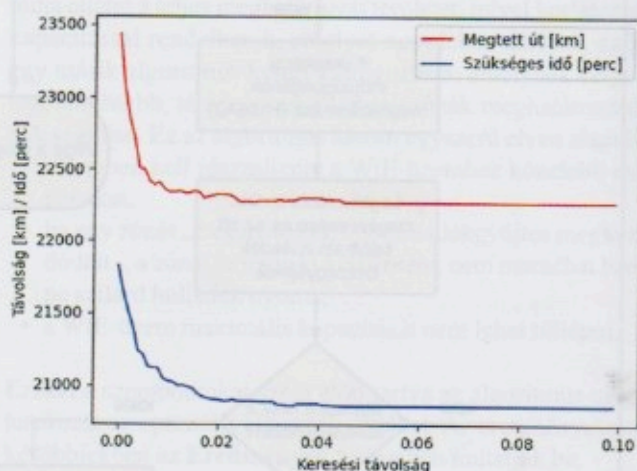
4.2. A keresési sugár optimalizálása

Korábban azt is említettük, hogy a keresési sugarat, amelyet a potenciálisan legközelebbi feltételezhető zóna meghatározásához használtunk, egy kezdeti értékkel, mint maximális keresési sugár szélességgel határoztuk meg. Ezt az értéket optimalizálni

kellett a lehető legjobb távolság és üzemanyag-fogyasztás-csökkentési érték megtalálása érdekében. E cél elérése érdekében az optimalizáló algoritmust 150 alkalommal futtattuk le, közben folyamatosan regisztráltuk az összes jármű által megtett teljes távolságot és a megtételéhez szükséges időt. Az algoritmust a következőképpen módosítottuk:

- a maximális keresési sugarat az első futtatáskor $r = 0$ -ra állítottuk be,
- és fokozatosan $0,001^\circ$ -kal növeltük minden egyes futtatási ciklusban.

A szimuláció eredményei a 8. ábrán láthatók.



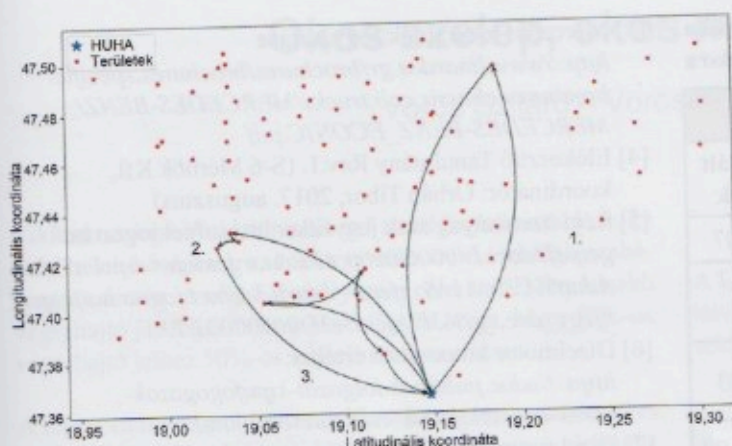
8. ábra. Az összes megtett távolság és a megtételhez szükséges időtartam ábrázolása a maximális keresési távolság függvényében

A piros grafikon a távolságot, míg a kék grafikon az időt mutatja. A grafikon elején tapasztalható meredek csökkenés a keresési sugár jelentőségével magyarázható, mivel a nulla szélesség azt jelenti, hogy nincs mód egy esetleges második ellátható zóna keresésére, tehát nincs keresési sugár. Megállapítható, hogy még egy kis keresési sugár szélessége is hatalmas hatással lehet a megtett távolság hosszára és a hozzá kapcsolódó időre. Nagyobb keresési sugárértékeknél fordított arányosság tapasztalható a sugár szélessége és a mért értékek között. A nagyobb szélesség az eredeti útvonaltól távolabb eső zónaválasztási lehetőségeket biztosít, így csökken annak valószínűsége, hogy az algoritmus ezeket másodlagos gyűjtőpontként választja. Bizonyos számú lefutás után mindkét mért értékben stagnálás figyelhető meg.

Az optimális maximális keresési sugár értéke $r = 0,045^\circ$ lett. Elméletileg lehetne nagyobb is, de ez nem változtatná meg a megtakarítások eredményét a stagnáló értékek miatt. Másrészt a számítási erőforrásigény gyorsan növekedne, ami az optimalizálási módszert rendkívül hatékonytalaná tenné.

4.3. Az útvonaloptimalizálás hatásai

Az optimális maximális keresési sugár szélességének ismeretében a programot úgy módosítottuk, hogy ezeknek az értékeknek megfelelően. Ebben a fejezetben az eredményeket a 9. ábrával kezdve részletezzük, amely egy kis betekintést nyújt az algoritmus belső működésébe.



9. ábra. Az algoritmus által meghatározott három hulladékgyűjtési útvonal vizualizációja

Az ábra szerkezete megegyezik a 8. ábráéval, a piros pontok pedig a 100 meghatározott zónát jelölik. Az első futtatásnál egy egyszerű utazás - gyűjtés - visszavitel típusú szilárd hulladékszállítás figyelhető meg. A második futtatás egy kissé bonyolultabb gyűjtési útvonalat mutat be, ahol a jármű egy zóna kiürítése után két további zónában keresi és gyűjti be a szilárd hulladékot, mielőtt visszatérne az erőműbe. A harmadik futtatást az algoritmus által készített legbonyolultabb útvonalként kézzel választottuk ki: az első zóna kiürítése után két további zónát ürített ki teljesen és egy további zónában gyűjtött be hulladékot a WtE-erőműbe visszatérve. Ez a rendkívüli eset összesen 4 ellátott zónát eredményezett egyetlen futáson belül, mindezt minimális kitérőkkel.

A fent bemutatott három futás azonosítószámai a látogatás sorrendjében a következők:

1. futás: 1.0 → 16.0 → 1.0
2. futás: 1.0 → 159.0 → 181.0 → 14.0 → 1.0
3. futás: 1.0 → 34.0 → 56.0 → 145.0 → 10.0 → 1.0

Mivel az egy hét alatt keletkező szilárd hulladék elszállításához szükséges összes futtatás száma 553, a teljes adatkészlet bemutatása meghaladná a cikk kereteit. Ezért csak egy részét mutatjuk be, hogy szemléltessük az 1. táblázatban látható szerkezetét.

Látogatott IDk [-]	Távolság [km]	Idő [min]	Visszatérő kapacitás [kg]	Zónán belüli távolság [km]	Teljes távolság [km]	Üzemanyag fogyasztás [l]	Költség [Ft]
1.0 → 16.0 → 1.0	83,30	66,00	0,00	25,27	108,57	79,26	64 992
1.0 → 16.0 → 1.0	83,30	66,00	0,00	25,27	95,67	69,84	64 992
1.0 → 16.0 → 112.0 → 1.0	70,40	71,00	0,00	25,27	66,71	48,70	51 013
1.0 → 64.0 → 1.0	62,00	48,00	0,00	4,71	66,71	48,70	39 932
1.0 → 64.0 → 1.0	62,00	48,00	0,00	4,71	66,71	48,70	39 932
1.0 → 64.0 → 1.0	62,00	48,00	0,00	4,71	66,71	48,70	39 932
1.0 → 64.0 → 1.0	62,00	48,00	0,00	4,71	66,71	48,70	39 932
1.0 → 64.0 → 1.0	62,00	48,00	0,00	4,71	66,71	48,70	39 932
1.0 → 64.0 → 1.0	62,00	48,00	0,00	4,71	67,41	49,21	39 932
1.0 → 64.0 → 54.0 → 1.0	62,70	51,00	0,00	4,71	60,31	44,03	40 351
1.0 → 54.0 → 1.0	55,60	39,00	0,00	4,71	60,31	44,03	36 101

Az algoritmust úgy építettük fel, hogy adatokat gyűjtson a zónák közötti utazás során megtett távolságról, az ehhez szükséges időről, a zónákon belül megtett távolságról, az ezen adatok alapján megtett teljes távolságról, az utazások üzemanyag-fogyasztásáról és végül a szállítással kapcsolatos költségekről. Ez az adatkészlet a szimulációban jelen lévő minden járműre vonatkozóan rendelkezésre áll. Ezen információk mellett az algoritmus a programkód minőségének ellenőrzése érdekében adatokat gyűjtött az útról visszatérő egyes járművek szabad kapacitására vonatkozóan is.

A végeredmény elérése érdekében a megtakarításokat a jelenlegi rendszerhez képest számítottuk ki. Az eredmények a 2. táblázatban és a következő oldalon bemutatott 3. táblázatban láthatók.

2. táblázat. Az optimalizáció gazdasági hatásai

Az optimalizálás hatása a költségekre		
	Direkt útvonalak	Optimalizált útvonalak
Heti távolság [km]	23 434,87	22 226,07
Heti időtartam [min]	21 824,00	20 857,00
Heti üzemanyag fogyasztás [liter]	17 107,46	16 225,03
Heti költségek [Ft]	14 028 115,72	13 304 528,04
Havi megtakarítás [Ft]	351	2 894

Az eredmények szerint havi 2 894 351 Ft potenciális megtakarítás érhető el, ami éves szinten 34 732 209 Ft megtakarításnak felel meg. Az optimalizálás a szilárd hulladékszállítás környezeti oldalára is hatással van, mivel a járművek széndioxid- és nitrogén-oxid-kibocsátása havonta 9459,6 kg-mal, illetve 45,3 kg-mal csökken. Ez az érték éves időtávlatban is megjeleníthető, ami 113,5 tonna CO₂ és 543,6 kg NO_x kibocsátás elkerülését jelenti évente a fővárosban.

1. táblázat. Az optimalizáló algoritmus által előállított adatstruktúra ábrázolása

3. táblázat. Az optimalizálás hatása a károsanyag-kibocsátásra

Az optimalizálás hatása a kibocsátásokra		
	Direkt útvonalak	Optimalizált útvonalak
Heti távolság [km]	23 434,87	22 226,07
Fajlagos CO ₂ kibocsátás [kg CO ₂ /l]	2,68	
Heti üzemanyag fogyasztás [liter]	17 107,46	16 225,03
Heti CO ₂ kibocsátás [kg]	45 847,99	43 483,09
Heti NOx kibocsátás [kg]	219,55	208,23
Havi CO ₂ megtakarítás [kg]	9 459,59	
Havi NOx megtakarítás [kg]	45,30	

Források

- [1] Magyarország Nemzeti Atlasza
https://emna.hu/map/Bp_nepsur_2011/@47.4663625,19.1274506,12.00z
- [2] Egy főre jutó éves hulladéktermelés Magyarországon
<https://www.europarl.europa.eu/news/hu/headlines/society/20180328STO00751/hulladekkezeles-az-eu-ban-trendek-es-statisztikak-infografika>

- [3] Mercedes teherautók fogyasztása
http://www.linardos.gr/brochures/brochures.specific/brochures.chassis.cab.trucks/MERCEDES-BENZ/MERCEDES-BENZ_ECONIC.pdf
- [4] Előkészítő Tanulmány Rev.1. (S-6 Mémők Kft, koordinátor: Orbán Tibor, 2017. augusztus)
- [5] Régi üzemanyag árak jegyzéke: https://net.jogtar.hu/getpdf?docid=000G0000.APE&targetdate=&printTitle=Adatt%C3%A1r&referer=http%3A//net.jogtar.hu/jr/gen/hjegy_doc.cgi%3Fdocid%3D000000003.TXT
- [6] Dízelmotor kibocsátási értékei:
<https://adoc.pub/szakdolgozat-kipufogogazok-karos-emissziojanak-cskkentese-re-.html>
- [7] Dízel üzemanyag kibocsátása:
https://en.wikipedia.org/wiki/Diesel_exhaust

Köszönetnyilvánítás

A Kulturális és Innovációs Minisztérium ÚNKP-22-2-I-BME-66 kódszámú Új Nemzeti Kiválóság Programjának a Nemzeti Kutatási, Fejlesztési és Innovációs Alapból finanszírozott szakmai támogatásával készült.



KULTURÁLIS ÉS INNOVÁCIÓS NEMZETI KUTATÁSI, FEJLESZTÉSI
ÉS INNOVÁCIÓS HIVATAL



Új Nemzeti
Kiválóság Program