



M Ű E G Y E T E M 1 7 8 2

BUDAPESTI MŰSZAKI ÉS GAZDASÁGTUDOMÁNYI EGYETEM
TERMÉSZETTUDOMÁNYI KAR
DIFFERENCIÁLEGYENLETEK TANSZÉK

DIPLOMAMUNKA

Célforgalmi mátrix becslése valós idejű közúti forgalmi paraméterek alapján

Ludvig Ádám

Témavezetők :

Dr. Hujter Mihály	Tettamanti Tamás
egyetemi docens	egyetemi tanársegéd
Differenciálegyenletek Tanszék	Közlekedés- és Járműirányítási Tanszék

Budapest, 2013. május 31.

Tartalomjegyzék

1. Bevezetés	6
2. Matematikai eszközök	8
2.1. Lineáris algebra	8
2.2. Lagrange-függvény	8
2.3. Konvex optimalizálás	9
2.4. Maximális entrópia elve	10
2.5. Gráfelmélet	11
2.5.1. k -legrövidebb út probléma	11
3. Közlekedésmérnöki eszközök	13
3.1. Fogalmak	13
3.1.1. Úthálózat	13
3.1.2. Forgalm nagyság	13
3.1.3. Célforgalmi mátrix	14
3.1.4. Ráterhelés	15
3.1.5. Használói egyensúly	15
3.1.6. Rendszeroptimum	15
3.1.7. Braess paradoxon	15
3.2. Forgalm szimuláció	16
3.2.1. Mikroszkopikus megközelítés	16
3.2.2. Makroszkopikus megközelítés	17
3.3. Forgalomtechnikai paraméterek mérése	17
3.3.1. Keresztmetszeti mérés	18
3.3.2. Vonali mérés	19
3.3.3. Hálózati mérés	19
4. Problémafelvetés	22
4.1. A célforgalmi mátrix meghatározásának nehézsége	22
4.2. Mobiltelefon alapú közlekedési igény meghatározás	23
4.3. Adatfúzió	23

5. Modell	24
5.1. Közút hálózat reprezentációja gráffal	24
5.2. Változók	25
5.3. Ellenállás függvény	27
5.4. Mérések	29
5.4.1. Keresztmetszeti mérés	29
5.4.2. Hálózati mérés	29
5.5. A célforgalmi mátrix kiszámolhatósága	30
6. Implementáció	31
6.1. Eszközök	31
6.1.1. Matlab	31
6.1.2. Mathematica	31
6.2. A használói egyensúly átfogalmazása optimalizálási feladattá	32
6.3. Utak meghatározása	35
6.3.1. Yen-féle k -legrövidebb út algoritmus	35
6.3.2. K paraméter megválasztása	38
6.4. A kifeszített alterek meghatározása	38
7. Validáció	41
7.1. Eszközök	41
7.1.1. VISSIM	41
7.1.2. VISUM	42
7.2. Teszthálózat	42
7.3. Gráfalgoritmusok	45
7.4. Forgalomszimuláció	46
8. Összegzés és továbbfejlesztési lehetőségek	49
A. Yen féle k-legrövidebb út algoritmus forráskódja	50
Ábrák jegyzéke	51
Táblázatok jegyzéke	52
Irodalomjegyzék	53

Kivonat

Napjaink népszerű és aktívan kutatott területe az intelligens közlekedési rendszerek (ITS) témaköre, melynek jelentős ága a növekvő közlekedési igények kielégítése a már létező közlekedési infrastruktúrán a hatékonyság javításával. A hatékony közlekedésirányítás elengedhetetlen eleme a közlekedési igények pontos és aktuális ismerete, amit a célforgalmi mátrix fejez ki.

Adott mérési lehetőségek esetén a célforgalmi mátrix számolhatóságáról és becslésének módszeréről, valamint annak számítógépes implementációjáról szól a dolgozat.

A közlekedési igények és a realizálódó járműáramlatok közötti összefüggést a ráterhelési feladat jelenti, aminek a megoldásában a konvex optimalizálás eszközeire is hagyatkozhatunk.

Az implementáció a matematikusok között népszerű Mathematica és az iparban széleskörűen használt Matlab szofvercsomagok segítségével történt. Az elkészült rutinok tesztelésére VISSIM és VISUM forgalomszimulációs eszközök lettek alkalmazva.

Abstract

Nowadays Intelligent Transportation Systems (ITS) is a popular research topic. One of its goals is to make the transportation more efficient on existing infrastructure by active flow control. The prompt and current estimation of the traffic demand is intrinsic for the efficient traffic control. The demand is formulated by the origin–destination matrix of the traffic system.

The determinability and estimation possibilities of the origin–destination matrix according to the existing measurement technologies and the corresponding computer implementations are the main achievements of the thesis.

The traffic assignment procedure gives the relation between the traffic demand and the evolving traffic flows on the links of a traffic network. Convex optimisation yields one way to solve effectively the traffic assignment problem.

The implementation of the used subroutines was achieved by the popular software Mathematica and Matlab which is extensively utilised in the engineering field. For testing the subroutines two software were used, VISSIM and VISUM, to simulate realistic traffic flows.

Köszönetnyilvánítás

Köszönöm a segítséget és a kitartást konzulenseimnek Tettamanti Tamásnak és Hujter Mihály tanár úrnak. Tamás, egykori tanárom, mérhetetlen mennyiségű és felbecsülhetetlen értékű tanácsot és segítséget nyújtott az együtt töltött évek alatt a Közlekedésautomatikai Tanszék Közúti Közlekedésirányítási laboratóriumában, melynek én is aktív tagja voltam. Hujter tanár úr szakmai tapasztalatával és praktikus tanácsaival segített a dolgozat megfelelő kialakításában, hogy az, mint matematikus dolgozat is megállja helyét.

Köszönöm barátomnak, Klicsu Zsoltnak, hogy mellettem állt és segítségemre volt a nehéz időszakokban.

1. fejezet

Bevezetés

Korunkban a korszerű, gyors és hatékony közlekedés eszközeivel a világ összezugorodott, a személyi és az áru fuvarozás egyre bővülő kínálata nyújt megoldást a világ inhomogenitásának haszonra fordításában. A növekvő igények és a fejlődő technika napról napra kihívás elé állítja a korokkal ezelőtt kialakult és sok esetben, mint például történelmi városok, változtathatatlan közlekedési infrastruktúrát.

Az intelligens közlekedési rendszerek (ITS), amik az EIT kutatási hivatal témáinak is szerves része¹, nyújtanak megoldást a növekvő igények kielégítésére és a közlekedés hatékonyabbá tételére.

A történelem során ritkák az infrastruktúra korlátossága miatt jelentkező kapacitásproblémák, mivel a gyors, olcsó és nagy kapacitású szállítóeszközök csak az utóbbi évszázadban terjedtek el. A korábbi emberi vagy állati szállítás sebessége és infrastruktúra igénye jelentősen eltér a korunkra jellemző, gépesített szállítási módoktól.

Az útépités első komoly képviselője a Római Birodalom volt, ahol a hadsereg gyors eljuttatása legalább annyira fontos feladat volt, mint a már elfoglalt területek kereskedelmének felvirágoztatása az utak által.

A hadvezérek repertoárjának is része volt a csaták környezetének kiaknázása, hogy a saját csapatukat segítsék, míg az ellenséges csapat közlekedése akadályoztatva legyen, reakálási lehetőségei szűküljenek.

Ennek ékes példája a Thermopülai csata, melyben a görög és a perzsa seregek ütköztek meg. A szoros feletti ellenőrzés stratégiai jelentősége hatalmas, mivel ezen keresztül lehetett Athénba jutni. A szoros kiváló lehetőséget teremtett a kis számú görög hadseregnek a perzsa sereg feltartóztatására, mivel az a szorosban nem tudta túlerejét érvényesíteni. Kapacitásproblémák léptek fel és szoros jellegéből adódóan a perzsa sereg cselekvőképessége jelentősen csökkent. A csatát a görög sereg végül

¹<http://www.eitictlabs.eu/innovation-areas/intelligent-mobility-and-transportation-systems/>

elvesztette egy árulás következtében, de rávilágított mekkora akadály lehet a szoros egy hatalmas hadseregnek.

Későbbre ugorva a Zentai csata nyújt szemléletes példát az infrastruktúra korlátozó hatásáról jelentős forgalom esetén. A tizenhetedik századi török hódoltság alatt álló Magyarország felszabadítására szervezett hadsereget Savoyai Jenő vezette, aki rájött, hogy a Tiszán átkelni igyekvő hatalmas török sereget épp átkelés közben a leghatékonyabb megtámadni. A folyón kialakított hajóhíd kettéosztotta a török sereget, amit a keresztény sereg a hídfőnél támadott meg. A sarokba szorított törökök védekezése a lassú menekülési út miatt összeomlott, sokan a folyóba vetették magukat, hogy úszva meneküljenek meg Savoyai seregétől.

A vázolt történetek sejtetik, hogy a meglévő infrastruktúra hatékonyabb használatával nem csak a gazdaság hatékonyságát, de akár életet is lehet menteni.



1.1. ábra. Jacques Ignace Parrocel festménye a zentai ütközetről

2. fejezet

Matematikai eszközök

Az alábbiak az alkalmazott matematikai eszköztár fontosabb elemei kerülnek megemlítésre, melyeket alkalmazásra, vagy megemlítésre kerülnek a dolgozat folyamán.

2.1. Lineáris algebra

A dolgozat erősen építkezik a lineáris algebrára, mivel a tárgyalt módszerek főleg lineáris egyenletekre és azok megoldására vezethetőek vissza. Ebben Wettl Ferenc munkájára hagyatkozik, a fogalmakat és jelöléseket, ha másként nem jelölt akkor onnan kölcsönzi[5].

A lineáris függetlenség, az egyenletrendszerek megoldhatósága és a megoldások terének fogalmai, a mátrixok sor- és oszlopterei valamint rangja a célfogalmi mátrix kiszámolhatósága szempontjából lényeges, amit az 5.5 szakasz tekint át.

2.2. Lagrange-függvény

A dolgozat egy kis részében hangsúlyos szerepet kap a Lagrange-függvény. A ráterhelési feladat felhasználói egyensúlyának megfogalmazását egyszerűsíti és alakítja konvex optimalizálási kérdéssé.

A mechanika újraformalizálására bevezetett Lagrange-függvény egy rendszer dinamikájának összességét írja le. A klasszikus mechanikának a Lagrange-függvénye a mozgási és a potenciális energia különbsége. Hatása túlmutat a mechanikán, a modern fizika legkisebb hatás elvét, a variációs számítást alkalmazó vizsgálatára ad lehetőséget[16].

Fontos alkalmazása a Lagrange-féle multiplikátor módszer, melyet a következő tétel részletez[23].

2.2.1. Tétel. Legyenek $f, g_1, g_2, \dots, g_k : \mathbb{R}^n \rightarrow \mathbb{R}$ folytonosan differenciálható függvények és tegyük fel, hogy az f függvénynek a $g_i = 0$ feltétel mellett feltételes szélsőértéke van a $p \in D(f)$ pontban. Valamint tegyük fel, hogy a g_i skalárfüggvényekből képzett \underline{g} vektorfüggvény Jacobi mátrixának rangja k , azaz teljes rangú.

$$\underline{g} = \begin{pmatrix} g_1 \\ g_2 \\ \vdots \\ g_k \end{pmatrix} \quad (2.1)$$

$$\text{rang}(J_{\underline{g}}) = k \quad (2.2)$$

Ekkor léteznek $\lambda_1, \lambda_2, \dots, \lambda_k \in \mathbb{R}$ számok, hogy az $F = f + \sum_{i=1}^k \lambda_i \cdot g_i$ függvénynek lokális szélsőértéke van a p helyen. Tehát

$$\forall j \leq n : \frac{\partial F}{\partial x_j}(p) = \frac{\partial f}{\partial x_j}(p) + \sum_{i=1}^k \lambda_i \cdot \frac{\partial g_i}{\partial x_j}(p) = 0. \quad (2.3)$$

2.3. Konvex optimalizálás

2.3.1. Definíció. Az $f : \mathbb{R}^n \rightarrow \mathbb{R}$ függvény konvex, ha minden $x, y \in \mathbb{R}^n$ és minden $\alpha, \beta \in [0, 1] \subset \mathbb{R}$ értékre

$$f(\alpha \cdot x + \beta \cdot y) \leq \alpha \cdot f(x) + \beta \cdot f(y), \quad (2.4)$$

ahol $\alpha + \beta = 1$.

2.3.2. Definíció. A konvex optimalizálási feladat az $f_0(x)$ függvény minimalizálása $f_i(x) \leq b_i, i \in \{1, \dots, m\}$ feltételekkel, ahol minden $i \in \{0, \dots, m\}$ értékre $f_i : \mathbb{R}^n \rightarrow \mathbb{R}$ függvény konvex.

A legkisebb négyzetek módszere és a lineáris programozás speciális esetei a konvex optimalizálásnak.

Nincs általános analitikus megoldóformula a konvex optimalizálási feladat megoldására, ugyanakkor léteznek nagyon hatékony megoldási módszerek. A belső pontos módszerek a gyakorlatban nagyon gyorsan és pontosan alkalmazhatóak, a célfüggvény szélsőértékének adott pontosságú eléréséhez nem haladják meg a polinomiális probléma osztályt a dimenziószám és a feltételek függvényében[3].

A ráterhelési feladat megoldásában alkalmazott Frank-Wolfe algoritmus, vagy más néven a feltételes gradiens módszer az optimalizált függvény lineáris közelítésével halad a gradiens mentén pontonként a globális szélsőérték felé. Az alábbi pszeudokódban f konvex függvény értékét kell minimalizálni a \mathcal{D} konvex tartományon[12].

Inicializálás Legyen $x_0 \in \mathcal{D}$ egy tetszőleges pont és $k = 0$.

Iránykeresés Határozzuk meg $s \in \mathcal{D}$ értékét úgy, hogy $s^T \cdot \nabla f(x_k)$ minimális legyen. Ez f függvény lineáris közelítésével a \mathcal{D} tartomány határán jelöli ki az s pontot.

Lépéshossz Határozzuk meg $\gamma \in [0, 1]$ értékét úgy, hogy $f(x_k + \gamma \cdot (s - x_k))$ értéke minimális legyen.

Frissítés Legyen $x_{k+1} = x_k + \gamma \cdot (s - x_k)$ és $k = k + 1$, majd ismét az Iránykeresés lépése következik.

Listing 2.1. Frank-Wolfe algoritmus

2.4. Maximális entrópia elve

2.4.1. Definíció. *Egy zárt termodinamikai rendszer állapotait különböző valószínűséggel veszi fel. Az $S = \{x_1, x_2, x_3, \dots\}$ állapothalmaz mutatja a lehetséges állapotokat. Az x_i rendszerállapot valószínűsége ω_i . A rendszer entrópiája az x_i állapotban*

$$S_i = -k \cdot \ln(\omega_i), \quad (2.5)$$

ahol k a Boltzmann-állandó.

Az entrópia fogalma kiterjeszhető általános dinamikai rendszerekre is, mint például a közlekedés. A rendszerben lejátszódó folyamatok során a rendszer entrópiája mindaddig nő, amíg az egyensúlyi állapot be nem áll, ezáltal annak az entrópiája maximális lesz.

Modellvizsgálatánál a maximális entrópia elve a rendszer méréseknek megfelelő modelljei közül a legnagyobb információtartalommal rendelkező, az egyensúlyi állapotot legjobban megközelítő állapotot jelöli ki[14].

A célforgalmi mátrix pontos meghatározása gyakran nem lehetséges az elvégzett mérések alapján, ezért annak megbecslésére a szabad paraméterek mentén a maximum entrópia módszert lehet alkalmazni[1].

2.5. Gráfelmélet

A közút hálózat reprezentációjára a gráfok természetes lehetőséget kínálnak. Többféleképpen lehet az úthálózatot gráfokkal leírni, jelen dolgozatban az általános, hurokél mentes, irányított gráf fogalmát értem gráf alatt.

A csúcsok a közúti csomópontoknak felelnek meg és gyakran csak csomópontként vagy kereszteződésnéven hivatkozom rájuk. Az éleket útszakasznak, a gráfelméleti utat, az él és csúcsismétlés mentes élsorozatot pedig gyakran útvonalnak nevezem.

Az élek súlya nem állandó nagyságú, hanem a rajtuk átfolyó folyam mértékétől függ, részletesen az 5.3 szakasz foglalkozik az élek ellenállásával.

2.5.1. Definíció. Egy $f_{o,d} : \mathbb{E}(\mathfrak{G}) \rightarrow \mathbb{R}$ függvényt folyamnak nevezünk a \mathfrak{G} gráfon, ha az o forrás és d nyelő csúcsok kivételével a \mathfrak{G} valamennyi csúcsára fennáll a folytonosság tétele:

$$\forall x \in \mathbb{V}(\mathfrak{G}) \setminus \{o, d\} : \sum_{e \in \mathbb{E}_x^+} f(e) = \sum_{e \in \mathbb{E}_x^-} f(e) \quad (2.6)$$

ahol \mathbb{E}_x^+ és \mathbb{E}_x^- az x csúcsba befutó és az onnan kiinduló irányított élek halmazai[9].

A közlekedési alkalmazásokban fontos szereppel bír a legrövidebb út meghatározása gráfon, ami két csúcspont között a legkisebb összhosszúságú útvonalat adja meg. Erre a klasszikus Dijkstra féle algoritmus kiválóan használható nem negatív élsúlyok esetén, ami a közlekedésben feltételezhető $\mathcal{O}(|\mathbb{E}(\mathfrak{G})| + |\mathbb{V}(\mathfrak{G})| \cdot \log |\mathbb{V}(\mathfrak{G})|)$ futásidőben, vagy konzisztens heurisztika alkalmazása esetén az A* algoritmus nyújt gyorsabb eredményt[11].

2.5.1. k -legrövidebb út probléma

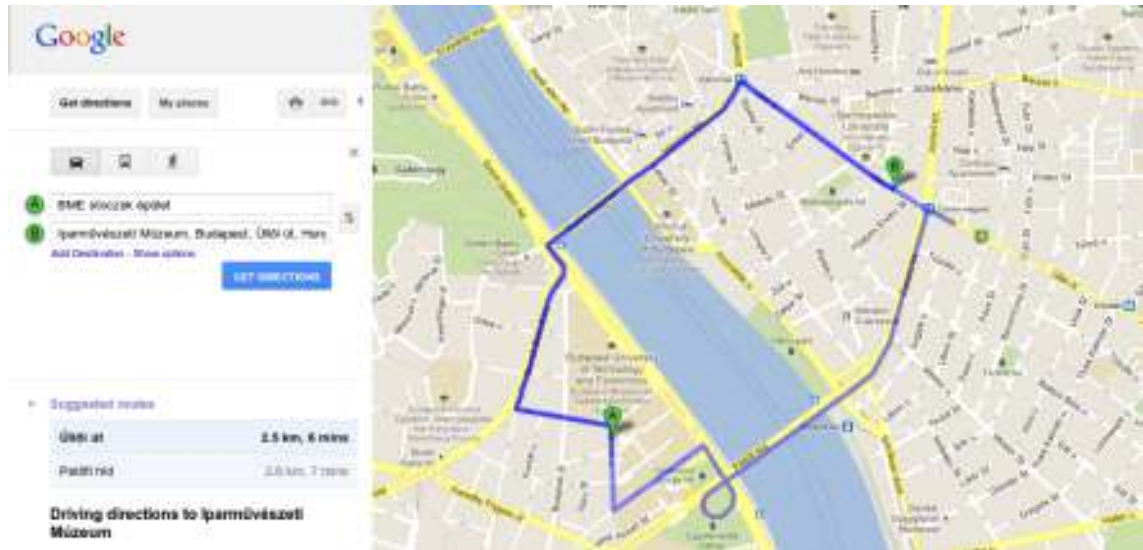
A hatékonyan számolható legrövidebb út mellett a közlekedésben gyakori igény több, vagy az összes útvonal figyelembe vétele két csúcs között, mivel az eljutás költségét, az élsúlyokat befolyásolja az aktuális, alkalmazott folyam. Az előre meghatározott útvonalak listájának ismeretében lehet az élek változó súlyával a különböző folyamatokat számolni.

A gráfban leghosszabb utat előállító algoritmus NP nehéz, mivel visszavezethető a Hamilton út keresésre. Egy n csúcsú gráfban egységnyi élsúlyok esetén egy $n - 1$ csúcsú leghosszabb út találása egyenértékű egy Hamilton út megtalálásával[22].

Ez a komplexitás a nagy és összetett gyakorlati alkalmazásokkal bíró közlekedési szakmában megengedhetetlen, így a feladat egy közelítő egyszerűsítése lehet csak a k -legrövidebb út figyelembe vétele, és k paramétert úgy megválasztva, hogy az ennél hosszabb útvonalak már ne jelentsenek reális alternatívát a valós közlekedésben.

A legrövidebb út problémáját többféleképpen is ki lehet terjeszteni, a dolgozat szempontjából leglényegesebb az összes hurokmentes útvonal közül a hossz szerinti rendezésre első k darab utat adó algoritmus, amivel Jim Y. Yen is foglalkozott[31]. Az algoritmus részletes kidolgozását a 6.3.1 szakasz mutatja be.

Ahogy a legrövidebb útkereső algoritmus is gyorsítható heurisztikával, a k -legrövidebb utat kereső algoritmus javítása is lehetséges heurisztika felhasználásával[2].



2.1. ábra. A Google útvonaltervezője által javasolt alternatív útvonalak

A napjainkban is elérhető útvonaltervezők is többféle útvonalat vesznek figyelembe és javasolnak, mivel a hasonló hosszúságú útvonalakon mérhető eljutási idő a kialakuló forgalomtól jelentősen függ. A 2.1 kép a Google útvonaltervezőjével készült kétféle útvonal ajánlást mutatja be a Műegyetem és az Iparművészeti múzeum épülete között két Duna hídon.

3. fejezet

Közlekedésmérnöki eszközök

3.1. Fogalmak

A fejezet a fontosabb közlekedésmérnöki fogalmakat tisztázza, melyek lényeges szerepet töltenek be a feladat meghatározásában.

3.1.1. Úthálózat

A közút hálózat, vagy röviden úthálózat, a lakosság számára közlekedésre, helyváltoztatásra szolgáló infrastruktúra, melyen többféle járművel – gyalog, kerékpár, személygépjármű, tehergépjármű – lehet közlekedni.

Az úthálózat útszakaszokból és csomópontokból áll. A csomópontokat az útszakaszok kötik össze, melyek lehetnek egy és kétirányúak. Jelen modellben csak egyirányú útszakaszokkal foglalkozom. A csomópontokban nem feltétlenül lehet minden kapcsolódó útszakaszra fordulni, ezt az 5.1 szakaszban bemutatott konstrukcióval lehet kezelni a matematikai modellben.

A valóságban jelentős lehet az útszakaszon bekapcsolódó vagy kiváló járműforgalom, ahogy a járművek leparkolnak az út mentén kialakított parkolóhelyekre vagy parkolóházakba, a modell ettől eltekint, csupán a csúcsokban történhet forgalomváltozás a folyamokban.

3.1.2. Forgalomnagyság

A forgalom vizsgálatának alapvető mennyisége a forgalomnagyság. Ez az egy keresztmetszeten áthaladó járművek számát jelenti az idő függvényében. Mértékegysége a $\frac{\text{jármű}}{\text{óra}}$.

Egy egyenes útszakasznak tipikusan $1800 \frac{\text{jármű}}{\text{óra}}$ a maximális forgalomnagysága forgalmi sávonként, ami két másodperces követési időt jelent járművenként. Ez a

követési idő és forgalomnagyság a mai kor járműveinek technikai jellegzetességeiből és a járművezetők pszichológiai jellemzőiből kialakított átlagos értéknek felel meg, ami $50\text{--}60\frac{\text{km}}{\text{óra}}$ körüli optimális sebességnél adódik. Nagyobb sebesség esetén a követési távolság négyzetes növekedésével az elérhető forgalomnagyság értéke csökken[19].

Mivel az útszakaszon homogén forgalmat feltételezünk, a keresztmetszetre értelmezett forgalomnagyság természetesen kiterjeszthető az útszakaszokra is.

3.1.3. Célforgalmi mátrix

A közlekedésmérnöki szakmában az egyik fontos paraméter a forgalom leírására a célforgalmi, vagy honnan-hová mátrix[10]. Ennek segítségével lehet leírni a közlekedési igényeket az úthálózaton.

A közlekedési igények kifejezéséhez a vizsgált területet diszjunkt, összefüggő Z_i régiókra bontjuk.

A célforgalmi mátrix $D_{i,j}$ eleme a Z_i -ből induló és Z_j -be érkező forgalomnagyságot jelenti, az egységnyi idő alatt áthaladó járműmennyiséget. Ez a folyamat fog áthaladni az úthálózat gráfján és fogja meghatározni az egyes útszakaszokon kialakuló forgalom minőségét.



3.1. ábra. Budapest és környéke a 2008-ban a KTI által végzett országos célforgalmi adatfelvétel térképén

A 3.1 ábra a Közlekedés Tudományi Intézet által 2008-ban végzett, Magyarországot lefedő, kikérdezéses célforgalmi adatfelvételének Budapest környéki régióit ábrázolja. A felbontás a belső kerületek összevonásától eltekintve követi a közigazgatási kerületek határait.

A különböző színű régiókon azok ötjegyű azonosító száma van feltüntetve, ami a célforgalmi mátrixban a régióhoz tartozó oszlopot és sort is jelöli.

3.1.4. Ráterhelés

A ráterhelés azt a közlekedésmérnöki számítási módszert takarja, ami a közlekedési igények és az úthálózat terhelésfüggő kapacitásának ismeretében meghatározza az egyes útszakaszokon folyó forgalom nagyságát[10].

Kétféle általánosan elfogadott elv létezik a ráterhelés kiszámolására, ezekről a következő két szakasz ad áttekintést.

3.1.5. Használói egyensúly

Wardrop első elve mondja ki: Minden utazó a számára kedvezőbb eljutási idejű útvonalat választja[29]. Ennek következménye, hogy az eljutási idő a kezdő és a végpont között minden igénybe vett útvonalon egyenlő és kevesebb, mint a kihasználatlan utakon. Minden utazó a saját költségét, utazási idejét akarja csökkenteni. Ez az állapot a Nash-egyensúly, vagy használói egyensúly a közlekedésben. Ekkor semelyik utazó nem tudja egyedi cselekedettel csökkenteni a költségét.

3.1.6. Rendszeroptimum

Ugyancsak Wardrop fogalmazta meg az optimális állapotot a közúti közlekedésben[29]. Ekkor a közlekedésben résztvevő valamennyi jármű átlagos utazási ideje minimális. Ezen állapotot a közlekedési rendszer optimumának nevezzük. Az állapot eléréséhez a járművek kooperatív viselkedése, gazdasági eszközök, modern forgalomirányítási eszközök vagy ezek kombinációja segítheti a rendszert.

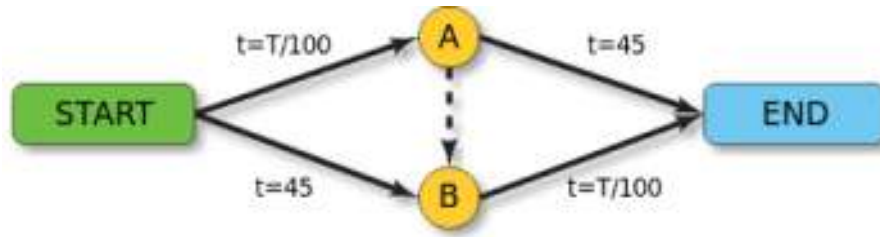
A mai modern forgalomirányítási stratégiák ennek az optimumnak az elérésével próbálják meg a közúti forgalmat hatékonytá tenni a járművek közvetlen vagy közvetett befolyásolásával.

3.1.7. Braess paradoxon

A Braess paradoxon szemlélteti a rendszeroptimum és a használói egyensúly közti különbséget[4]. A paradoxon kapacitásbővítéssel csökkenteni az úthálózat teljesítményét, mivel a beiktatott új út megváltoztatja a Nash-egyensúlyt és növeli az össz eljutási időt a rendszerben.

Vegyük a 3.2 ábrán látható fiktív úthálózatot. Ebben a Start \rightarrow A és a B \rightarrow End éleken az eljutási idő a forgalom századrésze, $\frac{T}{100}$ valamint a Start \rightarrow B és a A \rightarrow End éleken konstans 45, A és B pontok között nincsen út.

Tételezzünk fel 4000-es közlekedési igényt Start és End között. A kiinduló állapotban szimmetria miatt mindkét útvonalon (Start \rightarrow A \rightarrow End , Start \rightarrow B \rightarrow End)



3.2. ábra. Braess paradoxon szemléltető példa

2000-es folyamot kapunk, az eljutási idő pedig $\frac{2000}{100} + 45 = 65$.

Kössük össze A és B pontokat egy 0 eljutási idejű éllel. Ekkor Start és A-B pontok között még 4000-es folyamnál is a Start \rightarrow A élen rövidebb, 40, az eljutási idő, minden utazó ezt fogja választani. Hasonlóan a B \rightarrow End élen is. Ezáltal a az eljutási idő minden utazó számára Start és End között $40 + 40 = 80$ lesz, ami magasabb, mint a korábbi eljutási idő.

Ezáltal látható, hogy az A és B pontokat összekötő él hozzáadása növelte az eljutási időket minden utazó számára a rendszerben.

3.2. Forgalomszimuláció

A közúti közlekedés összetett folyamatának modellezésével lehet a költséges és pontatlan méréseket egyszerűen, gyorsan és pontosan kivitelezni. A forgalomszimuláció hasonló feladat a folyadékáramlás szimulációjához. A folyadékok áramlásának modellezésére alapvetően kétféle megközelítés létezik, a folyadék egyedi részecskéinek mozgását és egymásra hatását modellező részecske szimuláció a Monte Carlo módszerek szerint[17], illetve a tér felosztásával nyert hálózaton a Navier-Stokes áramlástan differenciálegyenletek numerikus közelítésével nyert véges elem módszer[24].

A forgalom modellezésében az előbbi megközelítést mikroszkopikusnak, az utóbbit makroszkopikusnak nevezzük.

3.2.1. Mikroszkopikus megközelítés

A közúti forgalom alacsony szintű modellezése a mikroszkopikus megközelítés. A forgalom alapegységei a járművek, a modell ezek mozgását és viselkedését írja le azok fizikai és vezetőik pszichológiai jellemzői alapján[19].

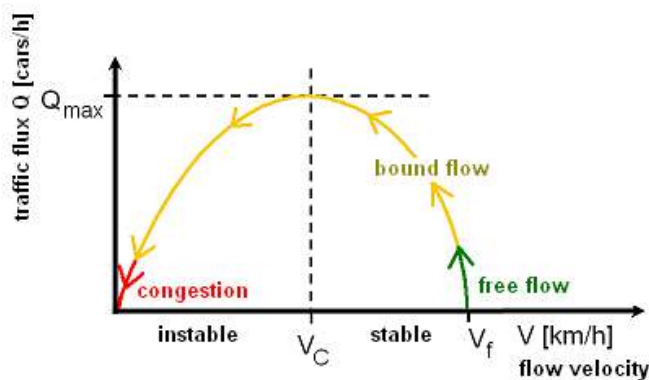
E megközelítés egyik fajtája a járműkövetési modell, ami azzal a feltételezéssel él, hogy a jármű mozgását csak az előtte haladó jármű befolyásolja. Analitikus függvények helyett diszkrét szabályok sokasága alkotja a modell alapját.

Az egyes járművek mozgásadatainak összegzéséből alakulnak ki a hálózatot jellemző forgalomtechnikai paraméterek.

3.2.2. Makroszkopikus megközelítés

A közúthálózat magas szintű, áramlástanai modelljét makroszkopikus modellnek nevezzük. E modell esetén a fundamentális diagram ad kapcsolatot a forgalomtechnikai paraméterek, a forgalom nagyság és a forgalomsűrűség között. A modelleket a felhasznált dinamika foka szerint lehet csoportosítani első-, másod- és harmadrendű modellekre attól függően, hogy csak a járműszám, vagy a sebesség és a sebesség szórásának a dinamikája is része a modellnek[19].

A fundamentális diagram tapasztalati összefüggést ábrázol a forgalom nagysága és sűrűsége között. Először Greenshields mutatott rá az összefüggésre 1935-ben mérései alapján[8].



3.3. ábra. A fundamentális diagram teremt összefüggést a forgalom nagyság és a forgalomsűrűség között

3.3. Forgalomtechnikai paraméterek mérése

Mint minden szabályzókörnek, a hatékony forgalomirányításnak is elengedhetetlen része a rendszer állapotainak – ez esetben a forgalom jellemzőinek – hatékony mérése. A mérés során célszerű a forgalomra jellemző mennyiségek pontos és aktuális értékeit mérni, melyeket lehet azonnali beavatkozásra felhasználni vagy aggregált adatként tervezésnél figyelembe venni.

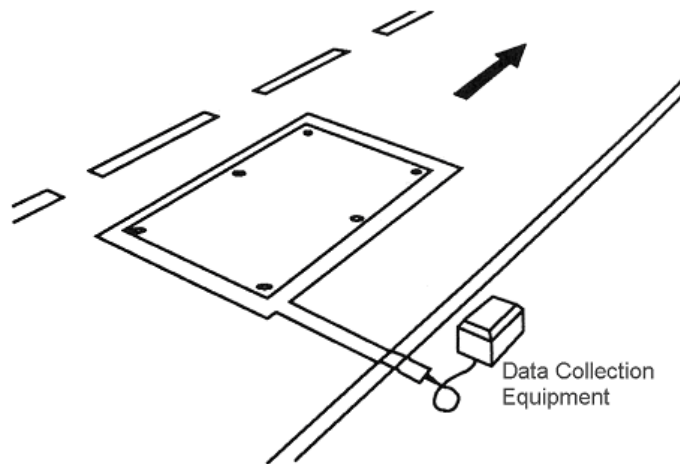
Technikailag kétféle mérési módszert különböztetünk meg, statikus és dinamikus. Statikusnak nevezzük, ha a mérőberendezés az úthálózat egy pontjához van rögzítve, az ott áthaladó forgalomról ad részletes, vagy összesített információt. A

dinamikus mérések a járművek egy csoportjáról adnak részletesebb információt az úthálózat nagyobb részét lefedve.

3.3.1. Keresztmetszeti mérés

A legelterjedtebb közlekedésmérnöki eszközök a különböző technológiák segítségével az út egy adott pontjában a forgalom több jellemzőjét vizsgáló keresztmetszeti mérési módszerek. Ennek legegyszerűbb változata a manuális forgalom számlálás, de az elektromágneses indukció elvén működő hurokdetektor, az ultrahangos jármű érzékelő és a bonyolult képelemzési algoritmusok sorozatát felvonultató kamerák is ebbe a kategóriába tartoznak.

Általánosan valamennyi technológia alkalmas a járműszám mérésére, ami az adott idő alatti forgalom nagyságának felel meg. Jelen dolgozathoz csak ezt a tulajdonságát használtam fel ezeknek a technológiáknak. Eszköztől függően alkalmasak lehetnek a sebesség, az időbeli foglaltság, a követési idő és a járműsűrűség mérésére is.



3.4. ábra. Hurokdetektor elhelyezése a közúton

Az indukciós hurokdetektor elhelyezését mutatja a 3.4 ábra. Ez egy, az aszfaltba mart horonyba fektetett elektromos hurok, amin a hozzá tartozó vezérlőelektronika folyamatosan mágneses teret indukál. A hurok felett elhaladó fém jármű megzavarja a keletkezett mágneses teret és ez a zavarás áramméréssel figyelhető meg a vezérlőberendezésben. Több berendezés a járműszámláláson felül a zavarás alakjából az elhaladó járművet kategorizálni is tudja.

3.3.2. Vonali mérés

Vonali mérésként tipikusan egy főbb útvonalon elhelyezett mérőeszköz sorozatot értünk. A városra jellemző arteriális útvonal forgalmát annak hosszában enged vizsgálni a mérőrendszer információt adva jellemző forgalomnagyságról, lokális és átlagsebességről, esetlegesen pedig a mért szakaszon található csomópontokban a fordulási arányokról, hogy mennyi jármű haladt tovább az útvonalon, illetve mennyi tért le róla.

A tipikus megvalósítása ennek a mérésnek az útvonal mentén elhelyezett rendszámfelismeréssel ellátott kamerahálózat, de pár városban már üzemel Bluetooth érzékelők serege is[7], amik az úton elhaladó járművekben található vezeték nélküli eszközök, például mobiltelefon vagy kihangosító, egyedi azonosítói alapján tudják megbecsülni a forgalom paramétereit.



3.5. ábra. Egy Route Hawk típusú forgalom figyelő kamera

A 3.5 képen látható eszköz Edinburgh városának egyik nagy forgalmú útján figyeli a járműveket. A forgalomirányító központ a kamerák adataiból számol utazási időket a város valamennyi arteriális útvonalán.

3.3.3. Hálózati mérés

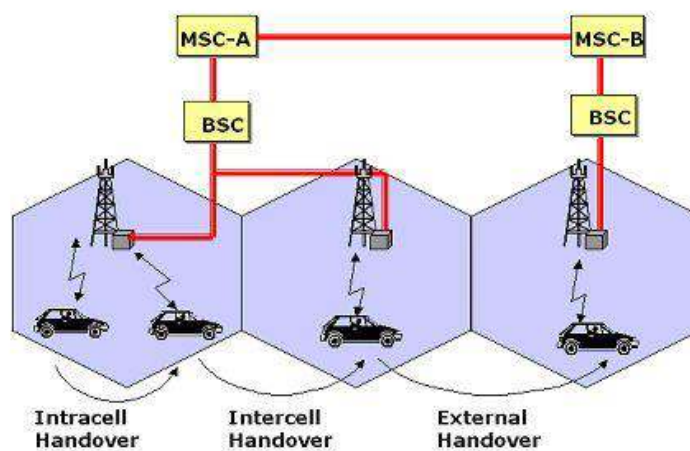
E mérésfajta esetén az egész közúti hálózatról kapható forgalmi információ. Ennek legjellemzőbb formája amikor egy-egy jármű út-idő trajektóriáját tudjuk rögzíteni a hálózaton.

Elterjedt módszer az úszó autós (Floating Car Data, FCD) forgalom figyelés, mikor a járművet felszerelik helymeghatározó eszközzel és annak a jelét rögzítik elég kis intervallumokkal ahhoz, hogy ne merüljön fel kétség a jármű útvonalát érintően. Ezzel több flottamenedzsment cég is foglalkozik mind Magyarországon, mind világszinten. A 3.6 ábrán egy budapesti mérés során begyűjtött adatsor vizualizációja



3.6. ábra. Úszó autós mérés rögzített pontjai Budapest belvárosában

látható a népszerű QGIS térinformatikai szoftverben.



3.7. ábra. Különböző típusú cellahatár átlépések a GSM hálózatban

A dolgozat szempontjából jelentős technológia a mobiltelefon hálózat felhasználása forgalommérésre. A celluláris felépítésű hálózat celláinak határán áthaladó mobiltelefon jelzést küld az operátornak, hogy már más cella területén található az eszköz, hol keresse azt a hálózat. A különböző fajta cella átlépéseket mutatja a 3.7 ábra[15]. Balról jobbra elsőként a cellán belüli átadást, ami szektorsugárzó antennák esetén fordul elő, majd azonos irányítón lévő két különböző cella közötti átadást szimbolizálja, végül pedig a különböző bázisállomás irányítók közötti átadási eseményt mutatja.

A cellaátlépés pillanatában a mobiltelefon helye és ideje elég pontosan behatározható[26]. Két cellaátlépés között pedig utazási idő számolható[18]. Két cellahatár átlépésének távolsága a helyi mobiltelefon hálózat kiépítettségétől függ, esetenként bizonytalan a jármű által bejárt útvonal a mért pontok között. Erre a bizonytalanságra az 5.4.2 rész szolgál megoldással.

4. fejezet

Problémafelvetés

A fejezet felvezetést nyújt a feladat mérnöki megközelítéséhez és rávilágít a vizsgált probléma szakmai gyökereire.

4.1. A célforgalmi mátrix meghatározásának nehézsége

A közlekedésmérnöki tervezés feladata a közlekedési igények kielégítése. Ehhez elengedhetetlen azok pontos és aktuális ismerete. A közlekedési igények leírásának eszköze a célforgalmi mátrix, mely kifejezi az úthálózat egyes régiói közötti forgalmakat.

A célforgalmi mátrix meghatározása hagyományosan mintavételezéses kikérdezés segítségével történik, mikor is a lakosság egy kis, de reprezentatív szeletének utazási szokásait és igényeit összeírják és ebből következtetnek a teljes lakosság igényeire[10].

Látszik, hogy a mérés rengeteg bizonytalansággal terhelt, a kis minta nem feltétlenül tud helyesen szolgálni az általánosítás alapjául, a kérdezett személy emlékezetből, bizonytalanul válaszolhat a kérdésekre, valamint az időbeliséget is nehezen lehet felmérni.

Az így kapott célforgalmi mátrix értéke pontatlan, a lakosság vagy a környezet változásait követni képtelen statikus adatként megfelelhet a közlekedés nagyobb volumenű tervezéséhez, de alkalmatlan a forgalom dinamikáját követő, gyors, valós idejű, alkalmazkodó forgalomirányításra.

4.2. Mobiltelefon alapú közlekedési igény meghatározás

A közlekedési igények aktuális és pontos meghatározására több modern technológia is nyújt lehetőséget. A mobiltelefonok alkalmazása a közlekedési igények felmérésében és a közlekedés segítésében kutatók széles körét foglalkoztatja.

A mobiltelefonok felhasználása új utakat nyithat meg a közlekedési és városrendezési módszerek között¹. A mobiltelefon felhasználása kétféle módon történhet.

A crowdsourcing az embertömegek bevonását jelenti a mérések jobbra és pontosabbá tételére. Több internetes szolgáltató, köztük például a Google vagy az Apple, gyűjt a felhasználói segítségével forgalmi adatokat, hogy azokkal támogassa meg a szolgáltatásait és nyújtson az ügyfeleinek kiemelkedő minőségű szolgáltatást a navigáció terén.

A másik megközelítés, hogy a mobiltelefon operátor a hálózat működtetéséből rendelkezésére álló információmorzsákból becsüli meg a forgalom paramétereinek aktuális értékét. Az így nyert aggregált adatsorok nem tartalmazzák az egyedi előfizetők adatait, csupán a hálózatról nyújtanak gyűjtő jellegű információt[26].

A mobiltelefonhálózat segítségével felmért igények és az azokra való gyors és szakszerű reagálásra követendő példa Isztambul városa, ahol a helyi operátorokkal együttműködik a városi közösségi közlekedést irányító szervezet, hogy az igényeket a lehető leghatékonyabban elégítse ki.

4.3. Adatfúzió

A változatos mérési technológiák adta adatsorok egyesítése és vizsgálata aktuális kérdés az egyre bővülő közlekedési és informatikai eszközök sűrűjében. Ezen térben, időben és megbízhatóságban heterogén jellegű információk megfelelő integrálásának problémája nyitott és aktuális kérdés. A dolgozat elsődleges célkitűzése a rendelkezésre álló adatok és a megbecsülni kívánt forgalmi paraméterek közötti kapcsolat vizsgálatának elősegítése matematikai eszközök és módszertan kialakításával.

¹http://www.ibm.com/smarterplanet/us/en/smarter_cities/overview/

5. fejezet

Modell

A feladat formalizálása következik alább.

5.1. Közút hálózat reprezentációja gráffal

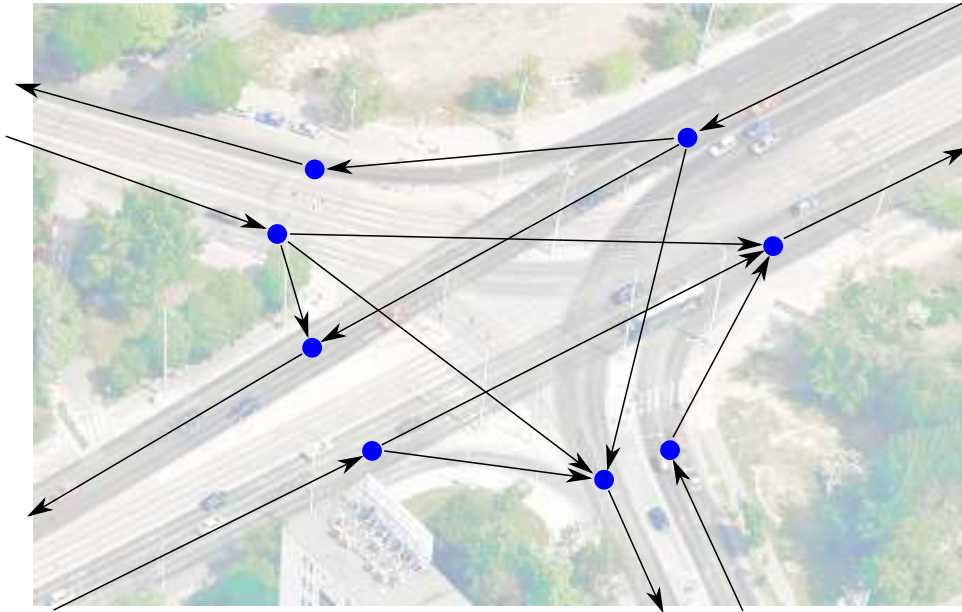
Az úthálózatból kialakítható gráf csúcsai a közúti csomópontok, az azok közötti útvonalakat pedig az élek reprezentálják.

Amennyiben egy csomópontban nem lehet minden irányba fordulni, az adott csomópont több csúcsra bontható, amik között a fordulási lehetőségeket a valóságnak megfelelően lehet élekkel kifejezni.

A közút hálózatot reprezentáló gráfot \mathcal{G} szimbólummal jelölöm a dolgozat további részében.



5.1. ábra. Bogdánfy utca és az Irinyi József utca kereszteződése



5.2. ábra. Az 5.1 ábrán feltüntetett csomópontból kialakított gráfrészlet

Az 5.1 és 5.2 ábrákon látható egy valós közúti csomópontból kialakított gráf részlet. Példának a Műegyetemtől nem messze található Bogdánfy és Irinyi József utcák forgalmas kereszteződését választottam, ahol a magas forgalom miatt korlátozták a fordulási lehetőségeket, például a Bogdánfy utcáról Karinthy Frigyes út felé.

5.2. Változók

A közlekedés egy időben és térben lezajló, összetett folyamat, mely állapotának pontos ismerete valamennyi, az úthálózaton tartózkodó jármű időfüggő helyzetének ismeretét jelenti. Ennek egyszerűsítése mellett döntöttem és redukáltam a rendszer állapotát közlekedési folyamatok időben állandó nagyságának ismeretére. A dinamikus, időben változó közlekedési folyamatok vizsgálatával a dinamikus ráterhelés témaköre foglalkozik.

A klasszikus közlekedésmérnöki forgalom felvétel régiókra bontja a vizsgált területet és a régiók közötti folyamat mértékét vizsgálja. A régiók közötti forgalom kiinduló és végpontja a régión belül bárhol lehet, a vizsgálat szempontjából érdektelen.

5.2.1. Lemma. *A ráterhelési feladat régiónkénti célforgalmi mátrixszal visszavezethető egy módosított úthálózatú ráterhelési feladatra, melyben a célforgalmi mátrix a csúcsok közötti forgalmat fejezi ki.*

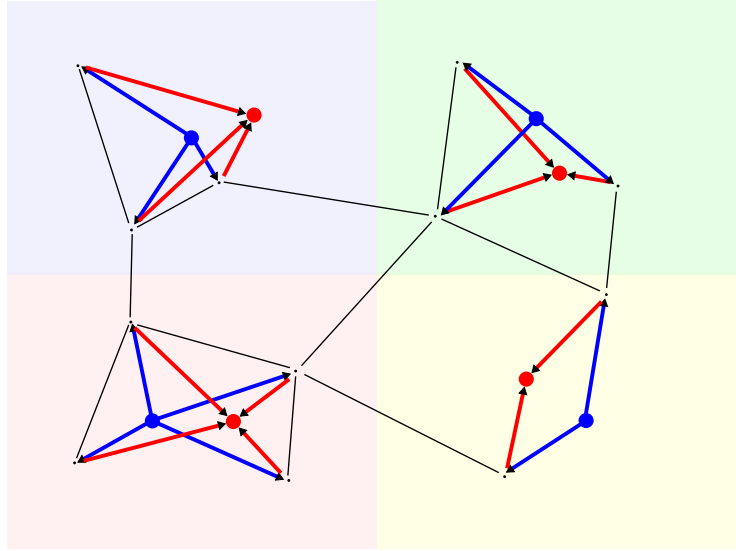
A régiók forrás és nyelő középpontjának régiónként egy-egy fiktív csúcsot felvéve és azokat megfelelően irányított, 0 súlyú élekkel a régió területén található

összes csomóponttal összekötve alkotható meg a módosított úthálózat. A korábbi célforgalmi mátrix az új csúcsok közötti célforgalmi mátrix lesz.

Ez a konstrukció nem befolyásolja az útvonal kereséseket az úthálózaton, mivel az új élek súlya nem befolyásolja az utak súlyát, az újonnan keletkező csúcsok pedig források és nyelők, útvonal azokon keresztül nem vezethet.

A konstrukciót az 5.3 ábra mutatja be, melyen az eredeti fekete úthálózat lett módosítva a kék és piros, forrás és nyelő csúcsok hozzáadásával.

Az 5.2.1 lemma fenti módon való alkalmazásával a célforgalmi mátrix nem változik, a feladat komplexitása csökken.



5.3. ábra. A közút hálózat régióinak kiegészítése virtuális forrás és nyelő csúcsokkal

$$F \subseteq \mathbb{V}(\mathfrak{G}), Ny \subseteq \mathbb{V}(\mathfrak{G}) \quad (5.1)$$

$$D = (D_{o,d} \mid o \in F, d \in Ny) \quad (5.2)$$

A vizsgált feladatban az úthálózatot leíró \mathfrak{G} gráf csúcshalmazának két részhalmaza az F források és Ny nyelők halmaza, amik között a célforgalmi mátrix felírható. A célforgalmi mátrixot D -vel, elemeit pedig $D_{o,d}$ -ként jelölöm.

Egy forrás és egy nyelő csúcs között több lehetséges útvonal létezhet a gráfnak megfelelően, ezeket sorban $p_{o,d}^k \mid o \in F, d \in Ny, k \leq K_{o,d}$ jelöléssel látom el, valamint jelölje \mathfrak{P} az összes útvonal halmazát.

$$\mathfrak{P} = \{p_{o,d}^k \mid o \in F, d \in Ny, k \leq K_{o,d}\} \quad (5.3)$$

A $p_{o,d}^k$ által jelölt útvonalon képződő forgalom folyam mértékét $f_{o,d}^k$ módon jelölöm. Valamint $K_{o,d}$ jelöli az o és a d csúcsok közötti lehetséges útvonalak számát.

$$K_{o,d} = |\{p \mid p \text{ út } \mathcal{G}\text{-on } o \text{ és } d \text{ között}\}| \quad (5.4)$$

A célforgalmi mátrix $D_{o,d}$ eleme az $o \in F$ csúcsból a $d \in Ny$ csúcsba tartó összes folyamatot jelenti.

$$D_{o,d} = \sum_{k \leq K_{o,d}} f_{o,d}^k \mid \forall o \in F, \forall d \in Ny \quad (5.5)$$

Az így meghatározott $f_{o,d}^k$ változók által kifeszített teret fogom vizsgálni a dolgozatomban.

5.3. Ellenállás függvény

Az egyes élek ellenállása vagy költsége függ a rajtuk kialakuló folyam mértékétől. Ezt a szakmai neve a Volume Delay Function, VDF, vagyis forgalomnagyság-késés függvény, mely a forgalomfüggő utazási időre utal[25]. Az útszakaszok ellenállása ugyanúgy a forgalomnagyság és a sebesség tapasztalati összefüggését fejezi ki, mint a 3.3 ábrán bemutatott fundamentális diagram, de más megközelítésből.

Egy útszakaszon az eljutási idő több tényezőtől függhet, úgymint az útszakasz hossza, geometriája, minősége és a rajta áthaladó forgalom nagyságától[27]. Ennek a leírására különböző iparági szabványok adottak, ezek közül mutatok be párat.

Az utazási idő függvények általános szerkezete

$$t(q) = t_0 \cdot f(sat) \quad (5.6)$$

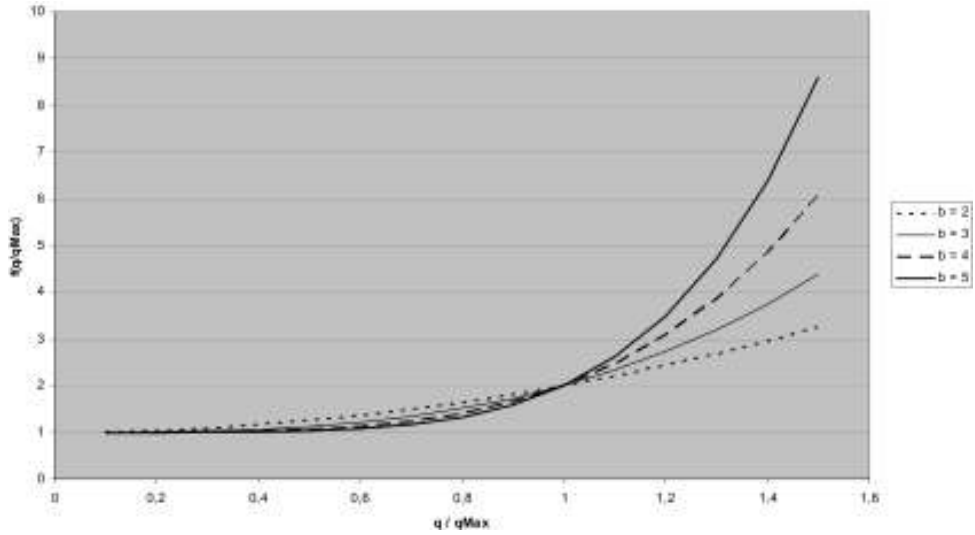
alakú, ahol t_0 az útszakaszon szabad áramlás közben való áthaladás idejét, a szabad áramlási utazási időt jelenti. Ez tipikusan a megengedett maximális sebességgel való haladást jelenti.

Az utazási idő függvény értéke a

$$sat = \frac{q}{q_{max}} \quad (5.7)$$

telítettségen keresztül függ a forgalomnagyságtól, ahol q_{max} az útszakasz elméleti, ideális maximális forgalomnagysága.

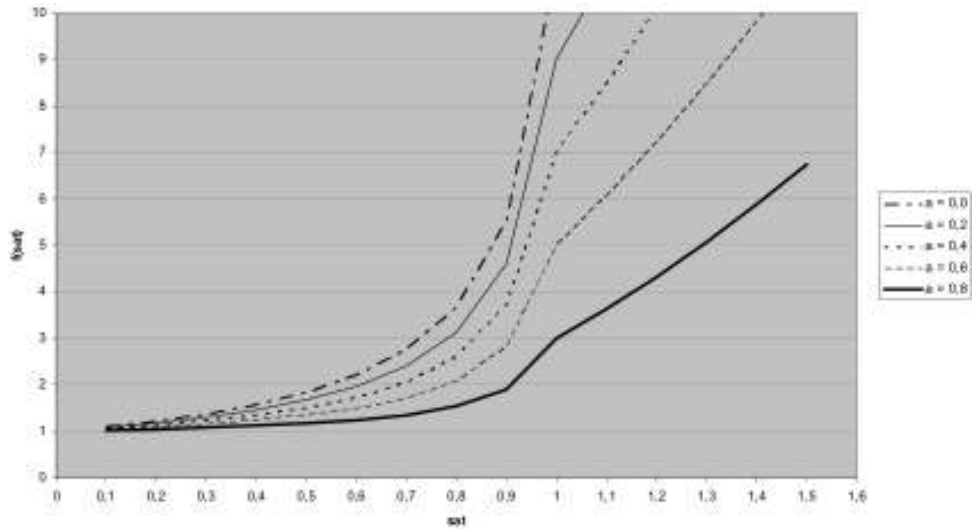
Az Egyesült Államok közúti közlekedéssel foglalkozó intézménye a Bureau of Public Roads által kiadott segédlet a ráterhelési feladatok számolásához a BPR nevű függvényt javasolja az utazási idő számolására[21].



5.4. ábra. A BPR görbe illusztrációja $a = 1$, $c = 1$, valamint többféle b paraméterrel

$$\text{BPR}(q) = t_0 \cdot (1 + a \cdot \text{sat}^b(q)) \quad (5.8)$$

A francia közlekedéskutatási intézet, az Institut National de Recherche sur les Transports et leur Sécurité, az INRETS alkalmazását ajánlja a ráterhelési számítások elvégzéséhez[13].



5.5. ábra. Az INRETS görbe illusztrációja $c = 1$ és többféle a paraméterrel

$$\text{INRETS}(q) = \begin{cases} t_0 \cdot \frac{1.1-a \cdot \text{sat}}{1.1-\text{sat}} & \text{sat} \leq \text{sat}_{crit} \\ t_0 \cdot \frac{1.1-a \cdot \text{sat}}{0.1} \cdot \text{sat}^2 & \text{sat} > \text{sat}_{crit} \end{cases} \quad (5.9)$$

Az INRETS számolásakor a sat_{crit} kritikus telítettség értékét 1-nek javasolja az intézmény.

A formulákban szereplő a és b paraméterek választását a fenti intézmények segédletei segítik, ezáltal lehetővé téve a modell finomhangolását.

A dolgozat folyamán ellenállásfüggvényként vagy eljutási, utazási időként hivatkozom az útszakaszokon vagy teljes útvonalon való végighaladás forgalomfüggő költségére. Ezt a függvényt az úthálózat $e \in \mathbb{E}(\mathfrak{G})$ élén $\tau_e(X_e)$ jelöléssel látom el, ahol X_e az 5.4 szakaszban leírt forgalomnagyság értéke az e útszakaszon.

5.4. Mérések

A közlekedésmérnöki gyakorlatban alkalmazott mérőrendszerek kétféle fajtáját és az általuk generált egyenleteket mutatom be a következő pontokban. Ezen mérések jellemzője, hogy a fentebb vázolt $f_{o,d}^k$ változók egy részhalmazára ad összefüggést.

5.4.1. Keresztmetszeti mérés

A leggyakrabban hurokdetektor eszközzel az úthálózat egy rögzített keresztmetszetében forgalomnagyság értéket lehet mérni, amit a járműmegmaradás feltételezésével a teljes élre ki lehet terjeszteni. Egy \mathfrak{G} -beli e él forgalomnagyságát X_e szimbólummal jelölöm és értéke a rajta áthaladó utak forgalom nagyságainak összege.

$$X_e = \sum_{\substack{p_{o,d}^k \in \mathfrak{P} \\ p_{o,d}^k \cap e \neq \emptyset}} f_{o,d}^k \mid e \in \mathbb{E}(\mathfrak{G}) \quad (5.10)$$

5.4.2. Hálózati mérés

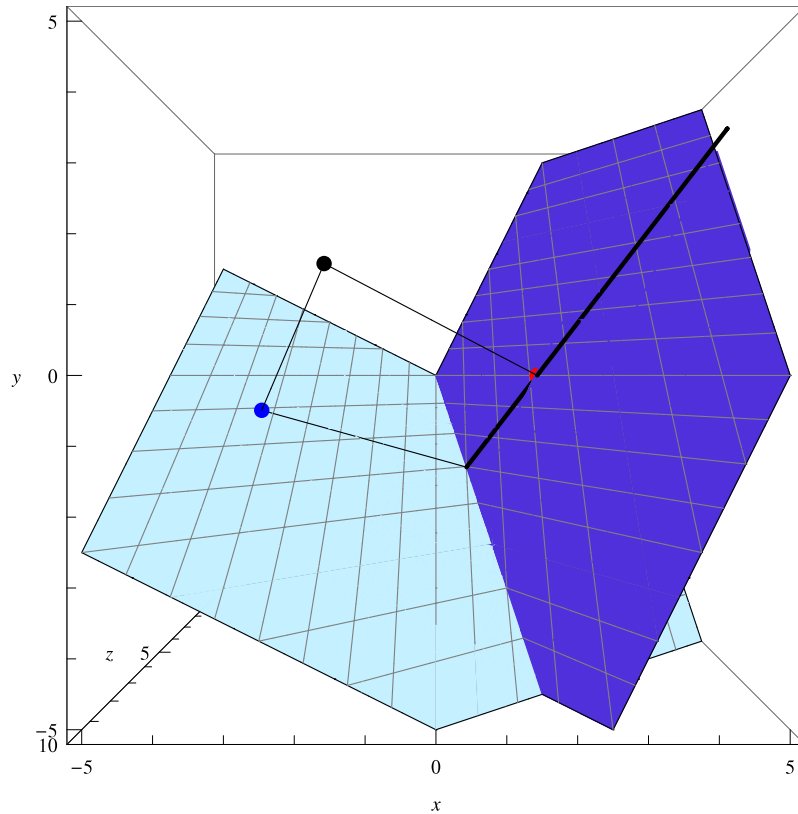
A klasszikus keresztmetszeti mérések mellett terjedőben vannak a vonali vagy hálózati mérést lehetővé tevő technológiák az utakon. Ezek jellemzője, hogy nem az út egy adott keresztmetszetében adnak a forgalomra jellemző értéket, hanem útvonalakra.

Ahogy azt a 3.3.3 részben leírtam, a jellemző formája a hálózati méréseknek egy jármű trajektória sűrűbb vagy ritkásabb mintavételezése. Az egyes minták között a jármű útja ismeretlen, a 3.1.5 részben hivatkozott felhasználói egyensúly alapján valamennyi lehetséges útvonalon azonos az eljutási idő a trajektória két pontja között.

Így egy hálózati mérés útvonalanként egy-egy egyenletet jelent.

5.5. A célforgalmi mátrix kiszámolhatósága

A rendelkezésre álló mérések alapján célforgalmi mátrix kiszámíthatósága a kérdés. A teljes rendszer folyamváltozóinak megállapítása nem szükséges ahhoz, hogy a célforgalmi mátrix elemeit ki lehessen számolni. Mivel a mátrix elemei lineáris kombinációi a folyam változóknak, elég megvizsgálni a mérések által kifeszített lineáris altér és a célforgalmi mátrix elemei által kifeszített altér viszonyát.



5.6. ábra. A célforgalmi mátrix elemei és a mérések által kifeszített alterek kapcsolatának illusztrációja

Az 5.6 ábra illusztrálja, hogy az $f_{o,d}^k$ folyamatok által kifeszített $|\mathfrak{P}|$ dimenziós valós térben a világos színnel jelölt alteret generálják a mérések. A fekete ponttal jelölt pont szimbolizálja a valóságot, annak kézzel jelölt vetülete a mérések alterére ami ismert és mérhető.

A célforgalmi mátrix elemei által generált alteret a sötétkék sík jelképezi, amin a valóság, piros színnel jelölt, vetületét a feladat meghatározni. Ha a célforgalmi altér nem esik bele a mérések alterébe, akkor a kívánt információ nem határozható meg egyértelműen, azaz marad szabadságfoka.

Az ábrán vastag fekete vonallal jelölt egyenes a célforgalmi mátrix által felvehető értékek altere a mérések ismeretében.

6. fejezet

Implementáció

Ebben a fejezetben a használt informatikai eszközök és a használatuk módjának bemutatása, valamint az elkészített programok és az azok elkészítéséhez alkalmazott matematikai háttér kerül tárgyalásra.

6.1. Eszközök

A matematikai modell kidolgozása után a konkrét implementáció (Matlab és Mathematica) két elterjedt matematikai programcsomag segítségével készítettem el, melyek alkalmasak mind egyszerűbb, mind komolyabb matematikai műveletek elvégzésére.

6.1.1. Matlab

A Matlab a MathWorks által fejlesztett numerikus számításokat segítő programcsomag, mely különböző modulokkal bővíthető és igazítható hozzá feladatok széles skálájához. Ilyen kiegészítő modul a földrajzi adatok feldolgozását elősegítő széleskörűen elterjedt Mapping Toolbox[20].









6.1.2. Mathematica

A számítástechnika rohamos fejlődése szélesre tárja a lehetőségeket a szimbolikus és numerikus matematika alkalmazásai előtt a kutatásokban. A Wolfram Research Institute által fejlesztett Mathematica szoftvercsomag egy kimagaslóan összetett, egyszerűen használható, kiváló megjelenítési lehetőségeket biztosító megoldás a kor kutatásainak segítésére[32].

A Mathematica 8-as változatával a kombinatorikai és gráfalgoritmusok egy része a kernel része lett, nem szükséges hozzá a különálló *Combinatorica* programcsomag

betöltése[30].

Alább mutatom be a Mathematica széles repertoárjából az általam használt, gráfokkal kapcsolatos eszközöket.

DirectedEdge $[u, v]$, **UndirectedEdge** $[u, v]$ függvények egy irányított vagy irányítatlan élet állítanak elő, aminek két végpontja a két paraméter, u és v . Egyszerűsített bevitelére a Mathematica felhasználói felületén az     vagy     billentyűsorozat ad lehetőséget.

Graph $\{e_1, e_2, \dots\}$ előállítja az átadott élhalmaz által generált gráfot. A csúcshalmazát külön is meg lehet adni, ami szükséges lehet abban az esetben, ha vannak 0 fokú csúcsok is a gráfban.

EdgeQ $[g, e]$ egy logikai függvény, ami az e él létezésére kérdez rá a g gráfban.

EdgeDelete $[g, e]$ új gráfot állít elő az e él törlésével a g gráfból. Az esetlegesen keletkező 0 fokú csúcsok részei maradnak az eredmény gráfnak.

VertexDelete $[g, v]$ új gráfot állít elő a v csúcs és a hozzá kapcsolódó élek törlésével a g gráfból.

FindShortestPath $[g, s, t]$ előállítja a legrövidebb útvonalat a g gráf mentén s és t csúcsok között. A g élsúlyaitól függően az alkalmazható leghatékonyabb algoritmust választja a Mathematica a függvény futásához. Az s és t paraméterek helyére g egy-egy csúcsát, vagy az *All* konstans értéket is lehet helyettesíteni. Ekkor az összes csúcshoz előállítja a legrövidebb utak listáját.

PropertyValue $\{obj, item\}, name]$ egy obj objektum $item$ elemének megnevezett tulajdonságának értékét adja vissza. Nem találtam egyszerűbb, elegánsabb módszert egy gráf élsúlyainak lekérdezésére, mint ennek a függvénynek az alkalmazását. A g gráf e élének súlyát az **PropertyValue** $\{g, e\}, EdgeWeight]$ függvényhívással kérdeztem le az elkészített programban.

6.2. A használói egyensúly átfogalmazása optimalizálási feladattá

A 3.1.5 szakaszban bemutatott használói egyensúly vagy optimum ekvivalens kifejezése, hogy az eljutási idők azonosak egy-egy adott forrás és nyelő pár között vezető minden lehetséges és igénybe vett útvonalán, valamint az igénybe nem vett

utak eljutási ideje magasabb az előbbieknél. Ezt írják le formalizálva a 6.1 – 6.3 egyenletek.

$$D_{o,d} = \sum_{k \leq K_{o,d}} f_{o,d}^k \mid \forall o \in F, \forall d \in Ny \quad (6.1)$$

$$f_{o,d}^k \geq 0 \mid \forall o \in F, \forall d \in Ny, k \leq K_{o,d} \quad (6.2)$$

$$f_{o,d}^k \cdot \left(\tau_{o,d}^k - \min_{l \leq K_{o,d}} \tau_{o,d}^l \right) = 0 \quad (6.3)$$

Ahol $\tau_{o,d}^k$ az $p_{o,d}^k$ úton vett eljutási idő, azaz

$$\tau_{o,d}^k = \sum_{\substack{e \in \mathbb{E}(\mathfrak{G}) \\ p_{o,d}^k \cap e \neq \emptyset}} \tau_e(X_e). \quad (6.4)$$

A 6.3 egyenlet fejezi ki, hogy amennyiben egy útvonalon a folyam értéke meghaladja a 0-t, azaz az adott útvonal igénybe van véve, annak utazási ideje megegyezik a minimális az adott kezdő és végpont közötti valamennyi útvonal utazási idejének minimumával.

6.2.1. Lemma. *Az egyensúlyi állapotot jelentő a 6.1, 6.2 és 6.3 egyenletek megoldása megegyezik a következő optimalizálási feladat megoldásával[6].*

$$C(\mathbf{x}) = \sum_{e \in \mathbb{E}(\mathfrak{G})} \int_0^{X_e} \tau_e(x) dx \quad (6.5)$$

Legyen $C(\mathbf{x})$ a célfüggvény, aminek az értékét minimalizálni kell az $X_e, e \in \mathbb{E}(\mathfrak{G})$ útszakaszok forgalmainak tere felett az alábbi feltételekkel:

$$f_{o,d}^k \geq 0 \mid \forall o \in F, \forall d \in Ny, k \leq K_{o,d} \quad (6.6)$$

$$D_{o,d} = \sum_{k \leq K_{o,d}} f_{o,d}^k \mid \forall o \in F, \forall d \in Ny \quad (6.7)$$

$$X_e = \sum_{\substack{p_{o,d}^k \in \mathfrak{P} \\ p_{o,d}^k \cap e \neq \emptyset}} f_{o,d}^k \mid e \in \mathbb{E}(\mathfrak{G}) \quad (6.8)$$

A 6.2.1 Lemma belátásához felírható a 6.5 – 6.8 egyenletek Lagrange függvénye:

$$L(\mathbf{f}, \mathbf{u}) = C(X(\mathbf{f})) + \sum_{\substack{o \in F \\ d \in Ny}} u_{o,d} \left(D_{o,d} - \sum_{k \leq K_{o,d}} f_{o,d}^k \right) \quad (6.9)$$

A Lagrange-függvény kifejezéséhez az útszakaszokon lévő forgalomnagyság a 6.8 egyenlet által lett kifejezve a folyamváltozók segítségével.

A Lagrange-függvény szélsőértéke az alábbi módon határozható meg:

$$f_{o,d}^k \cdot \frac{\partial L(\mathbf{f}, \mathbf{u})}{\partial f_{o,d}^k} = 0 \quad \forall o \in F, \forall d \in Ny, \forall k \leq K_{o,d} \quad (6.10)$$

$$\frac{\partial L(\mathbf{f}, \mathbf{u})}{\partial f_{o,d}^k} \geq 0 \quad \forall o \in F, \forall d \in Ny, \forall k \leq K_{o,d} \quad (6.11)$$

$$\frac{\partial L(\mathbf{f}, \mathbf{u})}{\partial u_{o,d}} = 0 \quad \forall o \in F, \forall d \in Ny \quad (6.12)$$

A 6.5 egyenletből kiindulva meghatározható $\frac{\partial L(\mathbf{x})}{\partial X_e} = \tau_e(X_e)$.

$$\frac{\partial C(\mathbf{x})}{\partial X_e} = \frac{\partial}{\partial X_e} \sum_{f \in \mathbb{E}(\mathfrak{G})} \int_0^{X_f} \tau_f(x) dx = \tau_e(X_e) \quad (6.13)$$

És ezáltal felírható a célfüggvény $f_{o,d}^k$ szerinti differenciája.

$$\frac{\partial C(\mathbf{x}(\mathbf{f}))}{\partial f_{o,d}^k} = \sum_{e \in \mathbb{E}(\mathfrak{G})} \frac{\partial C(\mathbf{x}(\mathbf{f}))}{\partial X_e} \frac{\partial X_e}{\partial f_{o,d}^k} \quad (6.14)$$

Ahol $\frac{\partial X_e}{\partial f_{o,d}^k}$ értéke 1, ha az e él része az $p_{o,d}^k$ útnak, egyébként pedig 0 a 6.8 egyenlet alapján. Tehát

$$\frac{\partial C(\mathbf{x}(\mathbf{f}))}{\partial f_{o,d}^k} = \tau_{o,d}^k. \quad (6.15)$$

Ezáltal az alábbi egyenletre jutunk,

$$\frac{\partial L(\mathbf{f}, \mathbf{u})}{\partial f_{o,d}^k} = \tau_{o,d}^k - u_{o,d}, \quad (6.16)$$

amit felhasználva a Lagrange-függvény szélsőértékére az alábbi feltétel kapható:

$$f_{o,d}^k \cdot \tau_{o,d}^k - u_{o,d} = 0 \quad \forall o \in F, \forall d \in Ny, \forall k \leq K_{o,d} \quad (6.17)$$

$$\tau_{o,d}^k - u_{o,d} \geq 0 \quad \forall o \in F, \forall d \in Ny, \forall k \leq K_{o,d} \quad (6.18)$$

$$f_{o,d}^k \geq 0 \quad \forall o \in F, \forall d \in Ny, \forall k \leq K_{o,d} \quad (6.19)$$

$$\sum_{k \leq K_{o,d}} f_{o,d}^k = D_{o,d} \quad \forall o \in F, \forall d \in Ny \quad (6.20)$$

A 6.18 egyenlet szerint az $u_{o,d}$ Lagrange multiplikátor értéke minden esetben kisebb vagy egyenlő, mint az eljutási idők o és d közötti különböző útvonalakon és a 6.17 szerint az igénybe vett útvonalakon megegyezik az eljutási idővel, ebből következik, hogy $u_{o,d} = \min_{k \leq K_{o,d}} \tau_{o,d}^k$. Tehát a két feltételrendszernek azonos a megoldása[6]. ■

6.3. Utak meghatározása

A F forrás és Ny nyelő csomópontok közötti folyamok meghatározásához elengedhetetlen az ezen csomópontok közötti utak meghatározása a közúti hálózat mentén. Kisebb hálózatokra a naiv összes út kereső algoritmusok megfelelőhetnek, de komolyabb méretű úthálózatra már túl költséges egyszerű algoritmusokkal meghatározni a lehetséges útvonalakat.

Az összes útvonal meghatározása két pont között nem csak időigényes feladat, hanem szükségtelen is, mivel a lehetséges útvonalak halmazának tekintélyes része feleslegesen nagy kitérőt jelent az utazók számára. Az úthálózat részletes vizsgálatával szűkíthető az útvonalak halmaza az utazók által ténylegesen igénybe vett útvonalakra.

A potenciálisan használt útvonalak halmazának előállítását az első k -legrövidebb útvonal kiválasztásával oldottam meg, ahol k értéke forrás és nyelő pár függő is lehet, az alkalmazott stratégia feladatfüggő, kiválasztása egyedi vizsgálatot igényel.

6.3.1. Yen-féle k -legrövidebb út algoritmus

Yen algoritmus használható nem negatív élsúlyú gráfban $s \in \mathbb{V}(\mathcal{G})$ forrás és $t \in \mathbb{V}(\mathcal{G})$ nyelő csúcs között a $k \in \mathbb{Z}$ darab legrövidebb, hurokmentes út megtalálására. Yen 1971-ben publikálta az algoritmust, ami a két csúcs között megtalálja a legrövidebb utat és annak $k - 1$ variációját[31].

Jelölések

Az algoritmus k egymás utáni lépésben nyilvántartja és frissíti a lehetséges rövid utak listáját. A felhasznált jelölések következnek alább.

s, t a keresett útvonalak kezdő és végpontja amiket a forráskódban source és sink jelöl.

origGraph az eredeti gráf, amit vizsgálunk.

graph a minden ciklusban vizsgált, az origGraph-ból előállított részgráf, amin újabb utakat keresünk.

A a legrövidebb utak listája, amibe minden ciklus végén új elem kerül.

B a következő legrövidebb útra jelölt utak listája.

spoorNode az előző legrövidebb útvonal i -ik csúcsa, ahonnan kiinduló új utat keresünk a célig.

rootPath az előző legrövidebb útvonal aktuális spoorNode csúcsáig tartó részútja.

spurPath a spoorNode-tól a sink csúcsig előálló új útvonal.

totalPath a rootPath és a spurPath utak kompozíciójából előálló új út source és sink között.

Leírás

Yen algoritmusát a k darab legrövidebb út meghatározására a Mathematica programcsomagban implementáltam, annak részletes bemutatása következik alább.

```
1 YenKSP[origGraph_, source_, sink_, K_] :=
```

Az algoritmus bemeneti paramétere a origGraph gráf, source forrás és sink nyelő csúcsok valamint K paraméter, kimenete pedig a k darab legrövidebb út listája a gráfon a két csúcs között.

```
2 Module[{A = {FindShortestPath[origGraph, source, sink]}, B = {}},  
3 spurNode, rootPath, totalPath, spurPath, graph = origGraph},
```

Az algoritmus első lépése meghatározni a legrövidebb útvonalat source és sink csúcsok között és ezt eltárolja a kiválasztott utak A listájának első elemeként. Ez látható az A.1 állomány második sorában.

Ezután következik a frissítő ciklus, melyben az előző legrövidebb út valamennyi csúcsából megvizsgálja, hogy lehet-e azonos kezdő útvonallal, de más befejező útvonallal eljutni a sink csúcsba, ami az A.1 állomány 4–29 sorát lefedi.

```
6 spurNode = Last[A][[i];  
7 rootPath = Last[A][[1 ;; i];
```

A belső ciklus legelején lesz beállítva spoorNode és a rootPath kezdő útvonal értéke.

```
9 Do[  
10 If[Length[rootPath] < Length[path]  
11 ^ rootPath == path[[1 ;; i]  
12 ^ EdgeQ[graph, path[[i] → path[[i + 1]],  
13 graph = EdgeDelete[graph,  
14 path[[i] → path[[i + 1]]  
15 , True]  
16 , {path, A}];
```

A belső ciklus megcsonkítja a vizsgált gráfot, hogy más útvonalakat is meg lehessen találni. Ehhez először az eddigi összes megtalált útvonal spoorNode utáni élét törli a gráfból, aminek a kezdőszelete spoorNode csúcsig megegyezik az aktuálisan vizsgált, legutóbb megtalált útnak rootPath kezdőszeletével. Ezzel elkerülhető, hogy az új út korábban már bejárt utak mentén haladhasson tovább.

```
17     graph = VertexDelete [graph , Most@rootPath ];
```

Majd a tizenhetedik sorban a spurNode előtti csúcsát törli a gráfból, hogy a keresett új útvonal ne fordulhasson vissza a rootPath irányába ezzel hurkot okozva.

```
18     spurPath = FindShortestPath [graph , spurNode , sink ];
```

A következő lépés a kapott, módosított graph gráfban útvonalkeresés a sink csúcs-hoz.

```
19     If [spurPath == {} , True ,
20         totalPath = Join [Most@rootPath , spurPath ] ;
21     B = Append [B , totalPath ] ;
```

Amennyiben létezik kerülőút a sink csúcshoz a spurNode csúcsból, az út rootPath kezdőszeletét és a számított spurPath kerülőutat kombinálva adódó totalPath utat a B listához adva bővíthető a lehetséges legrövidebbnek jelölt útvonalak köre.

```
22     B = DeleteDuplicates [B ] ;
23     B = SortBy [B ,
24         Plus @@ MapThread [
25             PropertyValue [{ graph , #1 → #2 } ,
26                 EdgeWeight ] & , {Most@# , Rest@#} ] & ;
```

Majd a belső ciklusból kilépve a B listát úthossz szerint rendezve és a duplikátumokat eltávolítva megkapjuk a jelölt útvonalak rendezett listáját.

```
27     If [B == {} , Break [] , True ] ;
```

A rákövetkező, huszonhetedik sor a B listát ellenőrzi, ha az nem tartalmaz már jelölt útvonalakat, akkor nincs több lehetséges útvonal a source és sink csúcsok között. Ekkor megszakad a külső ciklus is a K érték elérése előtt.

```
28     A = Append [A , First@B ] ;
29     B = Rest [B ] ;
```

A külső ciklus végén, ami K alkalommal fut le, a rendezett B lista első elemét kivesszük belőle és hozzáadjuk az A listához, mint a következő legrövidebb utat.

```
30     A ] ;
```

Végül az utolsó sor egy technikai részlet, ami az A listát adja meg a Module visszatérési értékének.

A függvény teljes forráskódját az A appendix tartalmazza.

Tárigény

A gráf éleinek, a legrövidebb utakat tartalmazó A listának valamint a lehetséges utakat tartalmazó B listának a tárolásához $\mathcal{O}(|\mathbb{V}(\mathfrak{G})|^2 + k \cdot |\mathbb{V}(\mathfrak{G})|)$ memória

szükséges. Legrosszabb esetben a gráf teljes gráf, így $|\mathbb{V}(\mathfrak{G})|^2$ él tárolása szükséges. A és B listák tárolásához legfeljebb $k \cdot |\mathbb{V}(\mathfrak{G})|$ tár szükséges, mivel legfeljebb k darab utat kell tárolni, amik hossza legfeljebb $|\mathbb{V}(\mathfrak{G})|$ lehet[31].

Időigény

Az algoritmus időbeli bonyolultsága a felhasznált legrövidebb utat kereső algoritmus futásidejének függvénye. Nemnegatív élsúlyok esetén a hatékony Dijkstra algoritmust tételezzük fel, aminek az időkorlátja $\mathcal{O}(|\mathbb{V}(\mathfrak{G})|^2)$, vagy Fibonacci kupac alkalmazása esetén $\mathcal{O}(|\mathbb{E}(\mathfrak{G})| + |\mathbb{V}(\mathfrak{G})| \cdot \log(|\mathbb{V}(\mathfrak{G})|))$. A Yen algoritmus $k \cdot |\mathbb{V}(\mathfrak{G})|$ alkalommal hívja meg a legrövidebb útkereső rutint, így az algoritmus futásideje $\mathcal{O}(k \cdot |\mathbb{V}(\mathfrak{G})| \cdot (|\mathbb{E}(\mathfrak{G})| + |\mathbb{V}(\mathfrak{G})| \cdot \log(|\mathbb{V}(\mathfrak{G})|)))$ [31].

6.3.2. K paraméter megválasztása

A k paraméter meghatározása, hogy forrás, nyelő páronként hány alternatív útvonalat veszünk figyelembe külön vizsgálat tárgyát képezheti. Lehet egy konstans érték, amit a valós úthálózatok jellemzői alapján lehet megállapítani, de az útszámoló algoritmus inkrementális jellegéből fakadóan akár egy abszolút vagy relatív hossz növekedési határ is megállapítható.

A Yen-féle algoritmus leállási feltétele könnyen módosítható így egyéni feltételeknek is megfelelő relatív k érték állapítható meg a teljes úthálózat gráfját és az adott csomópontok jellegét figyelembe véve.

6.4. A kifeszített alterek meghatározása

A figyelembe vett utak meghatározása után pár segédfüggvényt készítettem, amik a mérések illetve a célforgalmi mátrix elemei által kifeszített alterek mátrixát határozzák meg.

```

1 EdgePathMx[g_, paths_] :=
2   Boole@Table[
3     MemberQ[EdgeList[path], edge], {edge, EdgeList[g]}, {path, paths}]

```

Listing 6.1. Él-út mátrix meghatározása

$$X_e = \sum_{\substack{p_{o,d}^k \in \mathfrak{P} \\ p_{o,d}^k \cap e \neq \emptyset}} f_{o,d}^k \mid e \in \mathbb{E}(\mathfrak{G}) \quad (6.21)$$

A 6.1 állományban mutatott segédfüggvény, a EdgePathMx, az élek és az utak lineáris kapcsolatát kifejező mátrixot állítja elő, ami így megfelel az éleken forga-

lomnagyságokat nyújtó mérőrendszer által kifeszített altérnek a folyamok tere felett, ahogyan azt a 6.21 egyenlet kifejezi.

```

1 Source [g_] := Map[# [1] &,
2   Select [Transpose [{ VertexList [g], VertexInDegree [g] }], #[2] == 0 &]]
3 Sink [g_] := Map[# [1] &,
4   Select [Transpose [{ VertexList [g], VertexOutDegree [g] }], #[2] == 0 &]]

```

Listing 6.2. Egy gráf forrás és nyelő csúcsainak meghatározása

A 6.2 állományban található két segédfüggvény egy gráf nyelő és forrás csúcsait válogatja ki a foksámok alapján. Tipikusan az útvonal gráfok kezdő és végpontjainak kijelölésére lettek használva.

```

1 OdPairs [paths_] := Select [
2   Flatten [Outer [
3     {#1, #2} &,
4     DeleteDuplicates [Flatten [Map [Source, paths], 1]],
5     DeleteDuplicates [Flatten [Map [Sink, paths], 1]]],
6   , 1],
7   Not@SameQ[# [1], #[2]] &]

```

Listing 6.3. Forrás–nyelő párok előállítása adott útvonalakra

A forrás, nyelő párokat előállító függvény a 6.3 állományban kerül bemutatásra, ami az átadott útvonalgráfokon végigiterálva, azok kezdő és végpontjaiból készített külső szorzatot szűri le a megfelelő párok megkeresése végett.

```

1 OdPairPathMx [g_, paths_] :=
2   Boole@Table [{ First@Source [path], First@Sink [path] } == pair,
3     { pair, OdPairs [paths] }, { path, paths }]

```

Listing 6.4. Forrás-nyelő párok és utak kapcsolatát leíró mátrix meghatározása

$$D_{o,d} = \sum_{k \leq K_{o,d}} f_{o,d}^k \mid \forall o \in F, \forall d \in Ny \quad (6.22)$$

A 6.4 állomány tartalma, az OdPairPathMx nevű segédfüggvény, ami adott útvonalakra meghatározza a $(o, d) \in F \times Ny$ pár és az utakon realizálódó folyamok közötti összefüggést a 6.22 egyenlet szerint. A mátrix egy sora összegzi az adott kezdő és végpontok közötti összes útvonal forgalomnagyságát, ezáltal a mátrix a célforgalmi mátrix elemei által kifeszített alteret határozza meg.

```

1 RankOfEdgesODPath [g_, paths_] :=
2   MatrixRank@Join [OdPairPathMx [paths], EdgePathMx [g, paths]]

```

Listing 6.5. A célforgalmi mátrix elemei és a mérések által kifeszített altér dimenziójának számolása

Végül a 6.5 állományban mutatott segédfüggvény az előzőleg meghatározott mátrixok által közösen kifeszített altér rangját számolja ki, hogy így legyen összevethető a két altér viszonya, egymástól való függésük és kiszámolhatóságuk.

7. fejezet

Validáció

A fejezet az előzőekben bemutatott és implementált algoritmusok tesztfuttatásaival foglalkozik, azok helyességét vizsgálja.

7.1. Eszközök

A közlekedésmérnöki kutatások elvégzésekor gyakorta van szükség a gyakorlatban költséges, vagy kivitelezhetetlen forgalomtechnikai mérésekre. Ezek kiváltására iparágilag elfogadott szimulációs szoftvereket szoktak használni, amik részletekbe menően imitálják a valódi forgalom alakulását.

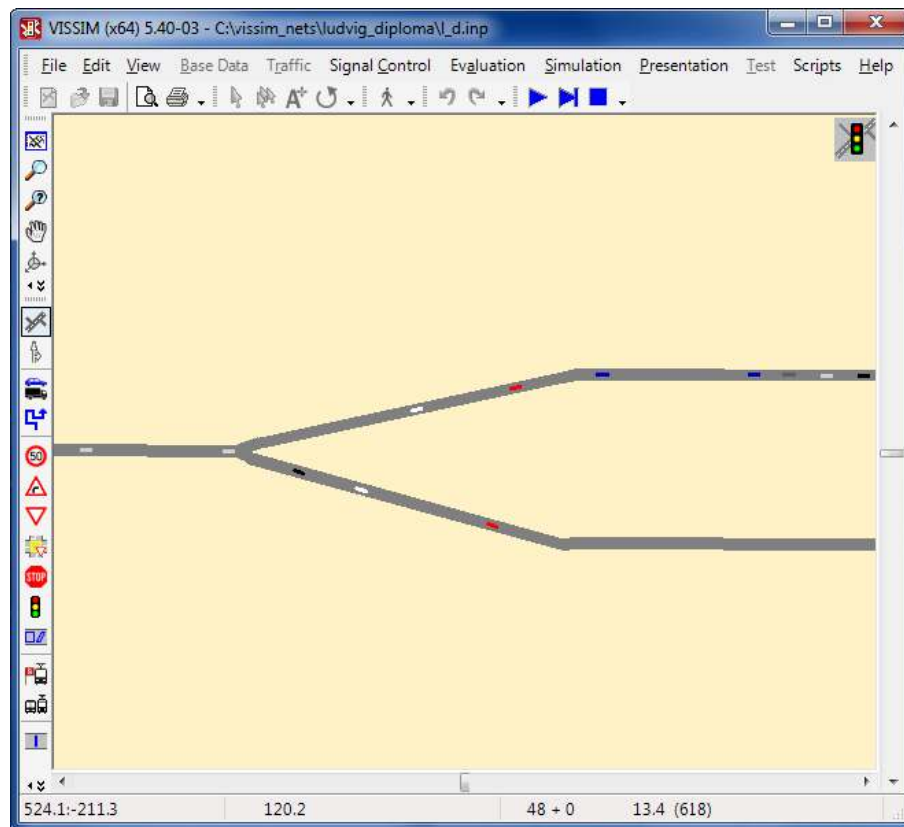
7.1.1. VISSIM

A német PTV Ag. által készített VISSIM egy mikroszkopikus, diszkrét idejű, viselkedés alapú forgalomszimulációs szoftvercsomag, amit elsősorban a városi multimodális közlekedés jármű és gyalogos áramlatainak szimulációjára terveztek[28].

A szoftver az egyéni és a közösségi közlekedés vizsgálatát teszi lehetővé különféle útgeometria, forgalom összetétel, jelzőlámpa konfiguráció és még számos paraméterben eltérő forgalmi helyzet variációk összevetésével. Így az alternatívák hatékonysága összehasonlíthatóvá válik és a program a döntés előkészítés kiváló eszköze[28].

A szoftver a felhasználói felületen is széles skáláját nyújtja a szimuláció során lekérdezhető, fájlba vagy adatbázisba menthető forgalmi paramétereknek. De program rendelkezik egy összetett és fejlesztőbarát programozási felülettel is, ahol a szimuláció legapróbb paraméterei is mind lekérdezhetőek vagy akár befolyásolhatóak[28].

A program tipikus kezelői felületét a 7.1 ábra mutatja. Az egyes járműveket apró színes foltok jelképezik a közúthálózaton.



7.1. ábra. A VISSIM mikroszkopikus forgalom szimulációs szoftver kezelőfelülete

7.1.2. VISUM

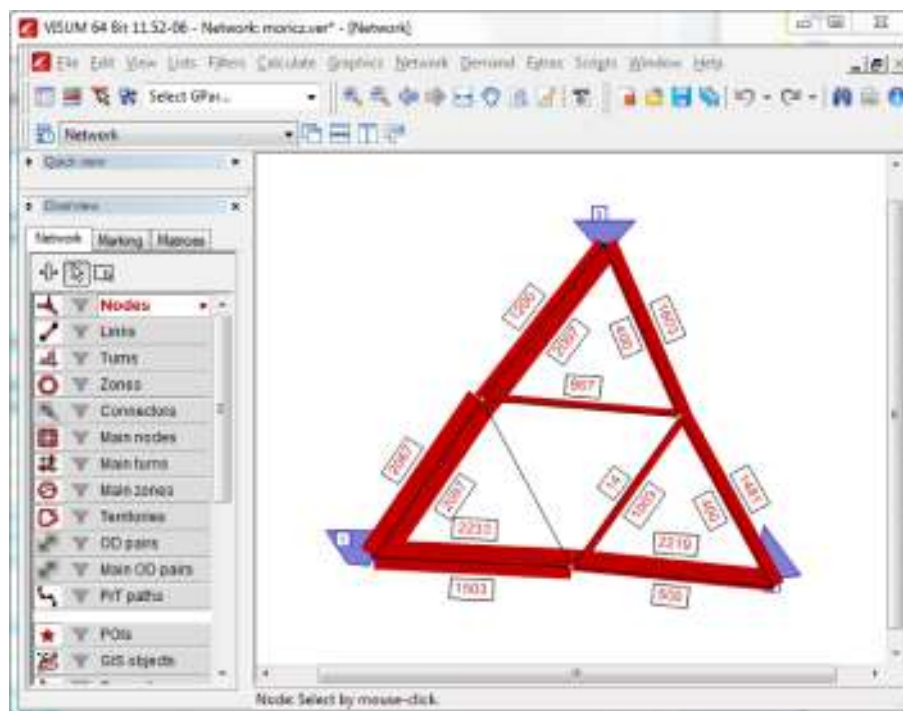
A VISUM szintén a német PTV Ag. által fejlesztett programcsomag, mely makroszkopikus, multimodális közlekedésszimulációs és térinformatikai feladatok megoldását teszi lehetővé nagy méretű városi vagy helyközi közút hálózaton.

A program egyszerű integrációt biztosít a mikroszkopikus VISSIM forgalomszimulációs eszközzel a nagyobb hálózat kisebb részleteinek a részletes modellezéséhez. Ezen felül egyszerű, szkriptelhető felhasználói felületet biztosít a mérések gyors és hatékony elvégzéséhez a minden részletre kiterjedő programozási felület mellett.

A VISUM szoftver kezelői felületét a 7.2 ábrán lehet megtekinteni, amint a teszhálózaton a ráterhelés során kiszámolt forgalomnagyságokat vörös színnel, arányosan ábrázolja az útszakaszok mentén.

7.2. Teszhálózat



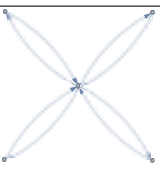
Az elkészített algoritmusokat először kis gráfokon próbáltam ki, amiket a 7.3 ábra mutat be. Az eredmények összefoglalása a 7.1 táblázatban látható. A $f_{o,d}^k$ jelölésű oszlop a folyam változók által kifizített tér dimenzióját mutatja, ez az egész tér, amiben a modell értelmezve van. A $D_{o,d}$ oszlop a célforgalmi mátrix elemei által



7.2. ábra. A VISUM makroszkopikus forgalom szimulációs szoftver kezelőfelülete

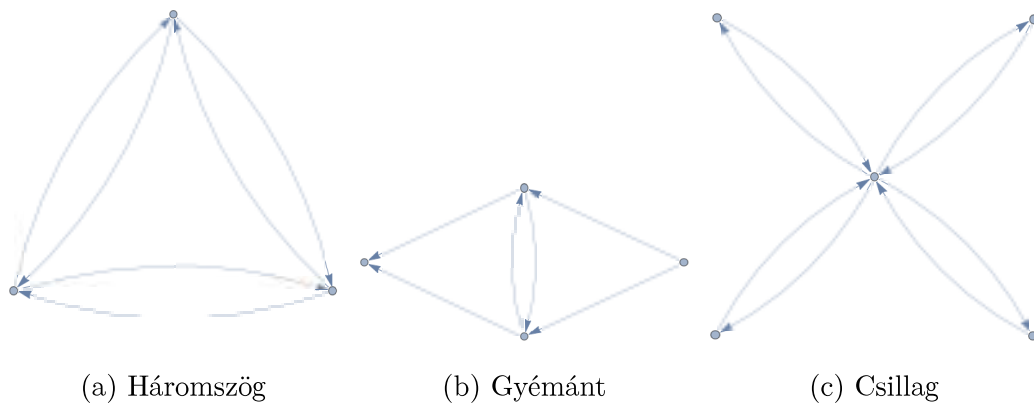
kifésztett alter dimenziója a folyamatok teljes terében. A következő, X_e jelű oszlop az éleken egyesével végzett méréseknek megfelelő alter dimenziója valamint az $D_{o,d} + X_e$ oszlop a mérések és a célforgalmi mátrix együttesen kifésztett alterének dimenzióját mutatja be a folyamatok terében.

Az utolsó deg_f oszlop a közösen kifésztett alter dimenziójának és a mérések alterének dimenziójának különbsége, kifejezi a célforgalmi mátrix szabadságfokát a mérésektől, annak függetlenségére utal.

gráf	$f_{o,d}^k$	$D_{o,d}$	X_e	$D_{o,d} + X_e$	deg_f
	12	6	6	10	4
	4	1	4	4	0
	12	12	7	12	5

7.1. táblázat. Kisebb teszhálózatok tulajdonságai

A három példa három különböző, kanonikus esetet mutat be.



7.3. ábra. Kisebb tesztesetek

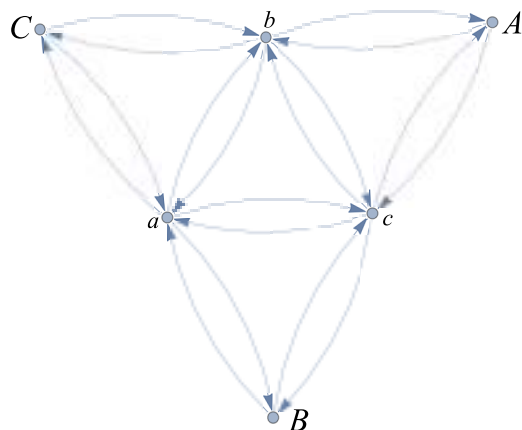
A háromszög nevű, a 7.3a ábrán látható tesztgráfban minden csúcsból minden csúcsba két különböző útvonalon lehet eljutni, ezáltal a folyamatok tere 12 dimenziós, mivel a célforgalmi mátrix diagonálistól különböző elemeinek száma 6. Ez a hat elem mind független egymástól, ezt mutatja a következő oszlop is, valamint a hat él forgalma is független, tehát az X_e altér is hat dimenziós. A vizsgált, együttesen kifeszített altér 10 dimenziós, így a célforgalmi mátrix négy paraméterben független a mérésektől, arra csak részleges becslés adható.

A 7.3b ábrán látható, gyémánt fantázianevű tesztgráfnak a célforgalmi mátrixa 1 elemű, mivel az irányított élek miatt abban csak jobbról balra lehet haladni. A lehetséges útvonalak száma négy, ezt a mérések tere teljesen ki is feszíti, a célforgalmi mátrix tehát pontosan megállapítható a mérések alapján, annak szabadságfoka nincs.

A csillag nevű gráfnak, amit a 7.3c ábra mutat, a négy periférikus csúcsa közötti célforgalmi mátrixát határoztam meg, ami a lehetséges hurokmentes utak egyértelműsége miatt kifeszíti a teljes 12 dimenziós teret. Az elvégezhető mérések ebből csupán egy 7 dimenziós alteret határoznak meg, így a célforgalmi mátrix szabadságfoka 5 marad.

Komolyabb vizsgálatnak a kettős háromszög nevű, a 7.4 ábrán látható kitalált közúthálózatot reprezentáló gráfot vettem alá. Ennek a külső csúcsai közötti célforgalmi mátrix becslhetőségét vizsgáltam az éleken elvégzett forgalmi mérések szerint.

A gráfnak 6 csúcsa és 18 éle van, a célforgalmi mátrix a kis háromszög gráf célforgalmi mátrixával egyezik meg, ennek is 6 meghatározandó eleme van, amiket a 7.2 mátrix ábrázol.



7.4. ábra. Kettős háromszög

	A	B	C
A	—	$D_{A,B}$	$D_{A,C}$
B	$D_{B,A}$	—	$D_{B,C}$
C	$D_{C,A}$	$D_{C,B}$	—

7.2. táblázat. A kettős háromszög célforgalmi mátrixa

7.3. Gráfalgoritmusok

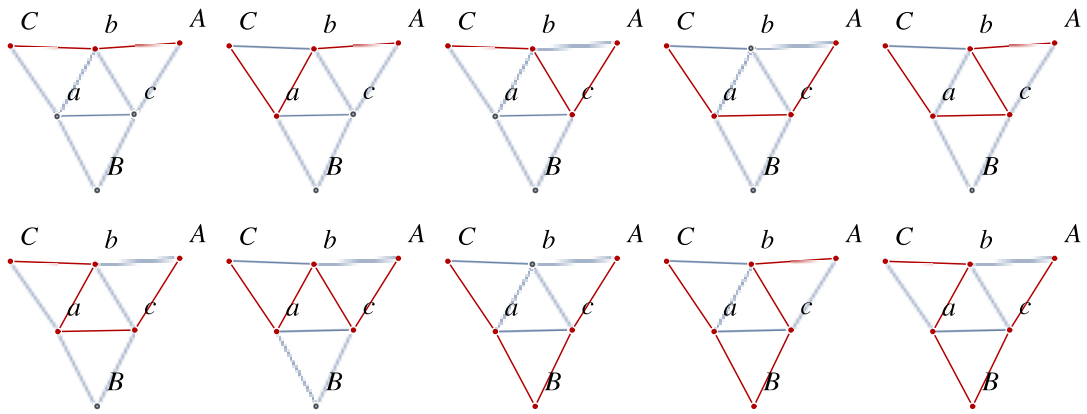
A 7.4 ábrán látható kettős háromszög vizsgálatához szükséges volt megállapítani a lehetséges útvonalak listáját a vizsgált, szélső csúcsok között. Ehhez a 6.3.1 szakaszban bemutatott Yen-féle k -legrövidebb utat kereső algoritmust használtam megfelelően nagy k értékkel.

A talált útvonalakat az A és C csúcsok között a 7.5 ábra mutatja, mint látható egy csúcspárhoz 10 különféle útvonal létezhet, így a szimmetrikus szerkezet miatt a folyamatok tere 60 dimenziós.

Mivel nem feltételeztem célforgalmat a belső csúcsok között, ott be- vagy kilépő forgalom sem fordulhat elő a modell szerint. Ezáltal a 18 él által kifeszített tér lineárisan összefüggő és csak 15 dimenziós alteret alkot, amint az a 7.3 táblázatból kiolvasható.

gráf	$f_{o,d}^k$	$D_{o,d}$	X_e	$D_{o,d} + X_e$	deg_f
	60	6	15	19	4

7.3. táblázat. Kisebb teszhálózatok tulajdonságai



7.5. ábra. Utak a háromszögön

A 7.3 táblázat tartalmazza a teszthálózatra elvégzett számíthatósági vizsgálat eredményét. A célforgalmi mátrix altere 4 dimenzióban független rendelkezésre álló mérések terétől, annak értékére csak eléggé pontatlan becslés adható.

7.4. Forgalmuszimuláció

A teszthálózat forgalmuszimulációs szoftverekben való előállításával lehetőség nyílt tesztelni a célforgalmi mátrix kiszámolhatóságát, becslést adni rá és az eredményeket összehasonlítani.

A tesztelés során a teszthálózat célforgalmi mátrixának elemeit egyesével állítottam négy lépésben a 7.4 táblázatnak megfelelő értékekkel. Az útszakaszok azonos hosszúak és $1800 \frac{\text{jármű}}{\text{óra}}$ kapacitásúra lettek állítva.

Lépés	Forgalomnagyság $\frac{\text{jármű}}{\text{óra}}$
1	100
2	1100
3	2100
4	3100

7.4. táblázat. A forgalmuszimuláció során használt forgalomnagyság skála

A célforgalmi mátrix hat értékének variálása hat dimenziós paraméterteret adott. A mátrix beállítása után elvégzett ráterhelés után a kapott adatokat rögzítettem és a 7.6 ábrán látható a tér két dimenziós szelete, ahol a célforgalmi mátrix elemeinek $D_{A,B}$, $D_{B,A}$, $D_{C,A}$ és $D_{C,B}$ rögzített értékei mellett $D_{A,C}$ és $D_{B,C}$ értékek módosításával kapott felületi vannak ábrázolva.

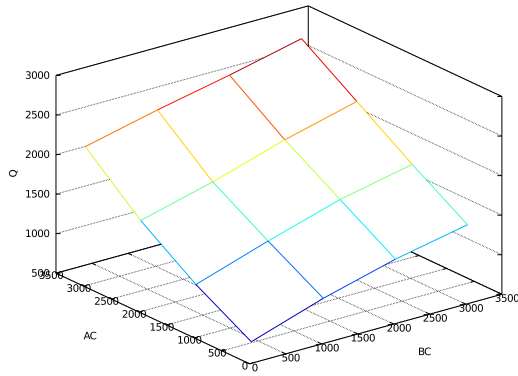
	A	B	C
A	—	2100	$D_{A,C}$
B	2100	—	$D_{B,C}$
C	1100	1100	—

7.5. táblázat. A célforgalmi mátrix rögzített és változó értékei a 7.6 ábrának megfelelően

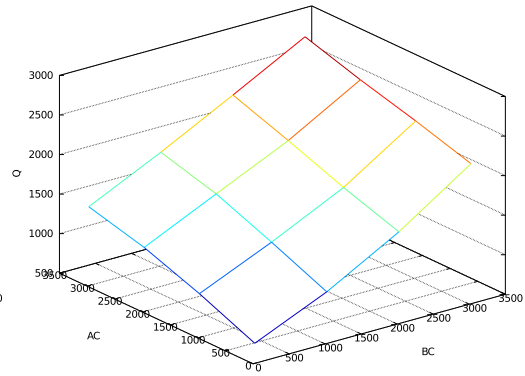
A 7.6 ábrán a két vízszintes tengelyen $D_{A,C}$ és $D_{B,C}$ értékei változnak, a felület pedig az adott mátrixelem forgalomnagyságának becsült értékét ábrázolja.

A várt eredmény a becsült értékekre a 7.6a és 7.6b ábrákra az egyik tengely mentén lineárisan növekvő, a másik tengely mentén konstans felület lenne. A többi értékre pedig konstans, vízszintes felületek az elvárt eredmények.

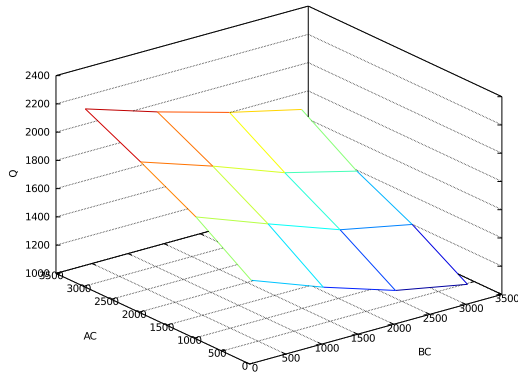
A kapott eredmény ettől némiképp eltér, de jellegében felfedezhető az elvárt felület a becsült értékekben. A szignifikáns hiba a becslő eljárás kidolgozatlanságára vezethető vissza, mely további vizsgálatot igényel.



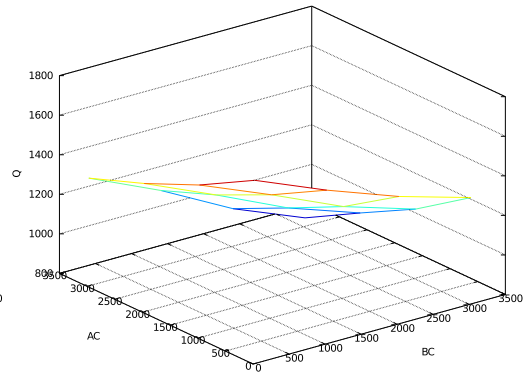
(a) $A \rightarrow C$



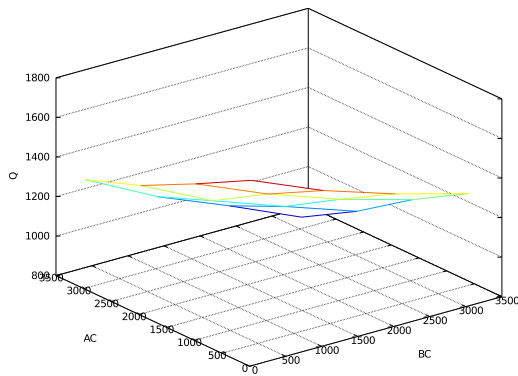
(b) $B \rightarrow C$



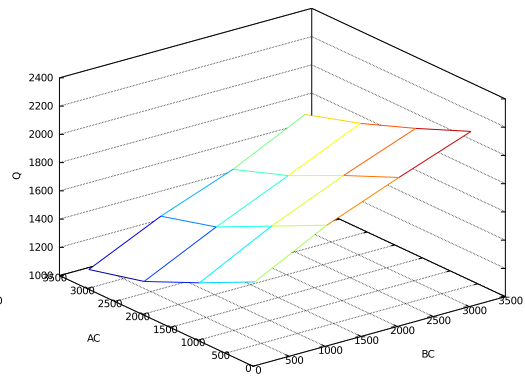
(c) $A \rightarrow B$



(d) $C \rightarrow B$



(e) $C \rightarrow A$



(f) $B \rightarrow A$

7.6. ábra. A célformalmi mátrix becsült értékei

8. fejezet

Összegzés és továbbfejlesztési lehetőségek

Megvizsgáltam a közlekedési ráterhelés feladatának matematikai modelljeit, valamint a célforgalmi mátrix számíthatóságának és becslésének lehetőségeit.

A kifejlesztett algoritmusok és azok számítógépes implementációja hatékonyan hozzájárulhatnak a közlekedési mérések és a forgalomtechnikai paraméterek becslhetőségének általános vizsgálatához, amennyiben a meglévő mérőeszközök hatékony kiaknázása vagy adott paraméterek becslhetőségéhez szükséges infrastruktúrafejlesztések elemzése a cél.

Az alkalmazott módszerek újszerűek az intelligens közlekedési rendszerek területén, a továbbfejlesztést tervbe vesszük mind elméleti, mind gyakorlati oldalról.

Különböző mérési technológiák alkalmazása esetén felmerül azok egymáshoz képesti megbízhatóságának, pontosságának a kérdése, mely további vizsgálatok tárgyát képezheti. Jelen dolgozat a vizsgált forgalmi paraméterek kiszámíthatóságára helyezi a hangsúlyt, nem az esetlegesen összefüggő, túlhatározott mérések korrekt és hatékony alkalmazására.

További kutatási téma lehet a választott k paraméter meghatározásának módja az útvonal alternatívák keresésekor, hogy a vizsgált rendszer hűen tükrözze a valós forgalmakat, ugyanakkor a számítási igénye se haladja meg a gyakorlati alkalmazhatóság praktikus korlátait.

Gyakorlati jelentőséggel bírna egy, a jelen dokumentumon alapuló, szakmabeliek számára elérhető és alkalmazható kidolgozott módszertan és szoftverimplementáció, amivel a gyakorlatban is könnyen vizsgálhatóvá válna a forgalomtechnikai paraméterek számolhatósága, az alkalmazott mérési módszerek határinak vizsgálata és az esetlegesen szükséges új mérési berendezések alkalmazásának módja.

A. Függelék

Yen féle k -legrövidebb út algoritmus forráskódja

```
1 YenKSP[origGraph_, source_, sink_, K_] :=
2   Module[{A = {FindShortestPath[origGraph, source, sink]}, B = {},
3     spurNode, rootPath, totalPath, spurPath, graph = origGraph},
4     For[k = 1, k < K, k++,
5       For[i = 1, i < Length[Last[A]], i++,
6         spurNode = Last[A][[i]];
7         rootPath = Last[A][[1 ;; i]];
8         graph = origGraph;
9         Do[
10          If[Length[rootPath] < Length[path]
11            ^ rootPath == path[[1 ;; i]
12            ^ EdgeQ[graph, path[[i] → path[[i + 1]]],
13            graph = EdgeDelete[graph,
14              path[[i] → path[[i + 1]]]
15            , True]
16          , {path, A}];
17          graph = VertexDelete[graph, Most@rootPath];
18          spurPath = FindShortestPath[graph, spurNode, sink];
19          If[spurPath == {} , True,
20            totalPath = Join[Most@rootPath, spurPath];
21            B = Append[B, totalPath]];
22          B = DeleteDuplicates[B];
23          B = SortBy[B,
24            Plus @@ MapThread[
25              PropertyValue[{graph, #1 → #2},
26                EdgeWeight] &, {Most@#, Rest@#}] &];
27          If[B == {}, Break[], True];
28          A = Append[A, First@B];
29          B = Rest[B];
30        A];
```

Listing A.1. Yen-féle k -legrövidebb út Mathematica-ban implementálva

Ábrák jegyzéke

1.1.	Jacques Ignace Parrocel festménye a zentai ütközetről	7
	http://hu.wikipedia.org/wiki/Fájl:Battle_of_Zenta.png	
2.1.	A Google útvonaltervezője által javasolt alternatív útvonalak	12
	http://maps.google.com	
3.1.	Budapest és környéke a 2008-ban a KTI által végzett országos célfor- galmi adatfelvétel térképén	14
	Közlekedés Tudományi Intézet, Országos Célforgalmi Adatfelvétel 2008	
3.2.	Braess paradoxon szemléltető példa	16
	http://en.wikipedia.org/wiki/Braess's_paradox	
3.3.	A fundamentális diagram teremt összefüggést a forgalomnagyság és a forgalomsűrűség között	17
3.4.	Hurokdetektor elhelyezése a közúton	18
	U.S. Department of Transportation, Traffic Detector Handbook	
3.5.	Egy Route Hawk típusú forgalom figyelő kamera	19
	Ludvig Ádám, Edinburgh közlekedési rendszerei, Közúti informatika 2012	
3.6.	Úszó autós mérés rögzített pontjai Budapest belvárosában	20
3.7.	Különböző típusú cellahatár átlépések a GSM hálózatban	20
5.1.	Bogdánfy utca és az Irinyi József utca kereszteződése	24
	http://maps.google.com	
5.2.	Az 5.1 ábrán feltüntetett csomópontból kialakított gráfrészlet	25
5.3.	A közút hálózat régióinak kiegészítése virtuális forrás és nyelő csú- csokkal	26
5.4.	A BPR görbe illusztrációja $a = 1$, $c = 1$, valamint többféle b paramé- terrel	28
	VISUM 11.5 – Basics[27]	
5.5.	Az INRETS görbe illusztrációja $c = 1$ és többféle a paraméterrel	28
	VISUM 11.5 – Basics[27]	
5.6.	A célforgalmi mátrix elemei és a mérések által kifizített alterek kap- csolatának illusztrációja	30

7.1. A VISSIM mikroszkopikus forgalom szimulációs szoftver kezelőfelülete	42
7.2. A VISUM makroszkopikus forgalom szimulációs szoftver kezelőfelülete	43
7.3. Kisebb tesztesetek	44
7.4. Kettős háromszög	45
7.5. Utak a háromszögön	46
7.6. A célforgalmi mátrix becsült értékei	48

Táblázatok jegyzéke

7.1. Kisebb teszhálózatok tulajdonságai	43
7.2. A kettős háromszög célforgalmi mátrixa	45
7.3. Kisebb teszhálózatok tulajdonságai	45
7.4. A forgalomszimuláció során használt forgalomnagyság skála	46
7.5. A célforgalmi mátrix rögzített és változó értékei a 7.6 ábrának megfelelően	47

Irodalomjegyzék

- [1] Torgil Abrahamsson. Estimation of origin-destination matrices using traffic counts – a literature survey. Technical report, International Institute for Applied Systems Analysis, 1998.
- [2] Husain Aljazzar and Stefan Leue. K*: Heuristics-guided, on-the-fly k shortest paths search. *Sixth Workshop on Model Checking and Artificial Intelligence*, 2010.
- [3] Stephen Boyd and Lieven Vandenbergh. *Convex Optimization*. Cambridge University Press, 2004.
- [4] Dietrich Braess. Über ein paradoxon aus der verkehrsplanung. In *Unternehmensforschung 12*, 1969.
- [5] Wettle Ferenc. *Lineáris algebra*. Typotex, 2011.
- [6] Christian Gawron. *Simulation-Based Traffic Assignment*. PhD thesis, Universität zu Köln, 1998.
- [7] Swarco Traffic Systems GmbH. BLIDS flyer. http://www.swarco.com/en/content/download/8058/101437/file/DRIVE-ON-1-2011_1.pdf, Sep. 2011.
- [8] Bruce D. Greenshields. A study of traffic capacity. In *14. Annual Meeting Highway Research Board*, 1935.
- [9] Péter Hajnal. *Gráfelmélet*. Polygon, 1997.
- [10] Balázs Horváth, Csaba Koren, István Prileszky, and Zsuzsanna Tóth-Szabó. *Közlekedéstervezés*. Széchenyi István Egyetem, 2007.
- [11] Gábor Ivanyos, Réka Szabó, and Lajos Rónyai. *Algoritmusok*. Typotex, 1998.
- [12] Martin Jaggi. Revisiting frank-wolfe: Projection-free sparse convex optimization. *Journal of Machine Learning Research: Workshop and Conference Proceedings*, 28(1):427–435, 2013.

- [13] Waclaw P. Jastrzebski. Volume delay functions. In *15th International EMME/2 Users' Group Conference*, 2000.
- [14] Edwin T. Jaynes. Information theory and statistical mechanics. *The Physical Review*, 106(4):620–630, 1957.
- [15] Axel Küpper. *Location-based Services*. John Wiley & Sons, 2005.
- [16] Lev D. Landau and Evgenij M. Lifšic. *Mechanika*. Tankönyvkiadó, 1984.
- [17] Lawrence M. Leemis and Stephen K. Park. *Discrete-event simulation: A first course*. Pearson Prentice Hall, 2006.
- [18] Ádám Ludvig, Tamás Tettamanti, and István Varga. Travel time estimation in urban road traffic networks based on radio signaling data. In *MITIP 2012*, 2012.
- [19] Tamás Luspay, Tamás Tettamanti, and István Varga. *Forgalomirányítás*. Typotex, 2011.
- [20] MathWorks. Import and export geographic data. <http://www.mathworks.com/products/mapping/description2.html>, Máj. 2013.
- [21] Bureau of Public Roads. *Traffic Assignment Manual*. U.S. Dept. of Commerce, Urban Planning Division, Washington D.C., 1964.
- [22] Alexander Schrijver. *Combinatorial Optimization: Polyhedra and Efficiency*. Springer, 2003.
- [23] Eszter Sikolya. Feltételes szélsőérték. http://www.cs.elte.hu/~seszter/oktatas/2008_09_2/BSc_mattanar_ea/Lagrange_multiplikator.pdf, 2008.
- [24] K.A. Surana, S. Allu, P.W. Tenpas, and J.N. Reddy. K-version of finite element method in gas dynamics: higher-order global differentiability numerical solutions. In *International Journal for Numerical Methods in Engineering* 69, 2007.
- [25] Henk Taale. *Integrated Anticipatory Control of Road Networks*. PhD thesis, Technische Universiteit Delft, 2008.
- [26] Tamás Tettamanti, Hunor Demeter, and István Varga. Route choice estimation based on cellular signaling data. *Acta Polytechnica Hungarica*, 9(4):207–220, 2012.

- [27] PTV Vision. *VISUM 11.5 - Basics*. PTV AG, 2010.
- [28] PTV Vision. *VISSIM 5.40-01 - User Manual*. PTV AG, 2011.
- [29] John Glen Wardrop. Some theoretical aspects of road traffic research. In *ICE Proceedings: Engineering Divisions 1*, 1952.
- [30] Wolfram. Combinatorica'. <http://reference.wolfram.com/mathematica/Compatibility/tutorial/Combinatorica.html>, May. 2013.
- [31] Jin Y. Yen. Finding the k -shortest loopless paths in a network. *Management Science*, 17(11):712–716, 1971.
- [32] Szili László és Tóth János. *Matematika és Mathematica*. ELTE Eötvös Kiadó, 1996.