

Szoftveres folyamatanalízis eredményei a nagyméretű közúti közlekedési hálózatok tervezésénél

Dr. Péter Tamás*, Stróbl András*, Fazekas Sándor*

*Budapesti Műszaki és Gazdaságtudományi Egyetem Közlekedéautomatikai Tanszék (tel.: +36-1-4631013; e-mail: peter.tamas@mail.bme.hu)

Absztrakt: Az előadás anyaga nagyméretű közúti közlekedési rendszerek modellezésére kifejlesztett speciális, szoftver, amely figyelembe veszi a hálózat hipergráf struktúráját. A működését az [1,2,3,4,5,6] által bemutatott nemlineáris hálózati modell szabja meg. A szoftver fejlesztése közel két éve indult el a Budapesti Műszaki és Gazdaságtudományi Egyetemen. Ez idő alatt számos verziót adtunk ki; a kezdeti hibákat orvosolva, majd a folyamatosan növekvő felhasználói igényekhez alkalmazkodva újabb és újabb funkciókkal bővült a program. A szoftver három főegységből áll. (A hasonló témájú programcsoportok esetében az úthálózat tervezése, a szimulációt végző program, valamint az analízisre alkalmas felületek külön-külön szoftverként kerülnek kiadásra.) Mi ezt a három részt egyesítettük.

1. BEVEZETÉS

Budapesten a dinamikus jövedelemnövekedés hatására az egy főre jutó gépkocsiszám 22 százalékkal nőtt az elmúlt bő tíz esztendőben. Mára a fővárosi gépkocsi-közlekedés legfontosabb költségét, a járműberuházáson túl, egyre inkább a forgalmi torlódásokhoz köthető kiadások jelentik. Az ezredforduló után a belvárosi kerületekben a gépjárművek átlagos sebessége a csúcsidőben 15 kilométer/óra alá csökkent, ami alacsonyabb, mint egy kerékpáros sebessége. A forgalom lelassulása elsősorban a munka-, illetve szabadidő kiesése miatt költséges, mivel a gépkocsikban munkavállalók ezrei várakoznak tétlenül. Emellett az alacsonyabb sebesség hatására megemelkedik az üzemanyag-fogyasztás, és a környezetszennyezés mértéke is növekszik.

A motorkerékpár egyéni közlekedésre használható az autóhoz hasonlóan, a forgalom számára azonban kisebb megterhelést jelent, és a torlódásokon is könnyebben átjut. A legtöbb kelet-európai városban a motorkerékpár egyelőre kevésbé népszerű, míg Rotterdamban és Lyonban ezer lakosra 177, illetve 97 motorkerékpár jut, addig Budapesten, illetve Varsóban mindössze 9, illetve 37.

Budapesten a bejegyzett gépkocsik száma 7 százalékkal nőtt az 1990-es évek közepe óta, és 2005-re megközelítette a 600 ezret. A gépkocsiszám annak ellenére emelkedett, hogy a lakosság 12 százalékkal csökkent, részben demográfiai okok, részben az agglomerációba történő kiköltözési hullám miatt. Ennek következtében a lakosságátlagos gépkocsi állomány dinamikusan bővült a fővárosban, és a vizsgált időszak végén ezer lakosra már 357 gépkocsi jutott.

A BKV adatai szintén a forgalom lassulását tükrözik, az autóbuszok átlagos sebessége az 1994. évi 17,2 kilométer/órától 15,2-re csökkent 2005-re. Különösen nagymértékű volt az autóbuszok sebességcsökkenése 2005-ben, egyetlen év alatt csaknem 1 kilométer/órás (5 százalékos) volt a lassulás.

A sebességcsökkenés mértéke a belvárosi kerületekben volt igazán kiugró. A pesti Nagykörúton belüli területeken a sebesség 5 év alatt 40 százalékkal csökkent, és 2003-ban már csak 13 kilométer/órával lehetett haladni.

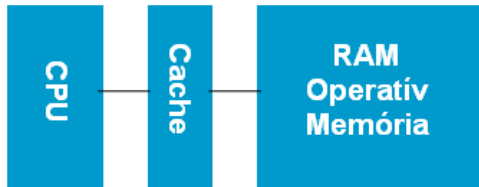
Összehasonlításként: Londonban és Stockholmban ennél magasabb, 14 kilométer/óra, illetve 20 kilométer/óra volt a járművek sebessége a csúcsidőben, amikor a belvárosi útdíj bevezetéséről döntés született (Transport for London [2006], Facts and Results [2006]). A belvárosi kerületekben tapasztaltnál kisebb, de szintén kiugró, 20-30 százalékos volt a külső kerületekben a forgomlassulás.

A forgalom lassulása azonban nem csak a fővárost érinti. Valamennyi jelentős forgalommal bíró településre igaz ez, arányosan. Erre az általános problémára célravezető volna egy általános megoldást találni. Ilyen lehet egy nagyméretű közúti közlekedési hálózatok folyamatanalízisét végző szoftver, tervezésben és közlekedés-irányításban történő használata.

2. A FEJLESZTÉS

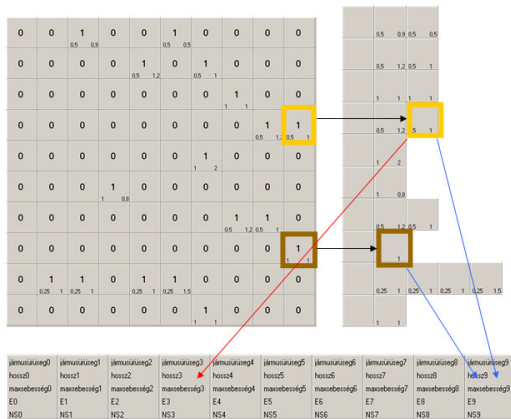
A fejlesztés során végig fontos szempont volt, hogy a számításokat végző kód minél közelebb legyen az optimális programozási megoldáshoz. Az egy évvel ezelőtt bemutatott verzióban az egyik

legjelentősebb sebességjavulást a mátrixtérképeknek köszönhetően: ezek segítségével a kisebb hálózatokon valóban sikerült az elméletileg is elvárt lineáris futásidő-növekedést elérni, azonban a futási időre vonatkozó mérési eredmények nagyobb hálózat esetén jelentős lassulást mutattak. Ennek oka a korábbi verzió pazarló memóriakezelésében keresendő: a mátrixok és a már említett mátrixtérképek hatalmas mérete rossz cache-kihasználást eredményezett, egyre „ritkább” volt a memóriában a valós kapcsolatot leíró adat.



1. ábra: A cache elhelyezkedése – a gyorsítótár kihasználása leromlott a nagyobb hálózatokon.

Az újraírt szimulációs motor (harmadik verzió) ezt a gyengeséget orvosolja: sokkal jobban gazdálkodik a memóriával, mint a korábbiak; lineárisan nő a tárigény a hálózat méretével. Gyakorlatilag a szimuláció úgy végez számításokat - a differenciálegyenletekkel és mátrixokkal leírt modellen -, hogy a program futása során nem is jönnek létre mátrixok. Ez a megoldás szintén, a korábbi verzióban megfigyelt tulajdonságra épül: közlekedési hálózatunk lokálisan strukturált rendszert alkotnak, azaz egy komponens (szakasz) csak korlátozott számú más komponenssel van kapcsolatban közvetlenül, függetlenül a komponensek teljes számától. Ezekre a lokális struktúrákra általában is jellemző, hogy inkább lineáris, mint négyzetes vagy exponenciális a komplexitás növekedése.



2. ábra – A korábbi mátrixos elrendezés és az új listás adattárolás kapcsolata

Az új megvalósításhoz a 2.ábra bal oldalán látható mátrixos adattárolás helyett a jobb oldalt és lent látható listás szerkezeteket vezettük be (fent baloldalt a nagy egyessel jelölt elemek jelölik a tényleges kapcsolatokat, a fent jobb oldalon látható listás szerkezet már csak ezeket tartalmazza.) A szerkezetet mutatókkal támogattuk meg, ezeket a pointereket pedig arra használtuk, hogy a mátrixon végzett műveleteket matematikailag értelmezhesük az új adatszerkezeten. Így minden lépést megvalósíthattunk, amit a matematikai modell megvalósított, emellett a lineáris tárigényt és a nagyságrendileg lineáris futásidő-növekedést is elértük!

A számítási teljesítményben elért javulás nem öncélú. A felszabadult számítási kapacitásoknak köszönhetően több bétával (-kapcsolat akadályozót/rásegítést ír le-) modellezett objektumot helyezhetünk el, ráadásul az új verzióban ezekhez már dinamikusan változó bétaértékek is rendelhetők. Ezeket megfelelő forgalmi környezetvizsgálattal beállítva, lehetővé válik, hogy a valóságot még inkább megközelítő szimulációs eredményeket adhasson programunk a felhasználó számára.

3. FUNKCIONÁLIS BŐVÍTÉSEK

A következőkben áttekintjük, hogy funkcionálisan mivel bővült alkalmazásunk.

Közlekedési lámpa üzemmódok: Immár különböző üzemmódokba állíthatjuk a közlekedési lámpákat. Normál mód alatt a „hagyományos” működést értjük. Itt a felhasználó különböző, napszakfüggő időintervallumokat állíthat be, ezeken belül ciklikusan változik a lámpák állapota.

Egy másik lehetőséget kínál a perem üzemmódban működő közlekedési lámpa. Ez az egész hálózat sűrűsége alapján dönt állapotáról. Jelenleg egy hiszterézises megoldást alkalmazunk: globálisan megadható egy intervallum. Amennyiben a belső szakaszok összsűrűsége meghaladja az intervallum szuprémumát, akkor pirosra vált az összes peremlámpa. Amennyiben az infimum alá süllyed az összsűrűség, ismét zöldre váltanak a peremrészeken elhelyezett közlekedési lámpák. Megjegyzendő azonban, hogy ezek alkalmazása gondos körültekintést igényel: valószínűleg az input szakaszok sűrűségének mérési eredményei „peremlámpa-mentes” esetben állnak majd rendelkezésünkre. Amennyiben a peremlámpa piros, a szimuláció nem veszi figyelembe, hogy az inputszakaszon a valóságban „beragadnak” a járművek, nem tudnak továbbhaladni. Szimulációinkban akkor is csökkenhet az input szakaszok sűrűsége, amikor az inputszakaszra helyezett peremlámpa piros jelzést mutat (egészen

pontosan akkor, amikor az inputfüggvény deriváltja 0-nál kisebb).

Adaptív módban beállítható egy t mintavételezési idő, valamint az „1. csoport” és a „2. csoport” címkékkel láthatjuk el a kapcsolatokat az adott kereszteződésben. Minden t -ik másodpercben megvizsgálja a program a lámpás kereszteződés pillanatnyi járműátadásait, és az a csoport kap zöld jelzést, amelyiken a jelenlegi sűrűségviszonyok alapján nagyobb járműátadás valószínűsíthető. Azonban egymás után csak n (beállítható) esetben kaphat zöld jelzést ugyanaz a kapcsolat-csoport, az $n+1$ -ik mintavételezéskor mindenképpen az a csoport kap zöld jelzést, amelyik eddig n esetben nem kapott.

A járműátadás becslését illetően meg kell említeni a következőt: ennek számítása a hagyományos módon történik, azzal a megjegyzéssel, hogy a statikus és dinamikus bétákkal nem számolunk. Ennek két oka van: egyrészt programozástechnikailag a lámpák értékét reprezentáló „kk” jeleket is „dinamikus bétaként” kezeli a program, azaz a bétákkal történő beszorzás esetén az, aki az előző mintavételezéskor pirosat kapott, most is pirosat kapna, mivel a becsült járműátadása nulla a piros jelzés miatt. Másrészt a szomszédos szakaszok sűrűségein kívül az adaptív lámpa semmilyen egyéb „ismerettel sem rendelkezik” a külvilágról, balesetekről, kerékpárút terheléséről, gyalogosforgalomról, stb., tehát nem is kalkulálhat a béta értékekkel.

Változó bétaértékek a modellben: jelentős előrelépés történt a valódi elsőbbségviszonyok szimulálásában. Tekintsük az alábbi ábrát:



3. ábra – Dinamikus béták nélkül valószínűleg helyzetek alakulhattak ki a szimuláció közben

A fenti 3. ábra a modell korábbi működését demonstrálja. Egy ilyen közlekedési helyzetben a teherautó a valóságban nem kanyarodhat ki, hiszen a kétsávos (minden bizonyos fölérendelt) úton haladóknak elsőbbségük van a balra kanyarodó

teherautóval szemben; ha nagy a forgalom azon a sávon, amelyről a teherautó érkezik, akkor a kétsávos úton közlekedő járművek hatására a betorkolló kis útszakasz be is dugulhat.

Az új modell ezzel szemben már lehetőséget ad a fenti, valószínűleg helyzetek kivédésére. Az elsőbbségviszonyok szimulálása a következőképpen történik: az alárendelt utak járműátadásait egy adott kereszteződésben a béta szorzóval lefojtjuk, amennyiben a fölérendelt utak járműsűrűségei (x_i állapotjellemzőik) nagy értéket vesznek fel. Ennek a bétatényezőnek a kiszámolása a következő képlettel történik:

$$\beta = \prod (1 - x_i) \\ i \in A,$$

ahol „ x_i ” az adott sáv sűrűsége, „ A ” halmaz pedig azokat a sávindexeket tartalmazza, amelyek az adott kereszteződésben bármely fölérendelt szakasznak forrassávjaihoz tartoznak.

Ez a formula persze minimális mértékben egyszerűsít, hiszen ha a fölérendelt szakasz bedugult, akkor is történik némi járműátadás, az alárendelt szakaszból „befurakodnak”, a fölérendelt szakaszon pedig „beenged(het)ik” a járművek az alárendelt szakaszból érkezőket.

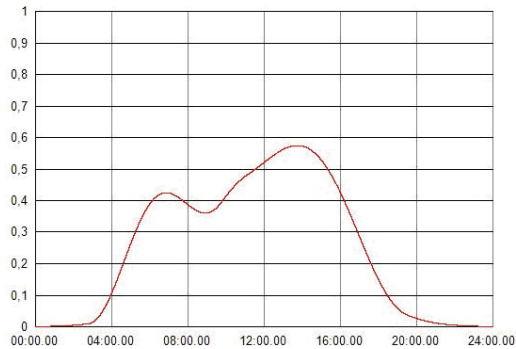
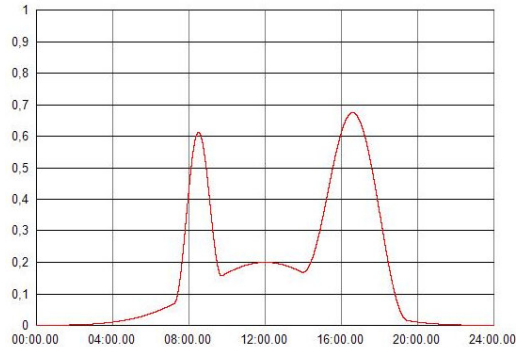
A dinamikus béták egy másik esetben is kiterjesztik, pontosabbá teszik a modellt: a korábbi verzió két egyirányú útként kezelte két párhuzamos szakaszt. A valóságban azonban ezek a szakaszok is kooperálhatnak (a lassabb járműveket könnyű kikerülni, nagyobb lesz az adott idő alatt tapasztalt járműátadás). Az új verzió szimulációjának elindításakor ezek a szakaszok automatikusan felderítésre kerülnek, és a két szakasz x_i, x_k állapotjellemzőitől függő bétával szorzódnak fel a járműátadások.

Paraméterezhető függvények: A modell adott városhoz történő igazításakor mindenképpen szükség lesz arra, hogy a kódban történő változtatás nélkül skálázzuk, a mérési eredményekhez közelítsük a bemenő adatokat. Ennek céljából kezdődött meg egy fejlesztés, melynek eredményeként egyedi függvényeket vehetünk fel, melyeket hozzárendelhetjük az egyes input vagy outputszakaszokhoz, parkolókhöz.

A függvények tárolása külön állományokban történik, ezekre hivatkozik a hálózatot tároló állomány. Ennek előnye, hogy az egyik hálózatban felhasznált függvény könnyen átvihető a másikba. Alapvetően két különböző típusú függvény adható meg. Az egyik egy tapasztalati úton felvett

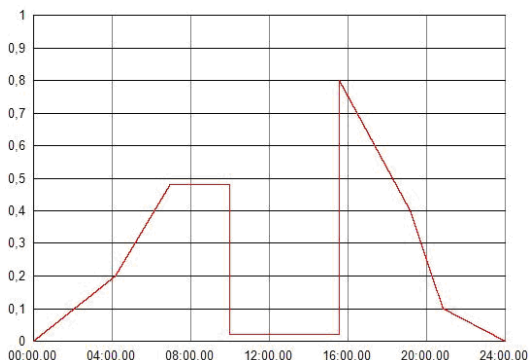
karakteristikájú grafikon, a második típus kvázi tetszőleges.

Az első típus azt veszi alapul, hogy egy átlagos úthálózaton háromszor figyelhető meg kiugró forgalomnövekedés a nap folyamán. A reggeli ill. délutáni csúcsidekben, valamint egy kisebb az ebédidő táján. Ezt a jellegzetességet figyelembe véve létrehoztunk egy összetett, folytonos sűrűség görbét, melyben az egyes paraméterek a felhasználó által változtathatók (időben és intenzitásban egyaránt).



4. ábra – Sűrűség görbék az egyes input vagy outputszakaszokhoz

A második típusban tetszőleges számú pontban megadható a függvény által felvett érték, és az, hogy a következő ilyen megadott pontig egységugrás vagy rámpa függvény jelleggel folytatódjon a grafikon.



5. ábra – Tetszőleges számú pontban megadható sűrűség görbék az egyes input vagy outputszakaszokhoz

Az elkövetkezendőkben szeretnénk egy splineszerkesztő funkcióval kiegészíteni az alkalmazás függvényszerkesztő-modulját. Elképzelésünk szerint a grafikonra kattintva lehet majd kontrolpontokat hozzáadni a függvényekhez, amelyek mentén egy valóságos sűrűségfüggvény görbéjét állítja elő.

4. A JÖVŐ, A PÁRHUZAMOSÍTÁS

Elmondható, a mega- vagy gigahertzekben mért processzor sebességnövekedésben tapasztalt lendület elfogyni látszik, ehelyett a gyártók több magos processzorokat építenek. Ezeket azonban a hagyományos módon algoritmizált szimuláció nem tudja kihasználni: jelenlegi szimulációs modellünk egy szálon fut, emellett egy másik szál felelős a felhasználói felület megjelenítéséért és figyeléséért. A jövőben a jobb számítási hatások elérése érdekében többszálúra szeretnénk módosítani a szimulációt, ehhez meg kell oldani a szálak szinkronizációját.

További terveink között szerepel, hogy növeljük a számítási kapacitást a szimulálandó hálózat több számítógép közötti szétosztásával. Gondoljunk csak bele: adott egy hivatal, néhány számítógéppel, internetre kapcsolódó belső hálózattal, ezeken feltehetőleg böngésznek, vagy szövegszerkesztőt használnak az alkalmazottak. Ez alig terheli a gépüket, így ezek a gépek szabad kapacitásként is felfoghatóak. Amennyiben a hálózatot megfelelően szét tudnánk osztani a hivatal gépei között, akkor párhuzamosan haladhatna a szimuláció, csakúgy mint a többmagos processzor esetén.

Fontos hangsúlyozni, hogy ez egy olyan elosztott közlekedési hálózatot reprezentáló gráf lenne, melynek összes, térkép szerinti éle által reprezentált szakasz sűrűségéről egyik gép sem rendelkezik információval, minden gép csak a számára kiosztott résztérkép, részhálózati sűrűségét számítja. Természetesen a számításokhoz olyan adatokra is szükség van, amelyek egy másik gép számára lettek kiosztva, itt gondoljunk elsősorban a részhálózatok peremén lévő azon szakaszokra, melyeknek a szomszédja már egy másik részhálózathoz tartozik, azaz az adott kapcsolat két külön gépen „helyezkedik el”. Ezeket az információkat a hálózaton keresztül kérdeznék le a „szomszédos” gépektől.

Emiatt törekedni kell arra, hogy a részgráfok kiosztása megfelelő legyen, azaz lehető legkevesebb hálózati kommunikációra legyen

szükség. Ebből következik, hogy a béták értékének számításánál is törekedni kell a „lokális” végrehajtásra, elképzelhető tehát, hogy a perem üzemmódban működő lámpák nehézséget okoznak: az összes szakasz sűrűségét ismernie kell egy adott időpillanatban annak a gépnek, aki a lámpa értékét beállítja, sok gép esetén megnőhet a hálózati kommunikáció mennyisége.

A hálózaton keresztül történő kommunikáció működéséhez meg kell oldani, hogy a rendszer ne álljon le egy-egy gép kilépésekor sem: ha lehetséges, akkor az addigi mérési eredményeket a mérést kezdeményező gépnek kell elküldeni, aki majd ismét kiosztja ezt a részhálózatot egy másik gépnek. A legrosszabb esetre felkészülve pedig visszalépési pontokat kell beiktatni: például minden modellidő szerinti negyedórán elküldik a résztvevők az eredményeiket a mérést kezdeményező gépnek.

A fenti lehetőség pedig még mindig nem feltétlenül a végleges: akár egy GRID rendszer is kialakítható az ismertett modellel: a program segítségével az interneten keresztül is használhatóvá válhatnak az otthoni erőforrások. Természetesen ez nagyon sok izgalmas problémát rejt magában: ki és belépő gépek kezelése, a részhálózatok adaptívan történő változtatása, stb.

5. ÖSSZEFOGLALÓ

Szoftverünk nagyméretű közúti közlekedési hálózatok átfogó modellezésére és analízisére alkalmazható. Jelenlegi állapotában már közel lineáris a hálózati elemek számától függő futásidő. A szoftver által reprodukált közlekedési hálózat a valóságnak megfelelően képes – a felmért hálózat paramétereinek ismeretében, azt – a nap 24 órájára vonatkoztatva – szimulálni, majd az eredményeket grafikus formában is rendelkezésre bocsátani. A szoftver három főegységből áll. A hasonló témájú programcsoportok esetében az úthálózat tervezése, a szimulációt végző program, valamint az analízisre alkalmas felületek külön-külön szoftverként kerülnek eladásra. Mi ezt a három részt egyesítettük. Rámutattunk arra, hogy a jövő - a jobb számítási hatások elérése érdekében - a többszálúra módosított szimuláció, ehhez azonban meg kell még oldani a szálak szinkronizációját.

IRODALOM

Dr. Péter T. Stróbl A. – Fazekas S.: Szoftveres folyamatanalízis, nagyméretű közúti közlekedési hálózatok optimalizálására, *A jövő járműve, 1-2. Budapest, 2008*

Dr. Péter T. – Stróbl A. – Fazekas S.: Hazai szoftverfejlesztés a nagyméretű közúti közlekedési hálózatok folyamatanalízisére, Budapest, 2007 Magyar Mérnökakadémia: Innováció és Fenntartható Felszíni Közlekedés <http://www.kitt.bmf.hu/mmaws/index.html>

Dr. Péter T., Nagyméretű közúti közlekedési hálózatok analízise MMA „Innováció és fenntartható felszíni közlekedés” - Konferencia, 2007. szeptember 4-5-6 Budapest, BMF <http://www.kitt.bmf.hu/mmaws/index.html>

Dr. Péter T., Nagyméretű nemlineáris közlekedési hálózatok modellezése *Közlekedéstudományi szemle, 9. 2007. Szept. LVII. Évf. pp. 322- 331.*

Dr. Péter T. - Dr. Bokor J.: Járműforgalmi rendszerek modellezése és irányításának kutatása. *A jövő járműve, Bp, 06, 1-2 pp 19-23.*

Dr. Péter T. – Dr. Bokor J.: Nagy méretű közúti közlekedési hálózatok nemlineáris modelljének kapcsolati hipermátrixa, *A jövő járműve, 1-2. Budapest, 2007*

Drew, D. R.: Traffic Flow Theory and Control, *New York, McGraw-Hill Book Company, 1968*

Maklári J.: Közforgalmú csomópontok teljesítőképességének vizsgálata. *Városi közlekedés 2001/4*

M. Papageorgiou: Concise Encyclopedia of Traffic and Transportation Systems. *Pergamon Press, 1991.*

Kachroo P. - Özbay K.: "Feedback Control Theory for Dynamic Traffic Assignment", *Springer, 1999.*

Kövesné dr. Gilicze É. - Dr. Debreczeni G.: Intelligens közúti közlekedési rendszerek és újtármű rendszerek matematikai modellezése és analízise, (2004). *IOM. Kutatási jelentés/*

Erhart Szilárd: A budapesti közlekedési dugók okai és következményei, *Közgazdasági Szemle, LIV. Évf. 2007. május*