Útvonalkereső és forgalom-ráterhelési eljárás közlekedési hálózatokban

A közlekedési modellszámítások fontos lépései, az útvonalkeresés és a forgalom-ráterhelés „alapeljárásai". Ezek alkalmasak a szoftverek működési elveinek vázlatos megértésére és adott esetben a gyakorlati megvalósítására. A gyári szoftvercsomagok ma már különböző stratégiáknak megfelelő eljárásokat tartalmaznak, de ezek általában a használók felé, mint „fekete doboz" működnek. A tanulmány a különböző eljárások típusainak vázlatos osztályozása mellett mind a szoftvereket alkalmazó „modell-futtatóknak", mind az eredményeket használó „társ-szaktervezőknek" egy útvonalkereső és a hozzá kapcsolódó ráterhelési eljárás példáján a technikai alapok fő vonásait írja le, és ugyanakkor az érdeklődőket a programozási folyamat szintjéig nyúló fogásokkal is megismerteti.

DOI 10.24228/KTSZ.2017.2.1

Dr. habil. Monigl János
FŐMTERV Zrt.
e-mail: monigl@fomterv.hu

1. BEVEZETÉS

A közlekedés személyek és javak valamely induló pontból – meghatározott indokból – valamely más célpontba való mozgását jelenti a közlekedési hálózatokon. Ezek a mozgások, vagyis egyes utak, az emberek, ill. a járművek vezetői által a kiindulási ponttól a célpontig választott, szakaszokon és csomópontokon áthaladó útvonalak mentén jönnek létre.

Az útvonalválasztás döntően a racionális emberi értékrendnek megfelelően, az út megtételéhez, leküzdéséhez szükséges ráfordítások (pl. idő, költség, stb.) minimálásával történik, aminek eredménye egy-egy kiinduló pont és célpont viszonylatában „legrövidebb útvonalnak" tekinthető.

Tekintettel arra, hogy az emberek nem képesek a hálózatot és a forgalmi körülményeket teljességgel áttekinteni, illetve az egyes hálózati elemeken (szakaszokon, csomópontokon) való áthaladás „ellenállását" pontosan felmérni, továbbá a vezetőket útközben lokális behatások is érhetik, ezért egy-egy viszonylatban többféle útvonal is kialakulhat, amely mozgások összességükben a hálózat és elemei forgalmi terheléseit eredményezik.

A közlekedéstervezés egyik alapvető feladata a közlekedési igények jövőbeli alakulása függvényében a hálózati kapcsolatok és a létesítmények megfelelő tervezése és kialakítása, amihez a várható hálózati terhelések ismerete elengedhetetlen. Ezeket ma már célszerűen közlekedési modellek alkalmazásával lehet meghatározni. A közlekedési igénymodellekhez (forgalomkeltési, szétosztási, megosztási modellekhez) csatlakozó forgalmi modellek – megfelelő „útvonalkereső eljárás" alkalmazásával – teszik lehetővé az utazási igények (áramlási mátrixelemek) ráterhelését a hálózatra. Ehhez a közlekedési hálózat az utazási igények keletkezési helyeihez (települések, körzetek) kapcsolódóan modellszerűen leképezésre kerül, aminek következtében a hálózat egyes elemei, a szakaszok és csomópontok összefüggő „gráfot" képeznek. A hálózati elemeket megfelelő geometriai és forgalmi jellemzőkkel írják le (pl. szakaszhossz, sávszám, sebesség, szabályozási mód, stb.). Ebben a vonatkozásban a közlekedési hálózaton belül különbség adódik a közúti és a közösségi közlekedési hálózat leképezésében és az alkalmazható eljárások sajátosságaiban.

2. AZ ÚTVONALKERESŐ ELJÁRÁSOK TÍPUSAI ÉS LÉNYEGI VONÁSAI

A hálózati számítások során – a közlekedési jelenségek térbelisége miatt – alapvető feladat tehát a közlekedési hálózat minden pontjából minden pontjába a (legrövidebb) útvonalak meghatározása, azaz az útvonalak ellenállása mértékének (ellenállásvektor (W)) meghatározása, valamint az útvonalak menetének, azaz a szakasz–csomópont sorozatok rögzítése (útvonalvektor (R)).

Ez többféleképpen is történhet, azaz több különböző típusú útvonalkereső eljárás létezik, amelyek legfontosabb különbségismérvei a következők:

– a **munkamódszert** tekintve léteznek ”mátrix-eljárások", amelyek valamennyi pont közötti útvonalakat egyszerre, szimultán módon, fokozatos javításokkal, hosszadalmasabban határozzák meg, valamint „vektoreljárások", amikor egyszerre csak egy „gyökérponthoz" tartozó útvonalfát és ágait határozzák meg,
– az **útvonalak számát** tekintve léteznek csak az 1. legrövidebb utat és több (2., 3. …k-dik legrövidebb) utat számító eljárások,
– a **keresés technikáját** tekintve léteznek „próbálgatásos" *(trial and error)* lassúbb, heurisztikus eljárások, valamint „célirányos"*(once through)* gyorsabb, direkt eljárások,
– az **ellenállások alapját** tekintve léteznek távolságalapú (pl. km, m), időalapú (pl. perc, mp), költségalapú (pl. Ft) eljárások, vagy ezen mértékek kombinációit használó eljárások,
– **az ellenállások meghatározottsága** szerint beszélhetünk „determinisztikus" eljárásokról, amelyek a hálózati elemek ellenállásainak fix értékeit veszik alapul, valamint „sztochasztikus" eljárásokról, amelyek az ellenállásértékek „tökéletlen ismeretéből" kiindulóan az értékek eloszlása lapján, több értékkészlet alapján számolnak, ami eleve több útvonalat eredményez.

A különböző típusú eljárásokat részletesebben tárgyaló művek közül kiemelhető Sheffi rendszerezést is adó kötete [1], továbbá az alapokat tekintve Ortuzar és Willumsen modellezési könyve [2]. A magyar nyelvű irodalomból a Széchenyi István Egyetem könyv formájában is megjelent jegyzete [3] adhat áttekintést a különböző típusú módszerekről.

Az útvonalkeresési és a ráterhelési eljárások számítógépes programszintű megoldása során lényeges, hogy gyors eljárások álljanak rendelkezésre, hisz ezek a lépések a közlekedési modellezés legidőigényesebb lépései, amelyek egy-egy feladat során, a hálózat méretétől és a változatok számától függően, jelentős számítási időt kívánhatnak.

A következőkben egy „rámenős, az első legrövidebb útvonalat, fix szakaszellenállásokkal számító gyors fa-eljárás" lényegét és a gyakorlati, programozási megvalósításának logikáját ismertetem egy egyszerű példa alapján. Ennek alapjául Loubal [4] algoritmusa szolgál, amit Scherr [5] gyorsító továbbfejlesztésével később a TRANSMAN is alkalmazott a BKV Forgalmi modellben, ill. TRANSURS modell rendszerben [6].

Mindez azért tűnik fontosnak, mert ma már a közlekedési modellezés döntően gyári szoftvercsomagok (pl. VISUM [7], EMME2 [8], …) alkalmazásával történik, és az egyes részmodellek – főleg a hálózati számítások magját jelentő útvonalkereső eljárások – sokak számára, mint „fekete doboz" működnek, holott az azokba beépült forgalomtechnikai összefüggések ismerete és szempontunkból azok működési logikája nélkülözhetetlen mind a szoftvereket alkalmazó „modell-futtatóknak", mind az eredményeket használó „társ-szaktervezőknek". Ugyanis bármely „futtatási modellparaméter" változtatása vagy valamely hálózati elem módosítása a modelleredmények változását vonja maga után. Ezért célszerű az útvonalkereső eljárásoknak és a hozzájuk kapcsolódó ráterhelési eljárásoknak a technikai és gyakorlati alapjait is ismerni, amihez a következő összeállítás igyekszik vázlatosan, de ugyanakkor a programozási folyamat szintjéig nyúló fogásokkal hozzájárulni.

3. AZ ÚTVONALKERESŐ ÉS RÁTERHELÉSI ELJÁRÁSOK LÉNYEGE

3.1. Közlekedési hálózati modell és igénymátrix – Példa

Az útvonalkereső és ráterhelési eljárás elveinek ismertetéséhez vegyük a 7 csomópontból álló kis példahálózatot és a hozzá tartozó mátrixot (ill. annak csupán az 1. csomóponthoz tartozó 1. sorát), amely a többi pontba (körzetbe) menő utazások számát (Fij) tartalmazza (1. ábra).

A példában szereplő ellenállások „rögzített értékek”, míg a valóságban azok „változó értékek”, amelyek függnek a hálózati elemeken megjelenő mindenkori közúti forgalom nagyságától, a geometriai kialakítás (pl. sávszám), a szabályozás módja, stb. által befolyásolt sebességtől vagy a közösségi közlekedésben a pályajellemzőktől és eszközöktől függő sebességtől, a menetrendi gyakoriságtól, a menet- és átszállási időktől, stb.

3.2. Útvonalfa-építés – a gyökérponttól előrefelé

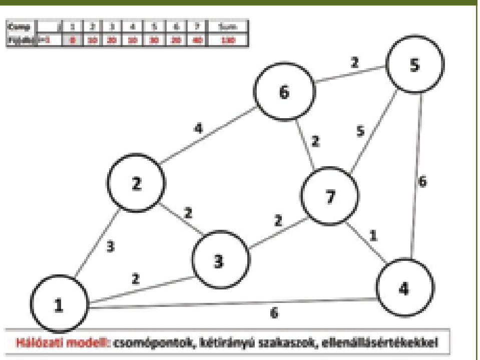
Az útvonalkereséshez, amely mindig egy adott gyökérpontból előrefelé „tapogatózva” a többi csomópont felé történik, ismerni kell még a hálózatban a „szomszédsági kapcsolatokat” (S(j,k)) és a szomszéd csomópontokig (k) menő szakaszok „ellenállás értékeit” pl. percben (D(j,k)), célszerűen az óra járásával megegyező irányban megadva (1. táblázat).

1. táblázat: Csomóponti szomszédsági kapcsolatok és szakasz-ellenállások

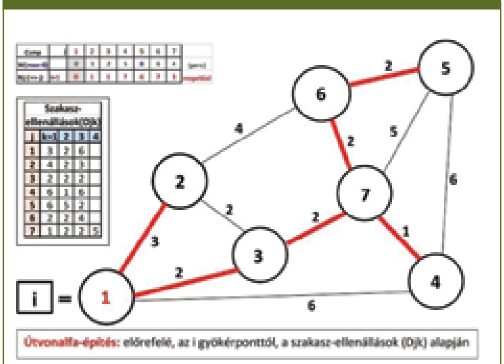
Szomszéd csmp-ok sorszáma (Sjk)				
j	**k=1**	**2**	**3**	**4**
1	2	3	4	
2	6	3	1	
3	1	2	7	
4	1	7	5	
5	4	7	6	
6	5	7	2	
7	4	3	6	5

Szakasz-ellenállások (Djk)				
j	**k=1**	**2**	**3**	**4**
1	3	2	6	
2	4	2	3	
3	2	2	2	
4	6	1	6	
5	6	5	2	
6	2	2	4	
7	1	2	2	5

1. ábra: Hálózati modell és az utazási igénymátrix (egy sora)



2. ábra: Az útvonalfa-építés eredménye adott gyökérpontból



A 2. ábra hálózati modelljének 7 csomópontja közül az i= 1. csomópontra, mint „gyökérpontra" vonatkozóan határozzuk meg az útvonalfát, azaz az egyes csomópontok (j=(1), 2.… 7) elérési ellenállását W(1j) és az odavezető legrövidebb útvonalat R(1j) eredményező vektorokkal.

A számítástechnikai megvalósításhoz, a közbülső keresési műveletekhez munkavektorokat (MW, MR) célszerű alkalmazni, amelyek hosszát a hálózat legtávolabbi pontjainak ellenállásértéke (dmax) határozza meg (gyakorlati, tár-takarékossági okokból ennél rövidebb „vektorméret" is meghatározható, ahol az értékek „revolvertárszerűen" cserélődnek).

Az útvonalkeresési eljárás során egy adott közbülső pontban (j) elért „ideiglenes" ellenállásértékhez (d(j) a szomszédos csomópontok ellenállását (D(jk)) hozzáadva nyerjük a következő, a már elért ponthoz legközelebbi pont vizsgálandó ellenállásértékét.

Az eljárás során a gyökérpontból nézve valamely csomópont elérési ellenállását (W) akkor mondhatjuk „véglegesnek", ha egy adott csomópontot (j) már minden szomszédja (k) felől megközelítettük és a legkisebb ellenállásértéket megállapítottuk (ld. ehhez a 2. táblázatot és az algoritmus folyamatábráját (3. ábra)).

Az MW és MR munkavektorok elemeit minden új fa megkezdésekor nulla értékűre (MW=0) ill. (MR=0) állítják be.

(Meg kell jegyezni, hogy a példában – a jobb szemléltethetőség érdekében – minden következő elért csomópont esetében külön MW és MR vektorokat vettünk fel, ahova „görgetve" fokozatosan beírtuk (megfelelő színekkel) a már megtalált csomópontokhoz tartozó adatokat is, hogy jobban kitűnjön, mi történik; a gyakorlatban egy-egy gyökérponthoz tartozóan csak egy-egy munkavektor létezik).

Ha a fában minden csomópontot „ledolgoztunk" – azaz a legrövidebb úton elértünk, – akkor a munkavektorokból kivesszük a „megfelelő értékeket" és eltesszük a végleges vektorokba (W; R); ez a gyakorlatban a példa 5. (legutoljára elért) csomópontja utáni állapotú MW-, ill MR-értékekből történik, amelyek az adott munkavektorok fokozatos „feltöltésével" nyerhetők.

3.3. A forgalmi áramok ráterhelése – a legtávolabbi pontból a gyökérpont felé

A hálózati ráterhelés a közlekedési mátrix viszonylati elemeinek (Fij) a ráhelyezését jelenti az „i" gyökérponthoz tartozó útvonalfa érintett szakaszaira.

Ehhez az ellenállás (W)- és útvonal (R)-vektorok mellett, – előnyös technikai fogásként – még ugyanabban a számítási menetben az „elérési sorrendvektort" (T) is meghatározzuk és fokozatosan feltöltjük. Ez a legtávolabbi csomópontból (5.) nézve, visszafelé, a W-értékek alapján a következő leghosszabb útvonalhoz tartozó végpont (6.) megadását jelenti, majd tovább a 4., a 7., aztán a másik ágon a 2. (az 1. gyökérpontig) és végül a 3. csomópont következik, amelyből már szintén elérjük az 1. gyökérpontot.

Az ily módon meghatározott T-vektor elemei biztosítják a forgalmi áramok „visszafelé", a gyökérpont felé való ráterhelésénél azt, hogy áramok ne maradjanak ki, ill. a már ráterhelt „áram-kötegekhez" az ellenállások szerinti csökkenő sorrendben még hátralévő csomópontok áramait hozzátéve, a gyökérponti szakaszok teljes terhelését megkapjuk.

4. AZ ÚTVONALKERESŐ ELJÁRÁS GYAKORLATI MEGVALÓSÍTÁSÁNAK LOGIKAI FOLYAMATA

Az útvonalkereső eljárás gyakorlati megvalósítása a kereső faalgoritmus számítógépes programba írását jelenti, amelynek logikai vázát ismertetjük (ld. folyamatábrát (3. ábra) is):

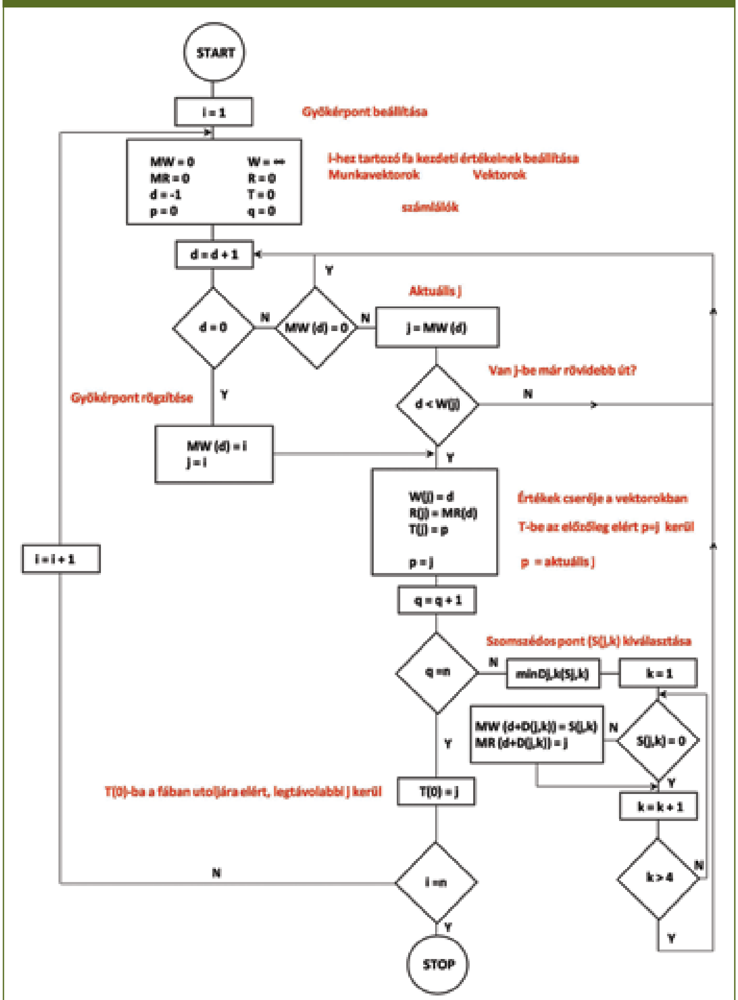
4.1. Kezdeti értékek beállítása

A hálózat elemeiből (i,j= 1,2,.. n) az első, i=1 csomópont kiválasztása és a vektorok kezdeti értékeinek beállítása:

– **d= (-1, 0, 1, 2,… dmax)** a gyökérponttól távolodóan, egységnyivel növekvő ellenállásértékek skálája (pl. percben), amelynek hosszúsága (L(d)) célszerűen nagyobb, mint dmax, a hálózat bármely két szélső pontja közti legnagyobb érték (L(d)>dmax),

– **MW(d) = 0;** munkavektor a célcsomópontok (j) ellenállásai (Wij) időleges tárolására; ennek elemei mindaddig változnak, amíg MW(d) nem lesz j-re nézve minimális; (ha a fakeresés végén a vektor valamely eleme MW(d) = 0, akkor az azt jeleni, hogy adott gyökérpontból nézve nem létezik olyan csomópont (j), amely i-től d távolságra van),
– **MR(d)=0;** munkavektor a célcsomópontokig menő útvonalak (Rij) időleges tárolására,
– **W(j)=999** (végtelen); ellenállásvektor a célcsomópontokig (j) menő legrövidebb útvonalak ellenállásértékeinek (Wij) „végleges"rögzítésére; az az érték, ahol MW(d)=dmin(j); (ha a fakeresés végén a vektor valamely eleme W(j)=999, akkor az azt jeleni, hogy adott gyökérpontból nézve nem létezik j-be útvonal, azaz szakadás van a hálózatban),
– **R(j)=0;** útvonalvektor a célcsomópontokig menő legrövidebb útvonalak (Rij) „végleges" tárolására; a vektorelemek útvonalcímkék, amelyek a legrövidebb útvonalfában a megelőző csomópontok sorszámait jelentik,
– **T(j)=0;** követő vektor, amelynek egy-egy eleme a legrövidebb útvonalfa mentén a csomópontok gyökérponttól való távolság szerinti csökkenő sorrendjét adja meg, a legtávolabbi ponttól (a T(0)-ban rögzítve) visszafelé a gyökérpontig, amely ismeretnek a „faösszehúzásos" ráterhelési eljárásnál van fontos szerepe,
– **p=0;** tároló elem, amely a legrövidebb útvonalfa mentén az éppen elért csomópont sorszámát őrzi meg (p=j), amíg a szomszédos csomópontok (k) közül a következő vizsgált csomópont a legrövidebb útvonalfában nem kerül megerősítésre; így ez a p=j a gyökérponthoz legközelebbi következő csomópont megelőzőjévé válik, és amely érték a követő vektor (T(j) elemét adja,
– **q =0;** számláló elem, a hálózatban található csomópontok db-számára vonatkozóan, (q=1,2,.. n; n=qmax).

3. ábra: Az útvonalkereső eljárás programozási folyamat-vázlata



4.2. Előrefelé tapogatózó, direkt útvonalkeresési megoldás

Érdemes elöljáróban megjegyezni, hogy az útvonalkereső eljárás során mind a d-skálafüggő munkavektorok (MW(d); MR(d)), mind a csomópont-azonosítóhoz kapcsolódó „végleges” vektorok (W(j), R(j), ill. T(j)) párhuzamosan használatban vannak, és elemértékeik folyamatosan bővülhetnek, módosulhatnak, amíg a hálózat minden pontja nem érhető el a legrövidebb úton (minW(j)), ill. az ezekhez az értékekhez tartozó útvonal- (R(j)) és követő vektort (T(j)) nem határozzuk meg.

4.2.1. A gyökérpont kezelése

A legkisebb sorszámú gyökérpont (i=1) választása után:

– indításként az MW(d=0) elembe beírjuk az i=j=1 értéket, és az MR(d=0) helyre is „0” érték kerül, hisz a gyökérpontnak nincs a fában megelőző csomópontja; ugyanezek az értékek a végleges ellenállásvektorokba W(j=1)=0 és az útvonal (rout-)vektorba R(j=1)=0 és a követő vektorba T(j=1)=0 is beírásra kerülnek, ill. a kezdeti értékek maradnak,

– ugyancsak az i=1 gyökérponthoz kapcsolódóan beírjuk a szomszédos csomópontok (S1k: 2; 3; 4) szakaszhosszai (Djk) alapján a vonatkozó értékeket: az ellenállás-munkavektorba sorrendben az MW(0+ D(1,2)=3)-nál a 2-es, az MW(0+(D1,3)=2 alapján a 3-as és az MW(0+(D1,6)=6-nál a 4-es csomópont sorszámát írjuk be, ill. a route-munkavektornál mindhárom esetben MR=1 kerül, hisz mindhárom szomszédos csomópontot az 1-es csomópontból értük el és a T(j=1)=0 értéket kap, ami azt jelenti, hogy ezek után már csak a gyökérpont van.

4.2.2. További pontok vizsgálata

A gyökérpont szinte mechanikus elintézése után:

– a szomszédos csomópontok (Sjk) közül a szakaszhosszak alapján a legközelebbi (minDjk=(D1,3=2)) az S1,3=3-as csomópontot, mint következő vizsgálandót választjuk ki,

– sorszáma (3) már szerepel a munkavektor d=2 hossznak megfelelő elemében (MW(2)=3), továbbá az ellenállásvektor j=3-hoz tartozó elemében W(j=3)=2; ily módon, mivel a gyökérhez a 3-as csomópont van legközelebb(d=D(1,3)=2 perc), biztos, hogy a legrövidebb úton van és W(3)=2 , továbbá R(3)=1, mint megelőző pont és T(3)=1 (magyarázata majd később),

– ugyancsak beírjuk a 3-as csomópont szomszédjainak (S3,k:1; 2; 7) megfelelő értékeit a munkavektorokba, így az MW(d=4) cellába a j=(1; 2; 7) értékek, hiszen ezek mind 2 távolságegységre vannak a 3-as ponttól és d=D(1,3)+ 2=4, ezeket időleges értékeknek tekintjük mindaddig, amíg meg nem bizonyosodtunk arról, hogy van-e hozzájuk rövidebb út is,

– így az 1-es pontra vonatkozóan az MW(d)=4 érték biztos, hogy kiesik hisz az 1-es pont távolsága a gyökértől 0!; (az eljárás a gyökérponthoz való visszautat is figyelembe veszi időlegesen, hisz „térképszerűen” nem ismert számára, hogy merre tapogatózik és a hosszabbnak bizonyuló kapcsolatot aztán később törli és figyelmen kívül hagyja),

– a 2-es pont MW(d)=4 értéke is törlődik, mivel a következő lépésben, mikor a 2-es csomópont szomszédjait (S2,k: 1; 3; 6) vizsgáljuk, kiderül, hogy a korábban, az 1-es pont szomszédjaként (D1,2=3) már beírtuk MW(d=3)=2 és W(j=2)=3 kapcsolat kedvezőbb, mint a 3-as csomóponton keresztül való MW (d=4)=2 ill. W(j=2)=4, ami azt erősíti meg, hogy W(j=2)=3, továbbá R(j=2)=1. Vagyis csak akkor találtunk legrövidebb utat, ha d<W(j) , azaz a gyökérponttól való ellenállás (W) az a legkisebb „d” érték, amelynél adott csomópont a legkorábban szerepel.

2. táblázat: A munkavektorok és az eredményvektorok párhuzamos, javításos használata

Munkavektorok (fokozatos javítással való feltöltés) Eredményvektorok

j=1.csp A gyökérpont közvetlen, szomszédos kapcsolatai — *A lehetséges útvonalak hossza d (perc)*

d	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
MW	1	0	3	2	0	0	4	0	0	0	0	0	0	0	0
MR	0	0	1	1	0	0	1	0	0	0	0	0	0	0	0

Az 1. gyökérhez a 3. csmp van legközelebb(d=D(1,3)=2 perc), így biztos, hogy a legrövidebb úton van

j	Kezd.ért.	1	2	3	4	5	6	7
W(j)	999»	0	3	2	6	999	999	999
R(j)	0 »	0	1	1	1	0	0	0
T(j)	0 »	0	0	0	0	0	0	0

j=3.csp A gyökérhez a legközelebbi csomópont kapcsolatai

d	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
MW	1	0	3	2	1,2,7	0	4	0	0	0	0	0	0	0	0
MR	0	0	1	1	3,3,3	0	1	0	0	0	0	0	0	0	0

A 3. csmp-ról már az előző lépésben kimondhattuk, hogy a legrövidebb útvonalon helyezkedik el

j		1	2	3	4	5	6	7
W(j)		0	3, 4	2	6	999	999	4
R(j)		0	1, 3	1	1	0	0	3
T(j)		0	0	1	0	0	0	0

j=2.csp A gyökérponthoz a következő legközelebbi csomópont kapcsoalatai

d	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
MW	1	0	3	2	1,2,7	3	4,1	6	0	0	0	0	0	0	0
MR	0	0	1	1	3,3,3	2	1,2	2	0	0	0	0	0	0	0

Miután a 2. csmp minden kapcsolatát megvizsgáltuk, mondhatjuk, hogy a legrövidebb úton van

j		1	2	3	4	5	6	7
W(j)		0	3, 4	2	6	999	7	4
R(j)		0	1, 3	1	1	0	2	3
T(j)		0	3	1	0	0	0	0

j=7.csp A gyökérponthoz a következő legközelebbi csomópont kapcsolatai

d	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
MW	1 2	0	3	2	1,2,7	3, 4	4,1, 3,6	6	0	5	0	0	0	0	0
MR	0,3	0	1	1	3,3,3	2, 7	1,2,7,7	2	0	7	0	0	0	0	0

Miután a 7. csmp-ba a legrövidebb uton fekvő 3.-on át jutottunk, az is a legrövidebb útvonalfa része

j		1	2	3	4	5	6	7
W(j)		0	3	2	6, 5	13	7, 6	4
R(j)		0	1	1	1, 7	7	2,7	3
T(j)		0	3	1	0	0	0	2

j=4.csp A gyökérponthoz a következő legközelebbi csomópont kapcsolatai

d	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
MW	1	0	3	2	1,2,7	3,4	4,1, 3,6	6,7	0	5	0	1	5	0	0
MR	0	0	1	1	3,3,3	2, 7	1,2,7,7	2,4	0	7	0	4	4	0	0

Miután a 4. csmp-ba a legrövidebb uton fekvő 7.-en át jutottunk, az is a legrövidebb útvonalfa része

j		1	2	3	4	5	6	7
W(j)		0	3	2	6, 5	13, 12	7, 6	4
R(j)		0	1	1	1, 7	7, 4	2,7	3
T(j)		0	3	1	7	0	0	2

j=6.csp A gyökérponthoz a következő legközelebbi csomópont kapcsolatai

d	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
MW	1	0	3	2	1,2,7	3,4	4,1,3,6	6,7	7, 5	5	2	1	5	7	4
MR	0	0	1	1	3,3,3	2, 7	1,2,7,7	2,4	6, 6	7	6	4	4	5	5

Miután a 6. csmp-ba a legrövidebb uton fekvő 7.-en át jutottunk, az is a legrövidebb útvonalfa része

j		1	2	3	4	5	6	7
W(j)		0	3	2	5	13, 12, 8	6	4
R(j)		0	1	1	7	7, 4, 6	7	3
T(j)		0	3	1	7	0	4	2

j=5.csp A gyökérponthoz következő legközelebbi csomópont kapcsolatai, j=n? - ha igen > végleges vektorok

d	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
MW	1	0	3	2	1,2,7	3,4	4,1,3,6	6,7	7, 5	5	2	1	5	7	4
MR	0	0	1	1	3,3,3	2,7	1,2,7,7	2,4	6,6	7	6	4	4	5	5

Miután a 5. csmp-ba a legrövidebb uton fekvő 6.-on át jutottunk, az is a legrövidebb útvonalfa része

Végleges vektorok

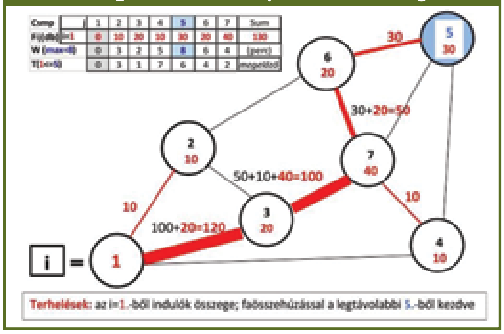
j		1	2	3	4	5	6	7
W(j)		0	3	2	5	8	6	4
R(j)		0	1	1	7	6	7	3
T(j)	5	0	3	1	7	6	4	2

Magyarázat: 3 csomópont legkorábbi helye a d-skálán; 1 megelőző csomópont sorszáma * (R); 3 csomópont; 2 ellenállása (W); 1 visszakövető címke (T)

5. RÁTERHELÉS – A FORGALMI ÁRAMOKNAK A GYÖKÉRPONT FELÉ VALÓ ÖSSZEHÚZÁSÁVAL

A 4. ábra az 1. gyökérponthoz tartozó utazási áramelemek (F1j) ráterhelését szemlélteti „fa-összehúzásos” technikával. Itt most a teljes mátrix-értékek ráterhelése, a „minden vagy semmi”-elv alapján kerül bemutatásra.

4. ábra: A munkavektorok és az eredményvektorok párhuzamos, javításos vizsgálata



Ebben a műveletben, a ráterhelésnél a „T” vektor nagy hasznunkra van, ugyanis segítségével nem kell minden F(ij) értéket külön-külön végig terhelni a saját legrövidebb útvonala mentén lévő szakaszokra, ami nagyhálózatok esetén nehezen elképzelhető, hanem ezt egy menetben, a fogalmi áramok „összehúzásával” lehet megoldani, és a legtávolabbi facsúcstól (5.) jőve, az aktuális áramnak, a már elintézett „mögöttes” mátrixelemek „kötegéhez” való hozzáadásával elég csupán egyszer végig menni a fa ágai mentén, a T-vektor által megadott sorrendben; így a példában:

- az 5.-6. szakaszra rákerül az F(1,5) forgalmi áram értéke (30 utazás),
- a 6.-7. szakaszra rákerül az F(1,5) + F(1,6)-értékek összege (30+20=50 utazás),
- a 4.-7. szakaszra rákerül az F(1,4)-érték (10 utazás),
- a 7.-3. szakaszra rákerül az F(1,5) + F(1,6) és az F(1,4) összege (50+10), valamint az F(1,7)-érték (40), összesen 60+40=100 utazás,
- a 2.-1. szakaszra rákerül az F(1,2)-áram (10 utazás),
- a 3.-1. szakaszra rákerül az F(1,5) + F(1,6) + F(1,4) + F(1,7) -értékek összege (100), valamint a F(1,3) utazások (20) összesen 100+20=120 utazás.

Ily módon az i=1. fa terhelése rendelkezésre áll. Ezt a műveletsort valamennyi csomópontra, mint gyökérpontra elvégezve megkaphatjuk a teljes hálózat terhelését és ábráját.

Ez persze csak az első legrövidebb utas, egylépéses „minden, vagy semmi” elvű ráterhelésnél történik így, amikor akár a közúti, akár a közösségi közlekedési hálózaton a forgalmi terhelések „előzetes” értékeire vagyunk kíváncsiak.

A közösségi közlekedésnél, ahol menetrend szerint, egy-egy szakaszon több viszonylat is haladhat és a viszonylatok közti utasmegoszlást is meg kívánjuk határozni, akkor az áramonkénti ráterhelés jöhet szóba, megfelelő megosztási módszer alkalmazásával, akárcsak a többútvonalas eljárásoknál.

A közúti ráterheléseknél, ha a forgalmi körülmények, ill. kapacitáskihasználtság ellenállás-befolyásoló hatását is figyelembe vesszük, akár „%-os részletekben” való, – túlterheléseket is kockáztató –, akár „egyensúlyi állapotra törekvő” eljárást kívánunk alkalmazni, akkor többlépéses – közben az ellenállások változását figyelembe vevő – ráterhelési eljárásra van szükség [1, 2, 6, 7, 8].

6. ZÁRÓ GONDOLATOK

Ma a gyári szoftvercsomagok (pl. VISUM [7], EMME2 [8], …) már különböző útvonalválasztási és ráterhelési stratégiáknak megfelelő eljárásokat tartalmaznak. Az eljárások legtöbbje, mint „fekete doboz” működik, amelyek mindegyikének valamilyen legrövidebb útvonalas eljárás az alapja, és ezért került sor egy ilyen „alapeljárás”, részletes, a gyakorlati megvalósítást is lehetővé tevő, ismertetésére. Továbbá azért is, mert létezhetnek olyan, közlekedési hálózatokon lejátszódó jelenségekkel kapcsolatos optimalizálási feladatok, amelyek nem teszik lehetővé valamely gyári szoftvercsomag útvonalkereső eljárásának közvetlen alkalmazását, hanem egyedi megoldást kívánhatnak, mely esetekben az útvonalkereső algoritmus feladathoz illeszkedő programozására lehet szükség.

Akit részletesebben érdekel az útvonalkereső eljárások fejlődése és alkalmazása, az Schrijver átfogó tanulmányából [9] megtudhatja, hogy az eljárások (pl. Shimbell, Ford, Moore, Dijkstra, Lee,... ismertebb nevekhez köthető néhány) először a telefon- és ideghálózati kapcsolatok vizsgálata kapcsán merültek fel és csak később a közlekedési hálózatoknál.

Megállapítható, hogy több tízezer szakaszt és csomópontot tartalmazó nagyhálózatokra alkalmas, gyakorlati használatba is került hazai közlekedési vonatkozású eljárás és program korábban nem sok készült, ezért méltányos megemlíteni a működő ill. működött programok kapcsán Scherr Károly (KTI), Nagy Károly (UVATERV), Marton László (SzTAKI-SzE) fejlesztési munkásságát.

FELHASZNÁLT IRODALOM

[1] Sheffi, Y.: Urban Transportation Networks (Prentice Hall, Engelwood Cliffs N.J., 1985)

[2] Ortuzar, J.de D. – Willumsen, L.G.: Modelling Transport (John Wiley& Sons, 1990)

[3] Horváth B. – Koren Cs. – Prileszky I. – Tóth-Szabó Zs.: Közlekedéstervezés (Jegyzet - Széchenyi István Egyetem, Győr)

[4] Loubal, P.: A Network Evaluation Procedure (Highway Research Record 205, 1967)

[5] Scherr K.: Ráterhelési modell (Városi Közlekedés 1977/ 4.-5. szám)

[6] Monigl J. – Koren T. – Nagy E. – Ujhelyi Z. – Berki Zs.,..: Budapest és környéke közlekedésének tér-idő-költség-elvű modellezése és hatásai átfogó értékelése (TRANSURS modellrendszer) (TRANSMAN-tanulmány, 1998)

[7] VISUM – http://vision-traffic.ptvgroup.com/en-us/products/ptv-visum/

[8] EMME2 – https://www.inrosoftware.com/en/products/emme/

[9] Schrijver, A. (2012): On the History of the Shortest Path Problem Documenta Mathematica · p155–167)

Routing and traffic assignment methods in transport networks

The "basic procedures" of routing and traffic assignment are important steps of transport modelling. The described methods are suitable for the schematic understanding of the operating principles of softwares, and, if applicable, for their practical implementation. The genuine software packages include appropriate procedures for different strategies, but from the point of view of users they are functioning as a "black box".
The study classifies the types of various procedures. Besides this, it describes the main features of the technical bases for the “model users” who apply the software, and for the “professional co-planners”, based on the example of a routing and the related assignment procedure. At the same time the article attempts to relate information and knowledge to the interested persons for the coding procedure.

Routensuch- und Verkehrsumlegungsverfahren in Verkehrsnetzen

Die "Grundverfahren" der Routensuche und der Verkehrsumlegung sind wichtige Schritte der Verkehrsmodellierung. Die beschriebene Verfahren eignen sich für das skizzenhafte Verstehen der Funktionsgrundsätze der Software und ggf. für ihre praktische Umsetzung. Die standard Software-Pakete enthalten schon entsprechende Verfahren für verschiedene Strategien, aber sie funktionieren in der Regel für die Benutzer als "Black Box". In der Studie wird eine schematische Klassifizierung der verschiedenen Verfahrenstypen gegeben, und es werden dabei die Hauptmerkmale der technischen Grundlagen für die "Modellbenutzer", die die Software anwenden, und für die "professionellen Co-Planer", durch das Beispiel einer Routensuche und dem zugehörigen Umlegungsverfahren beschrieben. Gleichzeitig werden den Interessenten Informationen und Kenntnisse für die Programmierung der Verfahren vermittelt.