Apáthy M. Sándor

Budapesti Corvinus Egyetem
Matematika Tanszék
Témavezető: Tallos Péter, CSc

Budapesti Corvinus Egyetem

Közgazdaságtudományi Doktori Iskola

Egy turisztikai ajánlórendszer modellje

PhD értekezés tervezet

Apáthy M. Sándor

Budapest, 2016

Nyilatkozat

Alulírott Apáthy M. Sándor kijelentem, hogy ezt a doktori értekezést magam készítettem, és abban csak a megadott forrásokat használtam fel. Minden olyan részt, melyet szó szerint, vagy azonos tartalomban, de átfogalmazva más forrásból átvettem, egyértelműen a forrás megadásával megjelöltem.

Budapest, 2016.02.15.

Apáthy M. Sándor

Declaration

Hereby, I, Sándor Apáthy M. declare that the present PhD Thesis is my own work, and I utilized only the sources indicated within. All parts taken from other works used word by word, or in a reedited way keeping the original contents, have been unambiguously marked by a reference to the source.

Budapest, 15.02.2016.

Sándor Apáthy M.

Tartalomjegyzék

Édesanyámnak

1. Bevezetés

Kevés rémesebb érzést tudok elképzelni, mint eltévedni egy idegen városban, főként, ha nyelvi akadályok miatt még segítségkérésre sem igen van lehetőségünk. Bár ez velem is megtörtént több alkalommal, sokkal inkább az ösztökél turisztikai ajánlórendszer tervezését célzó kutatásaim során, hogy az újdonság különleges és lebilincselő érzésének megélésében segítsek másokat. Abban, mikor egy addig számukra ismeretlen kultúrával találkoznak, vagy egy addig még sosem látott építészeti megoldással egy ház homlokzatán. Nem titkolt célom, hogy jelen kutatásra alapozva a későbbiekben mobil alkalmazás formájában is megméressenek eredményeim. Képtelen lennék elfogadni, hogy olyan kutatásba fektessek energiát, mely nem bír gyakorlati hasznossággal. Amennyiben egy majdani mobil alkalmazás akár csak egy felhasználót is hozzásegít egy olyan - pozitív - élményhez, mely annak hiányában elkerülte volna, már megérte a munka. Az alábbiakban a kutatás aktuális állapota kerül bemutatásra.

A témaválasztással kapcsolatos motivációkat, és annak relevanciáját - rendhagyó módon - az egyes fejezetek elején ismertetjük az adott témakörre koncentrálva. Fontos megemlíteni, hogy a dolgozat néhány kivételtől eltekintve öbbesszám első személyben íródott, de ez nem társszerzőkre utal, csupán a szerző stilisztikai döntése.

1.1. A kutatás célja

Életem során megannyi várost volt szerencsém bebarangolni, ám gyakran jelentett komoly kihívást olyan túrák megtervezése, melyek a rendelkezésemre álló idő minél hatékonyabb kihasználását teszik lehetővé. **Jelen dolgozat célja egy olyan turisztikai ajánlórendszer elméleti alapjainak lefektetése, mely képes elegendően pontos és személyre szabott útvonalak tervezésére egy idegen városban, figyelembe véve a felhasználó igényeit és lehetőségeit.** Ezen igények és lehetőségek megismerése, és modellbe történő beépítése nem magától értetődő feladat, a dolgozat szerzője azonban erre tesz kísérletet. Elsőként a turisták sebességének becslését célozzuk, adott körülményeket (mint például az út meredeksége, vagy épp pillanatnyi fizikai erőnléte) figyelembe véve, mellyel személyre szabottan tudjuk két helyszín közötti menetidejét becsülni. Ezt követően egy olyan modellt építünk, mely néhány információ alapján igyekszik a felhasználó ízlésvilágáról minél pontosabb képet alkotni, az ajánlórendszerek köréből származó módszerek segítségével. A továbbiakban ez lehet segítségünkre abban, hogy megértsük, a túra során relatíve mennyire

értékelne egy adott helyszínt, vagy nevezetességet az útvonalba beillesztve. Amennyiben a fenti információk már a rendelkezésünkre állnak, **lehetőségünk nyílik egy olyan útvonaltervező algoritmus megalkotására, mely ezeket felhasználva a túrázó igényeihez mérten leginkább testreszabott túrát tud ajánlani egy adott városban.**

A tézis során az alábbi kérdéskörökre igyekszünk választ találni:

1. Hogyan lehet a GPS alapú túranaplók adatait pontosabbá és elemzésre alkalmassá tenni? Létezik olyan digitális emelkedési modell (DEM), mely a valós magasság értékeket jól közelíti?

A GPS eszközök pontatlanságából adódóan korrekcióra szorulnak a szélességi- és hosszúsági értékek, valamint a magassági adatok. Előbbit az adatpontok korrigálásával, valamint Kálmán-filter segítségével pontosítjuk (2.5.1-es alfejezet), utóbbit két általunk alkotott digitális földfelszín modell bevezetésével (2.4-es alfejezet).

2. Létezik a sebesség és az adott szakasz meredeksége közötti összefüggést leíró ismert megoldásoknál pontosabb?

A gyalogos menetidőbecslő eljárások szakirodalma igen szűkös, melynek vélhetően egyik legfőbb oka, hogy igen nehéz jó minőségű túranaplókhoz hozzájutni. A tanulmányozott szakirodalom alapján kijelenthetjük, hogy jelen dolgozatban az eddigi legnagyobb adathalmazon végezzük a meredekség és sebesség közötti összefüggés becslését (2.5.3-as alfejezet), mely a korábbiaknál nagyobb pontosságot eredményez.

3. Milyen eljárással adható a túrázók menetidejének egy a jelenlegieknél pontosabb becslése?

Jelen dolgozatban két menetidőbecslő eljárás is bemutatásra kerül, melyek pontossága jóval túlszárnyalja az eddig ismert megoldásokat. Az első eljárás a korábban megismert meredekség és sebesség közötti becsült összefüggésre alapoz, melyet fittséget leíró korrekciós tényezővel pontosítunk, míg a második eljárás a korábban tapasztalt sebességekre alapozza becslését. A két eljárást a 2.5.4-es alfejezetben ismertetjük.

4. Hogyan lehet kevés kezdeti információ alapján turisztikai célú ajánlórendszert építeni, mely személyre szabott ajánlásokat tesz a felhasználónak?

Annak érdekében, hogy minimális kezdeti információ felhasználásával tudjunk ajánlásokat tenni, 17 turisztikai faktort határoztunk meg, melyek a látványosságokat hivatottak leírni, továbbá az empirikus vizsgálathoz létrehoztunk egy 3 város látványosságait felölelő adatbázist, melyben

klasszifikáljuk a látványosságokat a fent említett faktorok segítségével (3.7-es alfejezet). Az így megalkotott hibrid ajánlórendszer néhány kezdeti kérdésre adott válasz alapján képes ajánlásokat tenni.

5. Lehetséges a turistákat klasszifikálni a kezdetben magukkal kapcsolatban megadott információk alapján? Hogyan építhető fel ennek érdekében egy személyre szabott ajánlásokat adó rendszer kérdőíve?

Empirikus vizsgálatunk során sort kerítettünk a kérdőív kitöltőinek szegmentálására (3.8-as alfejezet), mely alapján szignifikánsan jobb ajánlásokat tudtunk adni, mint annak hiányában. A kérdőívben kulcs szerepet játszik a 17 turisztikai faktor meghatározása (3.7.6-os alfejezet).

6. Milyen hasznosságfüggvénnyel írható le általános módon a felhasználók látványosságokra vonatkozó értékelése? Milyen célfüggvény vezet a gyakorlatban a felhasználók számára leginkább elfogadható útvonaltervezéshez?

Az utazóügynök probléma óta minden útvonaltervező feladat célfüggvényét a gráf csúcsaiban gyűjtött profitok összegeként definiálják. Tesztjeink során ez a felhasználók elégedettsége szempontjából kifejezetten gyenge eredményekre vezetett, így egy minden korrábitól eltérő hasznossági- és célfüggvény került meghatározásra (4.4.2-es alfejezet), melynek célja a felhasználói igények minél pontosabb leírása.

7. Milyen algoritmus adható a kitűzött útvonaltervezési feladatra, mely alacsony futási idejével alkalmazhatóvá teszi azt egy valós applikációban?

Az általunk kitűzött útvonaltervezési feladatra egy olyan heurisztikus algoritmust adtunk megoldásként, mely első lépésként lecsökkenti a feladat méretét, majd tovább egyszerűsíti azt klaszterek kialakításával. Ezek segítségével még a viszonylag magas számításigényű optimalizáló szakasz ellenére is rövid futási időt sikerült elérni. Az algoritmust a 4.4.4-es alfejezetben ismertetjük.

1.2. A tézis felépítése

Jelen dolgozat a fentiekben megismert kutatási célokat követve hármas tagolású.

A 2. fejezetben kísérletet teszünk a menetidőbecslés modellezésére, melyhez a szakirodalom alapos tanulmányozása során sem tapasztalt mennyiségű túranapló adatot használtunk fel. A túranaplók

előkészítéséhez szükséges megbízható magasságadatok becslésére egy saját földfelszínt közelítő modellt építünk NASA adatokra alapozva, majd az így nyert magasság értékeket rendeljünk a meglévő szélességi és hosszúsági adatokhoz. Ezután Tobler korábbi munkája nyomán újrabecsüljük a sebességet az útszakasz meredekségével leíró függvényét, és ezt használjuk fel személyre szabott menetidő becslésre, illetve egy másik, átlagsebességeken alapuló eljárást is bemutatunk.

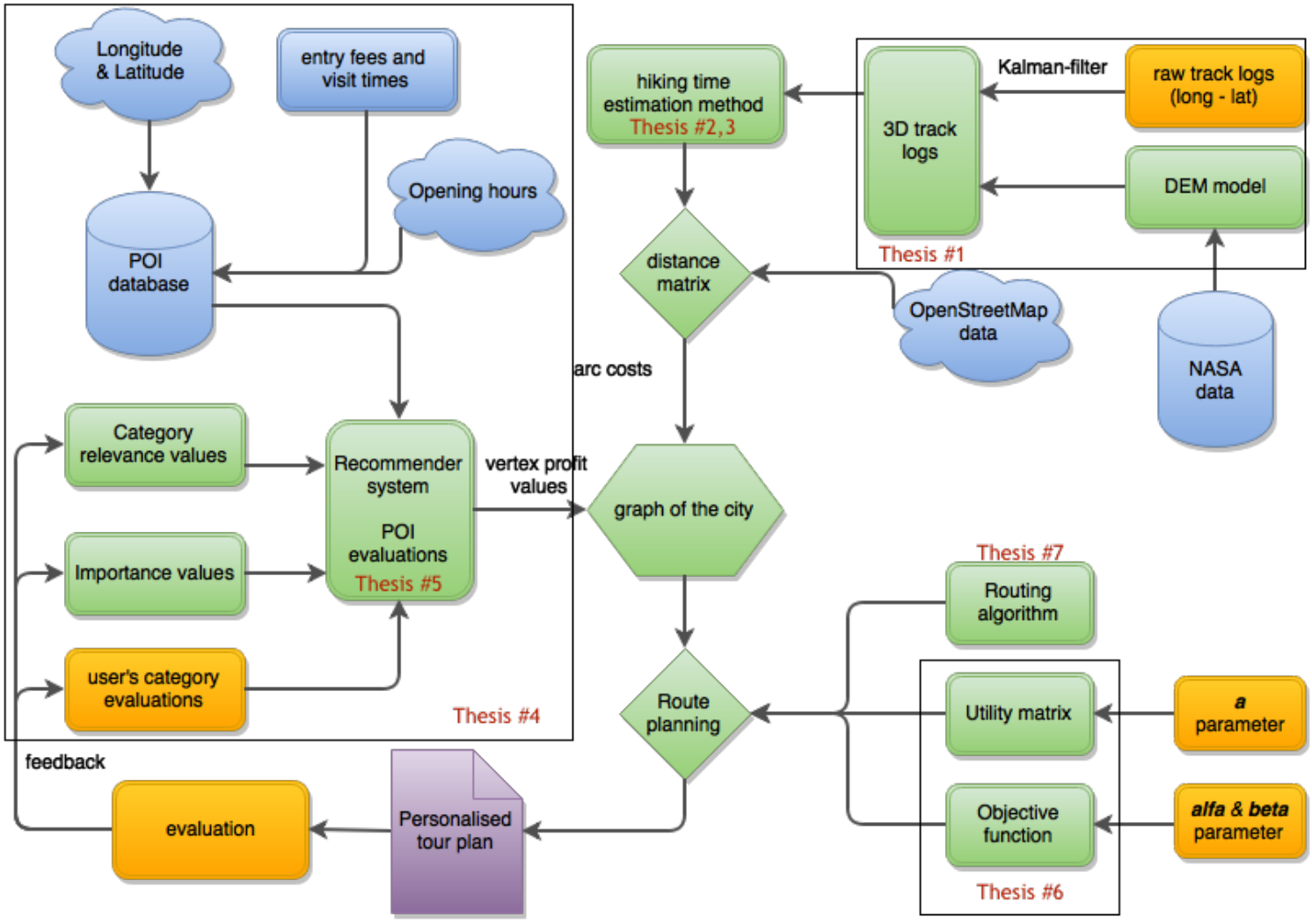
A turisták igényeinek személyre szabott kielégítése érdekében képesnek kell lennünk a preferenciáik feltérképezésére, és ezt figyelembe véve megállapítani az egyes helyszínek meglátogatásához kapcsololódó relatív profitokat. Ennek érdekében a 3. fejezetben bemutatásra kerül az ajánlórendszerek széles szakirodalma, valamint néhány jelenleg használatos turisztikai ajánlórendszer, és ezeket figyelembe véve alkotjuk meg saját ajánlások adására alkalmas hibrid modellünket. Ez egyszerre épít a felhasználóktól kapott információkra, mellyel feltérképezhetjük preferenciáikat, illetve látványosságokra, mint termékek komplex struktúrájára, mely segít megérteni azok szerkezetét. Az empirikus vizsgálat során gyűjtött információk segítségével módunk nyílik a turisták tipusainak meghatározására, mely lehetőséget ad az ajánlások pontosítására.

A 4. fejezetben egy, a turista igényeihez alkalmazkodni képes útvonaltervező algoritmust alkotunk meg. A modellezés során a várost egy N csúcsú irányítatlan gráffal jellemzünk, melynek minden csúcsa egy-egy potenciálisan meglátogatandó látványosságot jelképez, míg az élek a végpontjaikat összekötő legrövidebb utat hivatottak leírni. Az egyes csúcsokban begyűjthető profitokat a 3. fejezetben megalkotott ajánlórendszerünkből származtatjuk, így azok speciálisan az adott turista igényeihez igazodnak. A gráf élköltségei nem mások, mint a turista számára az adott két pont között vezető legrövidebb út menetideje. A feladat, hogy P nap alatt (napi T órában) olyan uta(ka)t járjon be a gráfon, mellyel a célfüggvénye értékét maximalizálja. Külön figyelmet szenteltünk egy olyan célfüggvény megalkotásának, mely merőben eltér az útvonaltervező algoritmusok gyakorlatától annak érdekében, hogy a felhasználók számára leginkább tetsző útvonalat legyünk képesek tervezni. Az *1. ábrán* foglaltuk össze a tervezés során használt adatforrásokat (ezeket kékkel jelöltük, és felhő alakú, amennyiben adatbányász eljárással jutottunk hozzá az internetről), a narancs színnel jelöltük a felhasználótól származó bemeneti adatokat, és zölddel az általunk kalkulált elemeket. Ahogyan az látható, a várost leképező gráfhoz három adat szükséges, a látványosságok helyzeti adatai, amit az OpenStreetMap alkalmazásból nyerünk, az élköltségek, amit a távolságmátrix szolgáltat, valamint a csúcsokban gyűjthető profitok, melyeket az ajánlórendszerből nyerünk. A távolságmátrix a menetidőbecslő eljáráson alapszik, melynek modellezése során a turautak.hu-tól szerzett túranaplókat használtuk fel a saját digitális földfelszín modellünk mellett (ami NASA adatokon alapszik). Az ajánlórendszerhez felhasználjuk az általunk

épített adatbázist a látványosságokkal és azok attributumaival (úgy mint helyzeti adatok, látogatási- és nyitvatartási idők, valamint belépődíjak), továbbá a felhasználók 17 kategóriára tett értékelését és a látványosságok kategóriákra vonatkozó relevanciaértékeit és "fontossági" paramétereit. Az így előálló gráfon tudjuk alkalmazni útvonaltervező algoritmusunkat, melynek kulcs elemei a felhasználók által adott paraméterek (pl. lustaság), és az általunk elkészített hasznossági- és célfüggvény, melyet a feladat során maximalizálni igyekszünk. Az algoritmus generálta személyre szabott útvonaltervre adott értékelés folyamatos visszacsatolást ad a rendszernek, mely tovább pontosítja az ajánlásokat.

Az 5. fejezetben a kutatási eredményeket foglaljuk össze, míg a kutatás lehetséges jövőbeli irányai témakörönként az adott fejezetek végén kerülnek kijelölésre.

1. ábra: A turisztikai ajánlórendszer logikai felépítése



2. Túrautak menetidejének becslése

2.1. Bevezetés

A digitális korban az emberek egyre inkább élik mindennapjaikat tervezetten. Nem kivétel ez alól a szabadidejük sem, melyet egyre inkább kívánnak "hatékonyan" eltölteni, ám kevés időt szánnának annak megtervezésére. Az okostelefonok elterjedésével egyre inkább életünk részévé válnak a tevékenységeinket megkönnyíteni szándékozó mobilalkalmazások. Tanulmányunkkal a túraútvonal tervező applikációk menetidő becslésének pontosságát kívánjuk javítani.

Az optimális útvonalat kereső algoritmusokat olyan gráfon értelmezzük, melynek csúcsai a meglátogatható lokációk halmaza, míg élei a lokációkat összekötő útszakaszok. Az élköltségek jellemzően az útszakasz megtételéhez szükséges időt jelölik. Ebben a tanulmányban ezeknek az élköltségeknek a minél precízebb becslését tűzzük ki célul nem titkolva, hogy ezt egy teljes útvonaltervező ökoszisztéma részének tekintjük.

Bár feljegyzések alapján már az ókori Rómában is éltek feltevésekkel a hadsereg haladási sebességével kapcsolatban, az első túrázóknak szóló menetidő becslés a XIX. század végéről származik, melyet Naismith-szabályként ismerünk. Azóta ennek finomítására sokan tettek kísérletet statisztikai és ökonometriai módszerek széles skálájával, melyet a 3. szakaszban tekintünk át, illetve itt kerül sor a téma tudománytörténeti elhelyezésére is.

Mint azt látni fogjuk, a menetidőt annyi környezeti- és emberi tényező befolyásolja, hogy pontosabb becsléshez ezek figyelembevétele elengedhetetlen. Célunk egy olyan becslőfüggvény megalkotása volt, mely szignifikánsan jobb eredményt ad a Naismith-szabály alapján kapott becslésnél a tesztadatokon. A vizsgálatokat Arthur Pitman et al. nyomán egy 360 elemű, teljes túraútvonalat tartalmazó mintán hajtjuk végre. A 4. szakasz több alfejezetre tagolódik, mivel itt kerül bemutatásra a vizsgálatok "melléktermékeként" elkészült földfelszín modell, mely önmagában is önálló eredményként értelmezhető. Ezt követően az 5. szakaszban Tobler nyomán a sebesség becslését pusztán meredekségi adatokra építve végezzük: turisták túranaplóit (tracklog) alapul véve fogunk a tanulóhalmazon a talaj meredeksége és a túrázó sebessége közötti összefüggést becsülni. Az így kapott becslőfüggvényt tovább finomítjuk a túrázó képességeinek klasszifikálásával, mely nyilvánvalóan befolyásolja a menetidejét. Ez követően bemutatjuk a tesztadatokon elért eredményeinket, és a becslés pontosságát demonstrálandó összevetjük azokat a teszt adathalmazon a Tobler-görbe alapján kalkulált becslésekkel. Végezetül levonjuk következtetéseinket a teszteredmények alapján, és kijelöljük a kutatás további lehetséges itányait.

2.2. Motiváció és a téma relevanciája

Bár gyerekkorom óta járok túrázni rendszeresen, mindig saját tapasztalataink alapján mértük fel, milyen útvonal teljesíthető számunkra, vagy mennyi idő szükséges egy adott útszakasz megtételéhez. Bevallom, arra számítottam, hogy a technika előrehaladásával születnek olyan megoldások, melyek képesek ezeket a "tapasztalati becsléseket" pontosítani, és egyúttal kényelmesebbé, könnyebbé teszik a navigációt a terepen. Valóban született néhány alkalmazás (például Strava, Endomondo, Locus, Komoot), mely segít a tervezésben, és a terepen való eligazodásban, azok pontatlansága és funkcionalitásaik hiányossága mégis arra sarkallt, hogy magam is elmélyüljek a menetidő becslési eljárásokban, és megpróbálkozzak azok pontosításával. A kutatás során rá kellett jönnöm, hogy a kollégáim legfőbb problémája az adatok, vagyis inkább a megbízható adatok, hiánya. Ennek leküzdése további lendületet adott, hogy a kitűzött célokat elérjem, és a jövőben ez a kutatás akár egy saját alkalmazás alapját képezhesse, mely - reményeim szerint - több gyakorlatban felmerülő kérdésre tud választ adni. Sajnos nem áll ma rendelkezésünkre olyan mobil alkalmazás, mely a menetidőbecslés mellett - legyen az bármilyen pontos - akár csak annyit is képes volna jelezni, hogy a tervezett út hátralevő része naplemente előtt már nem fejezhető be, és ezért alternatív útvonalat javasol. De felmerülhet valakiben az igény arra is, hogy korábbi eredményei alapján egy számára megfelő meredekségi profilú útvonalat tervezzen az eszköz, de legalábbis jelezze, ha az illető erején felül akar vállalni.

Természetesen nem szükségképpen kell a turizmus területére szorítkoznunk, mikor a menetidőbecslés relevanciáját kutatjuk: a tömegközlekedésben, a szállításban, légiforgalom irányításban éppúgy kulcs szerepet tölt be, ahogy a környezetvédelemben és a régészetben, mint azt a későbbiekben látni fogjuk.

2.3. Kapcsolódó kutatások

Az útvonaltervező algoritmusok a gráf éleit, mint lehetséges célállomások közötti élköltséget többnyire adottnak tekintik. Ez az élköltség jellemzően nem más, mint a két pont között megtett út menetideje, bár a szakirodalomban számos példa akad arra, hogy inkább a ráfordított energiát tekintik az út költségének (lásd [177], [178]). A menetidő becslését (Estimated Time of Arrival, röviden ETA) sokan és sok speciális területen kísérelték meg. A jelen tanulmány középpontjában a túrautakkal kapcsolatos menetidő becslés áll, mely alapját képezi a későbbi optimális útvonal tervezéseknek.

2.3.1 Turizmus

Már az ókori Római Birodalomban hadviselésében is nagy hangsúlyt fektettek a várható menetidő becslésére: *"A római légió katonáinak 24 mérföldet kell megtenniük 5 óra alatt a standard katonai lépést alkalmazva"*, olvashatjuk Vegetius *De Re Militari* c. művében [37]. A túrautakra vonatkozó menetidő becslés William Naismith, skót hegymászó 1892-ben meghatározott ökölszabályával kezdődött [25], mely szerint 1 óra alatt 3 mérföldet (4827,9 méter) tud megtenni egy "átlagos" kondícióval bíró személy, "tipikus" terepviszonyok mellett és "normál" körülményeket feltételezve (hőmérséklet, páratartalom, szél, stb.), míg minden 2000 láb (632 méter) emelkedő további 1 órát vesz igénybe. A gyakorlatban tehát a sík terepen és emelkedőn való mozgás ekvivalenciáját mondja ki, vagyis 1 egység emelkedő 7,92 egység sík terepen megtett távolsággal egyenlő idő alatt teljesíthető (ezt szokás 1:8 szabályként is emlegetni). Negatív meredekségű lejtőkön sajnos a szakirodalomban sok helyen a vízszintes felszínre vonatkozó becsléssel élnek a kutatók, például Scarf [152], vagy Verriest [179].

Mills [180] állítása szerint a Naismith-szabály Colin MacLaurin skót matematikustól eredeztethető az 1740-es évekből, aki megállapította, hogy a taposómalomban dolgozó férfiak 30 fokos lejtőn tartósan nagyjából 1 láb/sec (31,6cm/sec) sebességgel tudnak haladni felfelé, vagyis óránként kicsivel több mint 1800 láb (568,8 méter) emelkedőt tesznek meg (vertikális irányban). Akárhonnan is eredeztethető a fenti menetidő becslés, úgy tűnik, mindenképpen Skóciát illeti az érdem.

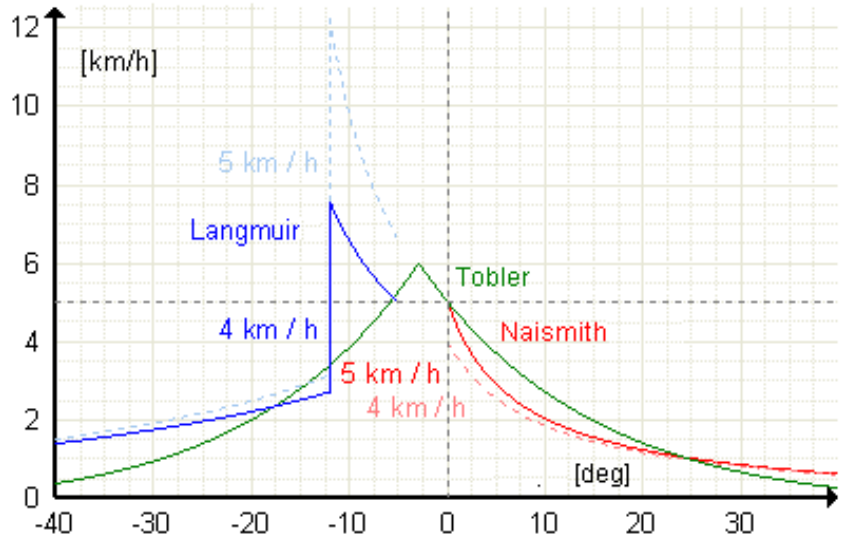
Az idők folyamán megannyi módosítási javaslat született:

- Aitken [27] feltevése szerint úton és ösvényen elfogadható a Naismith-szabály, de minden egyéb felületen 20%-kal gyengébben teljesít a túrázó.
- Langmuir [26] Naismith becslését ambíciózusnak tartotta, és 4km/h sebességet feltételezett sík terepen (+-5 fok eltérés esetén), továbbá javasolja, hogy minden 300m-en csökkentsük a becsült menetidőt enyhe lejtőn (5-12 fok között) és növeljük 10 perccel minden 300m-en meredek lejtőn (12 foknál nagyobb). A Langmuir által a menetidő becslésére javasolt függvény a következők szerint alakul: $T = a \times \Delta H + b \times \Delta V_a + c \times \Delta V_{dm} + d \times \Delta V_d$, ahol ΔH a horizontális elmozdulás, ΔV_a a vertikális emelkedés, ΔV_{dm} a vertikális moderált ereszkedés, és ΔV_d az erős ereszkedés, míg $a=0,72$; $b=6,0$; $c=1,9998$ és $d=-1,9998$.
- Tranter [27] korrekciót javasol az empirikus fittségi szintek és fáradékonyság alapján, melyet az alapján becsült, hogy a túrázó mennyi idő alatt tud 1 mérföldön 1/2 mérföld emelkedőt megtenni. Javasolja továbbá, hogy rossz talajon vagy nehezebb időjárási körülmények esetén a fittségi skála eredeti értékéhez képest 1-2 szintet csökkentve kaphatunk pontosabb becslést.

- Scarf arra hívja fel a figyelmet, hogy a korrekció nem csak nagyobb meredekségű emelkedő esetén használandó, de meredek lejtőn is, mely szintén igénybe veszi a túrázó képességeit [38].
- Tobler a gyaloglás sebességét exponenciális függvénnyel becsülte az út meredekségének függvényében [28]. Ennek maximuma kb. 6km/h kis meredekségű lejtőn, míg a sebesség 0-hoz közelít +-60 fok esetén, tehát extrém meredekségű emelkedőn vagy lejtőn.

A különféle becsléseket az 2. *ábrán* foglaljuk össze, ahol a becsült sebességet láthatjuk a meredekség (fokban mérve) függvényében.

2. ábra: Becslési eljárások összevetése



Látható, hogy Tobler eredményei, valamint Naismith-Langmuir görbéje pozitív értékek esetén egybe esik. Sík terepen mindkét módszer nagyjából 5km/h sebességet becsül, bár Tobler egy nagyon enyhe (-2.86°) lejtőn 6km/h maximum sebességgel számol, míg Langmuirnél -12° foknál éri el a maximum sebességet (7.5km/h), és ott - nehezen védhetően - hirtelen letörik.

Mindazokon túl, hogy a fenti becslések nem, vagy csak nehezen tudják számításba venni a terepviszonyokat, a legritkább esetben veszik figyelembe az időjárási körülményeket, a túrafelszerelés össztömegét, a megtett úttal fokozódó fáradást, a túrázó általános fittségét (kivéve Tranter) és a pillanatnyi/napi állapotát (vagyis azt a teljesítményt, amit magához mérten pillanatnyilag nyújtani képes), ezzel a kritikával élt például Aitken [27], Scarf [152], vagy Fritz és Carver [153]. Mivel ezekre a vizsgált adatsorokban nincs információnk, más módszerrel próbáljuk majd ezeket a becslés során figyelembe venni. Fontos megemlíteni, hogy jelen dolgozat nem terjed

ki a futók, biciklizők vagy tájfutók menetidőbecslésére, mivel azok merőben más tényezőktől is függenek, mint a túrázók mozgása, így kellő adat hiányában azok külön elemzésétől eltekintünk.

A személyre szabott túra menetidő becslésnek igen szűkös a szakirodalma. Pitman et al. [30] polinomiális becslőfüggvénnyel közelíti a túranaplók menetidejét szakaszonként olyan változókkal, mint az adott pontig megtett út hossza (%), adott szakaszon az emelkedő mértéke, az adott pontig megtett összes emelkedés és ereszkedés (%). Ezt tovább finomítják a túrázó saját teljesítményét tükröző faktorral. A szerzők egy későbbi munkájukban kísérletet tesznek a fenti eredmények javítására biciklis túraadatokon [31], ahol a legközelebbi szomszéd módszerével próbálják becslésüket finomítani. Eredményeiket összevetik a regressziós modell által becsült eredményekkel, azonban a becslések mind alul maradnak a korábbi eredményekhez képest, egyrészt talán azért, mert a kerékpározók mozgásának modellezése komplexebb feladat (a fizikai paraméterekre vonatkozó adatok nem álltak rendelkezésre), mint a túrázóké, másrészt a rendelkezésre álló túranaplók szűkössége miatt (a feldolgozott 3 túraszakaszon összesen 49 adatsor szerepelt). Ez utóbbi munkájuk jelentősége sokkal inkább abban rejlik, hogy már közösségi adatokat is alapul vevő ajánló rendszerek (Recommendation Systems) irányába mutat a túrázás területén, mely például a városnézést tervező applikációk terén korábban megjelent. A turisztikai témájú ajánló rendszerek jó összefoglalóját adja Ricci [32]. Tumas és Ricci [33] már úgy tervez útvonalat városban, hogy a becsült érkezési időt a tömegközlekedési eszközök menetrendjéhez hangolja. Az ajánló rendszerek a jövőben tartalom alapú szűréssel (content based filtering) és érkezési időpontokra vonatkozó személyre szabott becslésekből kalkulált elérhetőségi korlátokkal határozzák meg a következő lépésben a felhasználó számára ajánlott látnivalókat, ahogy azt Höpken et al. már 2010-es cikkében előre látta [136].

2.3.2. Sport és rekreáció

Hrncir et al. [173] cikkükben biciklisek számára készítettek útvonaltervező algoritmust, mely figyelembe veszi a menetidőt, kerüli a túlzott emelkedőket, és általában figyelembe veszi a biciklis komfortérzetét. Menetidő becslésre a Naismith-szabály multiplikátorokkal módosított változatát használja, melyet költségfüggvényként használ, míg az útvonaltervezésre A* algoritmust alkalmaz. Pribul és Price [174] tájfutók teljesítményét vizsgálták mindkét nemet és több korcsoportot összevetve 119 versenyzőből álló mintán. A t-teszt eredményei szerint nem tapasztalható szignifikáns eltérés a profi és nem profi futók stratégiája között, így az eredményeik közötti különbség inkább az erőnléti különbségekkel magyarázható. Szintén tájfutók eredményeit vizsgálva Scarf [152] a Naismith-szabályban túrázókra megfogalmazott horizontális és vertikális távolságok

megtételéhez szükséges idő ekvivalenciáját akarta futókra is kiterjeszteni (Naismith-nél ez α=7,92, vagyis 1km emelkedő megtételéhez szükséges idő megegyezik 7,92 km sík terepen történő gyaloglás idejével). Modelljében számolva a versenyzők fáradékonyságával is, log-lineáris modellt illeszt a tájfutók teljesítményét leíró adatsorra, és OLS becsléssel α=8 értéket kapja a férfi versenyzőkre és α=9,5-öt a nőkre. Norman [175] becslése során ehhez képest α=4,4 adódott, míg Kay [34] szintén futók eredményeit vizsgálva α=11,7-es értéket kapott OLS becsléssel, ahol a sebességet a meredekség 4-edfokú polinomjával magyarázta, és figyelembe vette a teljes út hosszát is. A Naismith-féle ekvivalencia paraméter értékének ilyen nagy eltéréseit Norman és Scarf is annak tulajdonítja, hogy a vizsgált utak körülményei nagyjából homogének a vizsgált mintán belül, azonban jelentősen eltérhetnek (pl. a talaj minősége) különböző kutatók mintaadatai között. Minetti [176] megállapítja, hogy $|m| > 0{,}15$ meredekség értékek esetén nem alkalmazható ugyanaz a modell, mint viszonylag sík terepen, így erre a két szakaszra külön illesztést javasol. Minetti et al. [177] kismintán vizsgálta a tájfutók elméleti sebességhatárait (olyan fiziológiai korlátokra alapozva, mint pl. az oxigénfelvétel), és bár meredek emelkedőkön a megfigyelt sebességhatárok jól közelítették a feltevéseit, meredek lejtőn a megfigyelések alatta maradtak a várakozásoknak. Ezt azzal magyarázza, hogy túl meredek lejtőkön az életösztön tartja vissza a futókat a nagyobb sebességtől. A legkisebb költségű utak keresését néhányan nem idő minimalizálásával oldották meg, inkább a felhasznált energiát igyekeztek minimalizálni. Rees [178] cikkében például Dijkstra algoritmussal kereste a leginkább “energiahatékony” útvonalat. Változatos felszínű terepen történő két pont közötti útvonal optimalizálásra Kay ad egy Euler-Lagrange-egyenleten alapuló variációszámítási megoldást [29]. Dacára annak, hogy olyan egyszerűsítési feltétellel élt, hogy a sportoló sebessége egyedül az út gradiens vektorától függ, eredményeit drámaian befolyásolja, hogy a gyaloglás vagy futás ütemét becslő függvénye nem közelíti eléggé a tesztadatokat. Hasonlóan optimális útvonalat keres Verries [179] is cikkében, de ő Kay-jel ellentétben nem időt minimalizál, hanem ráfordított energiát, és optimális irányítási technikával számolja a trajektóriát.

2.3.3. Közgazdaságtan

A közgazdasági modellek egy jelentős részénél játszik szerepet a távolság vagy idő, mint költségtényező. Ennek jó példája a piacszerkezetekből ismert Hotelling-modell [125], mely az ellátóhelyek optimális elhelyezését írja le. Ennek gyakorlati alkalmazása során használt költségfüggvényekben a becsült menetidő alapján kalkulálnak, melyre jó példa Steif lakáspiac modellje [122].

Igen fontos szerepet tölt be a menetidőbecslés a logisztika területén is, Asdemir et al. [267] például élelmiszerboltok házhozszállítási szolgáltatásainak árazását modellezi Markov-döntési folyamat alapú eljárással, mely során a kapacitáskorlátok, és a házhozszállítási időablakok mellett figyelembe veszik a szállítási időt is. Minden új megrendelésnél dinamikusan változnak a házhozszállítási árak úgy, hogy a hátralevő foglalási horizonton állandó maradjon a bolt várható haszna függetlenül attól, milyen házhozszállítási opciót választ a vásárló. Yang et al. [268] a rendelkezésre álló időablakokat is egyenként dinamikusan árazzák attól függően, mennyi az adott útszakaszon a várható (forgalomtól függő) menetidő és mennyi a teherautók szabad kapacitása az időszakban.

Egy speciális megközelítése az útvonaltervező és menetidőbecslő eljárásoknak a "közösségi kenyérmorzsáknak" (social breadcrumbs) nevezett információk alapján történő túraútvonal építés. De Choudhury et al. [269] az interneten (Facebook, Flickr, stb.) megosztott fotók és egyéb bejegyzések gyűjtése és szisztematikus válogatása alapján becsli a menetidőket a tervezett útvonalakon, összeegyeztetve a felhasználó előre kinyilvánított preferenciáival. Popescu és Grefenstette [252] korábbi munkája alapján lehetőség van az egyes helyszínek látogatási idejének, illetve a köztük megtett út menetidejének becslésére is úgy, hogy a feldolgozott fotók időbélyei (timestamp) alapján kalkulálnak. Letchner et al. [36] helyi lakosok autós GPS adatai alapján optimálishoz közelebbi útvonalat tudtak javasolni az átutazóknak, mint amit egyéb útvonaltervező alkalmazások, mert ők egy eddig fel nem használt információt építettek a tervezésbe: a tapasztalatot.

2.3.4. Környezetvédelem

Az utaktól távol eső területek elérési idejét talán először Fritz és Carver [153] modellezte. Ők teljes Skócia területére elkészített hőtérképük segítségével kimutatták a forgalomtól távol eső, nehezen megközelíthető területeket. Munkájuk során Dijkstra-algoritmust alkalmaztak a legrövidebb út meghatározására, és a Naismith-szabály alapján kalkulálták a menetidőket, figyelembe véve az esetleges akadályokat és a talajtípust is. Ezt alkalmazzák Yang et al. [154] cikkükben, ahol a nemzeti parkok veszélyeztett területeit tárják fel menetidőbecslési eljárással azt vizsgálva, mennyire frekventáltak az egyes, utaktól távol eső területek. Feltevésük szerint a környezet terheltsége egyenesen arányosa nő a terület megközelíthetőségével, így a veszélyeztetett területek folyamatos ellenőrzése különösen fontos. Li et al. [155] azt találta, hogy minél több körút található a kijelölt ösvények között, és minél inkább összefüggőek az utak, annál kevésbé terhelik a turisták a környezetet (például azzal, hogy letaposssák az aljnövényzetet). Lynn és Brown [156] már sokkal tudatosabb tervezés alapjait teszi le a természetvédelmi területek vezetői számára, és olyan

úthálózat kialakítását javasolják, mely minimalizálja a terület terheltségét, ugyanakkor szem előtt tartja a látogatók érdekeit is.

2.3.5.Régészet

Herzog [157] kimerítően tárgyalja az alkalmazható legkisebb költségű hálózatok (Least-cost Networks) modelljeit egy észak-Rajna-Vesztfáliai területre alkalmazva. Ismerteti, hogy figyelembe véve a középkori terepviszonyokat, a modellek által kalkulált útvonalak mennyiben egyeznek a történelmileg ismert, kialakult utakkal. Másik gyakori alkalmazása a gyűjtőterületek (Site catchment) modellezése, vagyis egy adott pontból bizonyos költségkereten (pl. idő, energia, stb.) belül elérhető terület. Kienlin et al. [158] például két késő bronzkori település 15 percen belüli gyűjtőterületét becsülték Tobler-görbe alapján kalkulált időkkel. Ullah és Bergin [159] ágens alapú modellel szimulálták spanyol falvak környezetre gyakorolt hatását. A legkisebb költségű utak (Least-cost Paths) kalkulálása során történő felhasználásnak az egyik jó példája Verhagen és Jeneson [160] munkája, akik a limburgi régióban igyekeztek rekonstruálni az ókori római via Belgica utat dombos területen. A témában megjelent tanulmányok közös gyengesége, hogy nem számolnak terhelésből származó lassulással (kivéve Rademaker et al [161]), holott ez különösen fontos lenne ott, ahol vizet vagy élelmiszert szállítanak, és csak a legritkább esetben veszik figyelembe alternatívaként vízi utakat. A menetidőbecslésen alapuló régészeti kutatások részletes összefoglalóját találjuk Herzog [162] cikkében.

2.3.6. Kitelepítés tervezés (Emergency Evacuation Modeling) és életmentés

Wood és Schmidtlein [163] Washington állam lakosságán szimulálták egy esetleges szökőár során alkalmazandó kitelepítési stratégiák eredményességét. Rámutattak, hogy az eredmények igen érzékenyek egyrészt az alkalmazott gyalogos menetidőket becslő függvényekre, másrészt a populáció összetételére, így különösen fontos, hogyan szegmentálják a teljes lakosságot mozgékonyságuk szerint.

Elveszett turisták keresése esetén kiemelkedően fontos annak a területnek a minél pontosabb behatárolása, ahova a csoport eljuthatott, hiszen minél kisebb területet kell átkutatni, annál könnyebb gyorsabban juthatnak eredményre. Magyari-Sáska és Dombay [40] Tobler-görbén alapuló menetidő becslést használtak a menetidő egy alsó becslésére, hogy meghatározzák azt a maximális területet, ahol egy elveszett turistát keresni kell. Magyari-Sáska [172] cikkében ennek továbbgondolásaként Dijkstra-algoritmust használ az útvonal tervezésre, és igyekszik szűkíteni a keresési területet.

2.3.7.Egészségügy (különös tekintettel a fejlődő országokra)

Gething et al. [164] a ghánai egészségügyi ellátás helyzetét vizsgálva azt találta, hogy a nők 34%-a él a klinikailag kritikusnak tartott 2 órás tűréshatáron kívül a legközelebbi ellátó központtól. Menetidőbecslési eljárások segítségével gyökeresen más szempontokat tudnak az egészségügyi infrastruktúra stratégiai tervezése során figyelembe venni. Noor et al. [165] tanulmányukban megmutatták, hogy a kenyai kormány malária, tuberkulózis és HIV elleni védekezésre telepített egészségügyi központjainak lakosság általi elérhetősége jóval túlbecsült (a lakosság 63% van 1 órányi távolságra, szemben a jelentésekben szereplő 82%-kal), így további központok létesítésére tesznek javaslatot a modell eredményeire alapozva.

2.3.8. Légiirányítás és reptéri optimalizálás

A légiirányítás alapfeladata, hogy a légiforgalmi igényeket és a reptéri kapacitásokat összeegyeztesse, miközben minimalizálja a késéseket. Carr et al. [166] olyan algoritmus megalkotását tűzte ki célul, mely a korábban használt érkezési sorrend alapú kiszolgálási elv (First-come-first-served) helyett egyéb légiforgalom-irányítási prioritásokat is figyelembe vesz. Menetidőbecslésen alapuló forgalmi modelljükkel (Estimated Time of Arrival, röviden ETA) jelentősen csökkentették az átlagos késést szinte minden légiforgalmi szegmensben.

Reptéri kapu hozzárendelési feladat (Airport Gate Assignment Problem, röviden AGAP) néven ismert a nemzetözi szakirodalomban a járatok kapukhoz rendelésének feladata, ahol a cél az utasok kényelmének biztosítása a reptéri operáció hatékonyságának magas szinten tartása mellett. Bolat [270] például kevert egészértékű lineáris programozási feladatként formalizálta a problémát, ahol a kapuk holtidejének tartományát minimalizálta. Maharjan és Matis [271] több-árucikkes bináris hálózati folyamként modellezi a problémát, és a gyalogos összes megtett útját minimalizálja a gépek üzemanyagfogyasztása mellett. A reptéri kapuk optimalizálásáról bővebben Bouras et al. [272] összefoglaló cikkében olvashatunk.

2.3.9. Lift ütemezés (Elevator scheduling)

Az egyre magasabb felhőkarcolók építése a lifteket tervező mérnököket is egyre nagyobb kihívások elé állítják. A lakók és látogatók zökkenőmentes szállítása érdekében a pontos menetidőbecslésen túl egy sor egyéb körülményt kell figyelembe venni a liftek priorizálásnál. Ennek jó példája Rong et al. [167], ahol a szokásos menetidőbecslő eljárásokat kiegészítették a várható megállások időtartamával, és azok átlagos várakozási időre gyakorolt hatását figyelembe véve engedik vagy blokkolják a további megállásokat. Xiong et al. [273] dinamikus programozási technikával optimalizálja a több liftből álló rendszert. Additív modellt alkalmazva egyedi liftek optimalizálására vezeti vissza problémát.

2.3.10. Közlekedés

Az intelligens közlekedési rendszerek (Intelligent Transport Systems) már jó ideje mindennapi életünk részét képezik. Céljuk az aktuális forgalmi helyzethez dinamikusan alkalmazkodó automatikus forgalomirányítás kialakítása és üzemeltetése. Sándor és Csiszár cikkükben [111] egy intelligens parkoló menedzsment modellt írnak le, mely dinamikusan képes kezelni a változó körülményeket és a felhasználók igényeit. Jó példa továbbá a maximális haladási sebesség dinamikus szabályozása, vagy alagutakban és hidakon az egyirányba haladó sávok számának dinamikus változtatása ugyanúgy, mint a forgalmi lámpák forgalomtól függő szabályozása (Al-Khateeb et al. [168]). A közlekedési lámpáknál történő sorbanállást hagyományosan input-output szemléletben modellezték, míg Lighthill és Whitham [275], valamint Richards [276] egymástól függetlenül megalkották a forgalmi lökéshullám elméletüket (Lighthill–Whitham–Richards shockwave theory), melyben klasszifikálják a forgalom szereplőit a forgalmi állapotra gyakorolt hatásuk alapján, és az interakcióik alapján jelzik előre a sorbanállás várható idejét. A modell egy továbbfejlesztését láthatjuk Logghe és Immers cikkében [277], ahol a különféle csoportok között non-kooperatív interakciókat feltételezve pontosabb becslésekhez jutottak a korábbi eredményeknél.

A tömegközlekedési eszközök menetrendjének betartása az utazók elégedettségének alapfeltétele. Az esetleges késések minél pontosabb előrejelzése, valamint az azokról történő tájékoztatás szintén javíthatják a felhasználói élményt, ahogy ezt Watkins et al. [278] is megfogalmazzák tanulmányukban. Zhou et al. [169] a buszok érkezésének becslését javította a buszon tartózkodó utasok mobil eszközének GPS adataival, hogy a buszra várakozókat minél pontosabban tudják tájékoztatni az érkezésekről, valamint a várható késésekről. Vu és Khan [280] munkájában a valós idejű GPS adatok mellett utasszámláló rendszerek, valamint historikus adatokon végzett mintafelismerés (pattern recognition) segítségével pontosítják az előrejelzéseket. Stover és McCormack [279] rámutatnak arra, hogy a menetidők előrejelzésének pontosságát jelentősen lehet javítani, ha az időjárási körülményeket is figyelembe vesszük. Vizsgálataik szerint az eső a legerősebb befolyásoló tényező, és a téli időszakban a legnagyobb annak menetidőre gyakorolt hatása. Sándor és Csiszár [281] cikkében a menetidőbecslés pontosságának javítását a historikus adatok felhasználásával érték el, kihasználva azt az egyszerű megfontolást, hogy az utasforgalmi létesítmények és az aktuális környezeti paraméterek kategorizálhatóak. Cathey és Dailey [273] a tranzit érkezési és indulási időpontokat jeleznek előre modelljükkel, mely a járművek GPS adatai alapján becsüli a menetidőket Kálmán-filter segítségével. A közlekedésben használt vezeték nélküli

kommunikációs eszközök adta lehetőségekről bővebben olvashatunk Rappaport et al. [274] cikkében, mely kitér azok közlekedésbiztonságban betöltött szerepére, valamint a technikai megvalósítás nehézségeire is.

2.3.11. Infrastruktúra tervezés

Közutak, alagutak és vezetékek tervezésénél szintén kézenfekvő a legkisebb költségű utak (Least-cost Paths) kalkulálására használt algoritmusok alkalmazása. Yu et al. [170] ST-algoritmusa (Smart Terrain algorithm) az A* algoritmuson alapszik, melyet kiegészítettek olyan gyakorlati megfontolásokkal, mint a hidak és alagutak figyelembevétele az autóút tervezésénél. Bár algoritmusuk ezen kiegészítő tereptárgyakról még azt feltételezi, hogy vertikális irányban nem mozgunk, ha áthaladunk rajtuk, ezt leszámítva is nagy előrelépést jelentett munkájuk a gyakorlati probléma megoldásában. Bagli et al. [171] villanyvezetékek tervezése során továbbfejlesztett algoritmusa már számos más tényezőt is figyelembe vesz az útvonal tervezésénél, köztük legfontosabb a környezeti hatások minimalizálása.

Úgy vélem, ezzel a rövid, és koránt sem teljes, összefoglalóval sikerült betekintést nyújtani a menetidő becslési eljáráson alapuló alkalmazások széles spektrumába, mely jól példázza a téma gyakorlati fontosságát, és néhol rávilágít annak hiányosságaira is. Látható, hogy a GPS eszközök - valamint a mobileszközökbe épített egyébb szenzorok, pl. gyorsulásmérő és interciális navigáción alapuló eszközök - elterjedésével, illetve a mért adatok tömeges feldolgozásával egyre pontosabb előrejelzést adhatunk arra vonatkozóan is, kit mikor és hol találhatunk, mely egyszerre bravúros és ijesztő. Ugyanakkor az új technológiai vívmányok felhasználásával olyan integrált rendszereket hozhatunk létre, melyek nagyban hozzájárulnak a közlekedéstervezés pontosítására, lehetőséget nyújtva az előálló szokatlan helyzetekre történő azonnali reakcióra, nagyobb kényelemre és hatékonyságra.

2.4. Földfelszín modell

2.4.1. Földfelszín modellekről általában

A korábbi szakaszban ismertetett menetidő becslő eljárások egyik közös problémája, hogy adatok hiányában igen nehéz jól modellezni a sportolók sebességét. Bár a technika lehetővé teszi, hogy

GPS eszközökkel nyomon kövessük a felhasználók mozgását, annak pontossága és megbízhatósága koránt sem megkérdőjelezhetetlen. A horizontális irányú mozgások viszonylag pontos követésére tett kísérletet egy későbbi alfejezetben mutatjuk be. Most a magasságadatokra koncentrálunk.

A szélességi és hosszúsági adatokhoz tartozó magasságok megállapítása nem magától értetődő, mivel sajnos a GPS eszközök által adott adatok pontossága finoman szólva is megkérdőjelezhető. Saját tapasztalat alapján mondhatom, hogy 3 különböző mobil helymeghatározó eszközzel mért magasság adatok között a Dobogókőn nagyjából 120 méteres eltérés volt tapasztalható a legkisebb és legnagyobb mért magasságérték között adott ponton. Ez ráadásul nem konzisztens, tehát más helyszínen mérve ezek a különbségek változnak. A helyzetet tovább nehezíti, hogy felhős időben még pontatlanabb értékeket mutatnak ezek az eszközök. Mindent egybevetve a GPS által rögzített túranaplók (tracklogok) magasság adatai nem használhatóak a menetidőbecslés során, mert az egyes útszakaszok meredekségének helyes kalkulálásához elegendően pontos magasság adatokra van szükségünk. A földfelszínt e célból a digitális magasság modellekkel (*Digital Elevation Model*, röviden DEM) szokták közelíteni. Bár rengeteg létezik, közös tulajdonságuk, hogy bizonyos elvek mentén egy jól körülhatárolt területhez (szélességi és hosszúsági adatokkal definiáltan) egy magasságértéket rendel. A magasság modellek áttekintésére alkalmas az alábbi összefoglaló [181].

Az általam használt magasságadatok a NASA által nemrégiben közzétett adatokon alapszanak [182]. Ez lényegében 30×30 méteres négyzetekre osztja a földfelszínt, és ezekhez rendel magasság értékeket. Létezik az USA egyes területeire 10 és 15 méteres finomságú felosztás is, ám ez a teljes világra nem elérhető, és túranaplóink Magyarország területén készültek.

Amennyiben a túranaplók néhány másodperces frekvenciával tartalmaznak helyzeti adatokat, akkor két ilyen mefigyelési pont könnyen kerülhet azonos 30×30 méteres négyzetekbe, mely azt eredményezi, hogy az adott szakasz meredeksége 0, holott valójában ez a legritkább esetben fordul elő, és javarészt emelkedők és ereszkedők során haladunk végig. A másik tipikus hiba akkor áll elő, mikor egy 30×30 méteres négyzet szélén haladva egyik pillanatról a másikra látszólag hatalmasat ugrik a magasság érték, amint egy másik négyzetre léptünk át. Mivel hasonló eredményt kapunk a túraszakaszaink döntő többségére, így a NASA magasság modellje nem alkalmas önmagában arra, hogy magasság adatokat rendeljünk a horizontális megfigyeléseinkhez. A földfelszín pontosabb közelítésére több technika létezik, akár a mozgóátlagok képzése, akár a lineáris regressziós modellek alkalmazása. A szóbajövő technikák és azok eredményességének összehasonlítását foglalja össze cikkében Skidmore [183]. A bemutatott 6 modell mindegyike szignifikánsan jobb eredményt ad, mint a kiindulásként alkalmazott DEM, de közülük is a két lineáris regresszión alapuló eljárás a legeredményesebb. Ezek komoly számításigénye miatt mi két másik, saját eljárást

mutatunk be. Mint látni fogjuk, mindkét megoldás jó közelítését adja a valós földfelszínnek, és szignifikánsan jobb eredményt ad, mint a NASA magasság modellje.

2.4.2. A háromszög-modell

Számításaink során az alábbi feltevésekkel élünk:

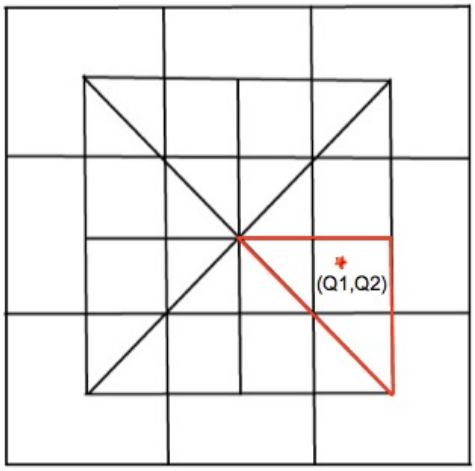
- Minden négyzetnek ismert a 4 csúcsának helyzete (szélességi és hosszúsági koordináták)
- Minden négyzethez hozzá van rendelve 1db magasság adat.

A NASA adatoknak köszönhetően ezen feltevéseink teljesülnek. Vegyük az adott ország minimális téglalap lefedését, mely 30×30 méteres négyzetekre van felosztva, és minden négyzethez tartozik egy magasság érték is, így négyzeteink valójában eltérő magasságú, négyzet alapú hasábok.

Földfelszín modellünk lényege, hogy a különböző magasságú, négyzet alapú hasábok helyett közelítsük a felületet a hasábok tetején elhelyezkedő szomszédos négyzetek középpontjai által kifeszített háromszögek összességével. Egy adott hasáb 8 másikkal szomszédos. Kössük össze minden hasáb fedőlapjának középpontját a vele szomszédos 8 másik fedő négyzet középpontjával. Így minden négyzet-hármas középpontja meghatároz egy háromszöget. Ezek vízszintes síkra vonatkozó merőleges vetülete minden esetben egy derékszögű, egyenlőszárú háromszög (lásd a 3. és *5. ábrán*).

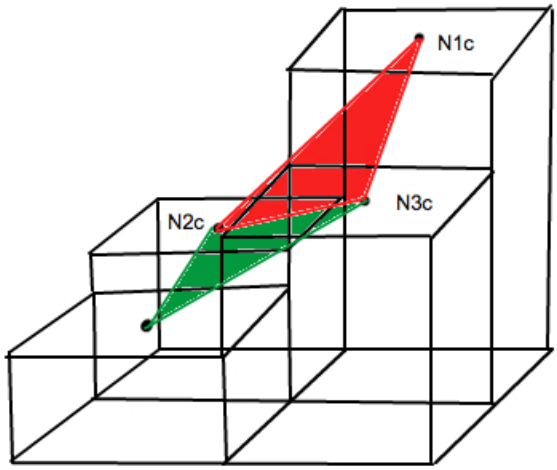
Számozzuk be fentről lefelé sorban haladva balról jobbra a hasábok alapját képező négyzeteket, és kapják ezt i indexként. A négyzetek csúcspontjait jelölje rendre: ($N_{i,1}$, $N_{i,2}$, $N_{i,3}$, $N_{i,4}$), ahol $N_{i,j}$ =($x_{i,j}$, $y_{i,j}$), így a négyzet középpontjának koordinátái: $N_{i,c}$ =(($x_{i,1}$+$x_{i,2}$+$x_{i,3}$+$x_{i,4}$)/4 , ($y_{i,1}$+$y_{i,2}$+$y_{i,3}$+$y_{i,4}$)/4). Feltételezzük, hogy a négyzetek középpontja éppen a négyzethez rendelt m_i magasságon helyezkedik el.

3. ábra: Földfelszín raszter



Ekkor egy adott $Q=(Q_1,Q_2,Q_3)$ pont Q_3 koordinátája kiszámítható, ha megkeressük azt a háromszöget, amelynek vetülete tartalmazza (Q_1,Q_2) -t (lásd a 3. *ábrán*), és meghatározzuk 3 dimenzióban a háromszög síkjának egyenletét. Legyenek annak a háromszögnek a csúcspontjai, melynek vetülete éppen az a derékszögű, egyenlőszárú háromszög a síkon, melyben (Q_1,Q_2) pont esik: $N_{1,c}=(p_{1,1}, p_{1,2}, p_{1,3})$, $N_{2,c}=(p_{2,1}, p_{2,2}, p_{2,3})$, $N_{3,c}=(p_{3,1}, p_{3,2}, p_{3,3})$, (lásd a 4. *ábrán*).

4. ábra: Háromszög-modell



Legyen az $(N_{1,c}, N_{2,c}, N_{3,c})$ háromszög síkjának egyenlete $Ax+By+Cz+D=0$

Behelyettesítve a 3 pont koordinátáit, és megoldva az egyenletrendszert kapjuk az alábbi paraméterértékeket:

$A= B((p_{3,2}-p_{2,2})/(p_{3,3}-p_{2,2})-(p_{1,2}-p_{2,2})/(p_{1,3}-p_{2,3})) / ((p_{1,1}-p_{2,1})/(p_{1,3}-p_{2,3})-(p_{3,1}-p_{2,1})/(p_{3,3}-p_{2,3}))=BK$

$C=B(K(p_{1,1}-p_{2,1})+p_{1,2}-p_{2,2}) / (p_{2,3}-p_{1,3})=BL$

$D= - B(Kp_{3,1}-p_{3,2}+Lp_{3,3})$

A megoldást visszahelyettesitve az egyenletbe kapjuk a 3 pont koordinátái által meghatározott sík egyenletét, mely tartalmazza a 3 pont által kifeszített háromszoget.

Most számoljuk $Q=(Q_1,Q_2,Q_3)$ pont magasságkoordinátáját, Q_3-t, ezert (Q_1,Q_2)-t helyettesítve a sík egyenletébe adódik Q_3. (Megjegyzés: a párhuzamos szelők tételén alapuló megoldással felírható egyszerűbb formában is a Q pont az $(N_{1,c}, N_{2,c}, N_{3,c})$ csúcsok helyvektorainak súlyozott átlagaként.

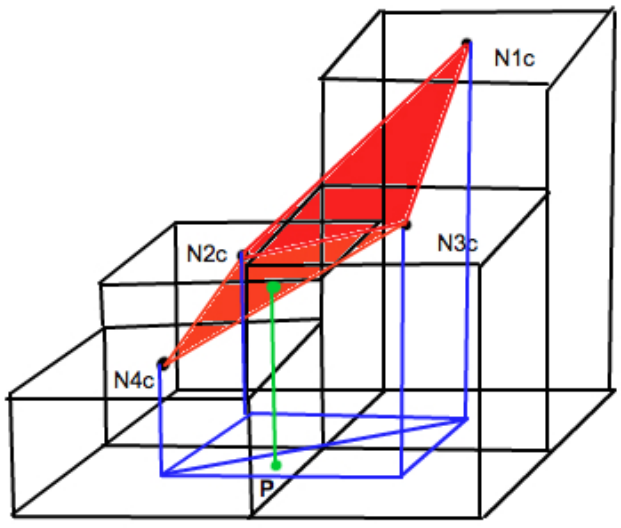
Mivel ennek számításigénye kisebb, így R-ben implementált földfelszín modellben ez került alkalmazásra.)

Kellően sűrű túranapló esetén ez igen jó közelítését adja a két rögzített pont közötti magasságkülönbségeknek. Bár nem tekinthetjük kőbe vésett igazságnak, hogy a Google földfelszínt közelítő alkalmazása által nyújtott magasság adatok helyesek, mi most ezt tekintjük viszonyítási alapnak. Az általunk kalkulált magasságértékeket a Google értékeivel vetettük össze a világ teljes felszínén 10000 mintapontot felvéve. Eljárásunkkal átlagosan 13cm-rel kaptunk magasabb értékeket, mint a Google magasságértékei. Ez a GPS eszközök által nyújtott helyenkénti 120 méteres eltérésekhez képest elenyésző. A modell által adott vizualizációt a *7. ábrán* láthatjuk.

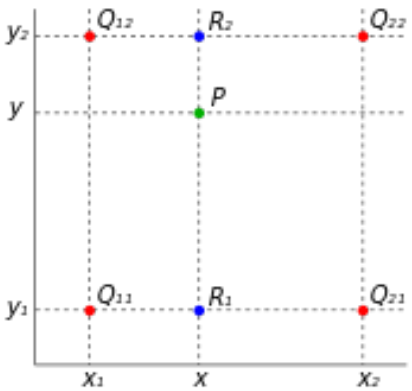
2.4.3. A négyszög-modell

A másik általunk alkalmazott földfelszín közelítés a bilineáris interpoláció elvén alapszik [184], ahol a 4 egymás melleti négyzet alapú hasáb fedlapjainak középpontjaira számolt átlagokkal közelítjük a valódi felszínt (lásd a 5. *ábrán*). Ekkor a felszínen adott P pont magasság koordinátáját úgy számoljuk, hogy a négyzetek felszínre eső merőleges vetületeivel vett téglalapok területének arányában súlyozzuk a megfelelő középpontokhoz tartozó magasság értékeket (lásd a 6. *ábrán*). Fontos megemlíteni, hogy az ($N_{1,c}$,$N_{2,c}$,$N_{3,c}$,$N_{4,c}$) csúcsok a legritkább esetben esnek egy síkba, így a földfelszínt sem az általuk kifeszített "négyszögekkel" közelítjük, hanem a helyvektoraik konvex kombinációjaként előálló vektorok összességével. Ennek értelmében a számított pont is csak véletlenül eshet ($N_{1,c}$,$N_{2,c}$,$N_{3,c}$,$N_{4,c}$) csúcsok közül bármelyik három által kifeszített síkba. A pontos számítást a *B mellékletben* ismertetjük.

5. ábra: Négyszög-modell

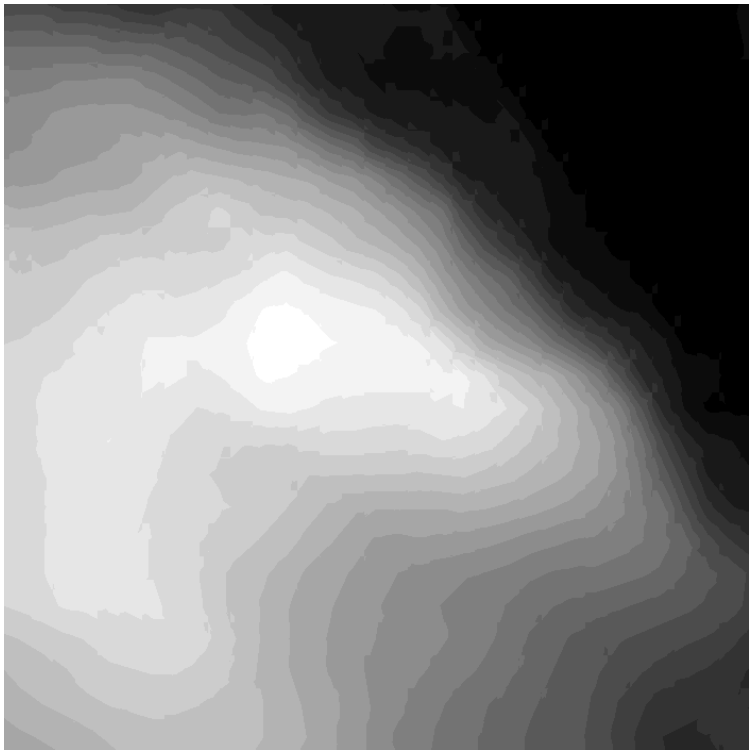


6. ábra: Bilineáris interpoláció

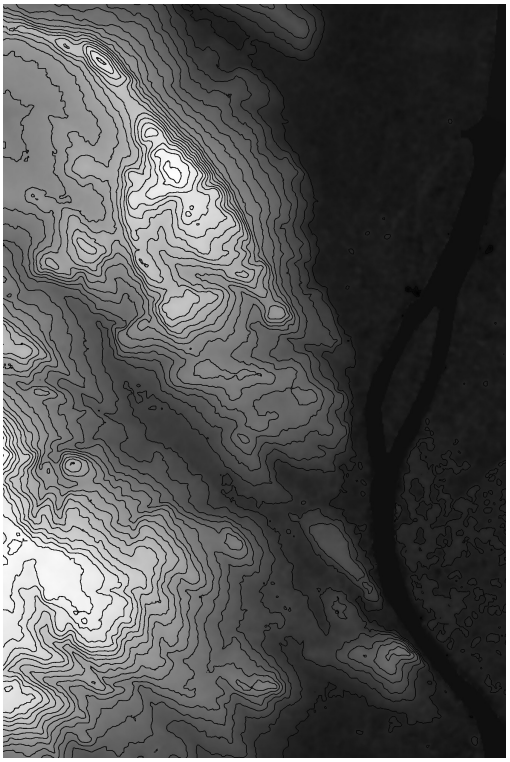


Az így számított magasságadatokat ismét összevetettük a Google alkalmazás által kalkulált értékekkel a korábbihoz hasonló módon, és kevesebb, mint 1 cm-en belüli eltérést tapasztaltunk, mely sejteti, hogy a Google is hasonló elvek mentén és hasonló alapadatokból (NASA, [182]) készítette földfelszín modelljét, mint a mi "négyszög-modellünk", ám erről további információ hiányában nem mondhatunk biztosat. A modell által adott vizualizációt a *8. ábrán* láthatjuk.

7. ábra: Gellért-hegy a háromszög-modellel



8. ábra:Buda a négyszög-modellel



2.5. A menetidőbecslő eljárások

Az alábbi alfejezetben kerül bemutatásra a rendelkezésünkre álló adathalmaz, valamint az azon alkalmazott becslési eljárások vizsgálata, ahol már felhasználjuk a korábbiakban ismeretetett DEM modellünk eredményeit.

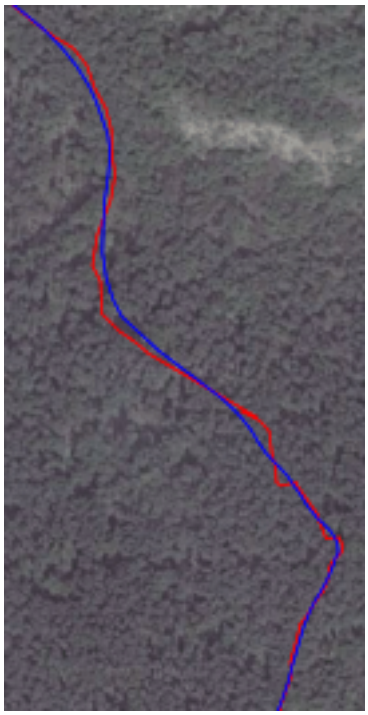
2.5.1. Az adatok és azok tisztítása

A tanulmányban felhasznált nyers adatokat a turautak.hu oldal működtetőivel való együttműködés keretein belül vált hozzáférhetővé. A rendelkezésünkre bocsátott mintegy 35.000 túranapló az ország teljes területét lefedi. Mivel a túrázók gyakran nem csak gyalogos, de biciklis, vagy akár

autós szakaszaikat is feltöltötték, így az adatokat szűrni voltunk kénytelenek. Az alábbi elvek mentén távolítottuk el a túranaplók egy részét:

- a túl rövid túrákat (ahol nem volt legalább 120 megfigyelt szakasz, azaz 40 perces, egybefüggő túra)
- a nem összefüggő, de összefűzött túrákat (ahol a túranapló tulajdonosa több, egymástól időben vagy térben elváló túrákat fűzött össze)
- azon túrákat, melyek 8m/s-nál magasabb sebességű szakaszokból több mint 5-öt tartalmaztak (hiszen itt vélhetően biciklis, vagy autós túrákról lehet szó
- a 0.5m/s-nál alacsonyabb átlag sebességű túrákat, mert ott vélhetően inkább sétáról lehetett szó
- a 3,5m/s-nál nagyobb átlagsebességű szakaszokat, mert ezek inkább futók vagy biciklisek túranaplói lehetnek

9. ábra: Túranapló simítása



A megmaradt 2400 túranaplóból eltávolítottuk még a 0,15m/s alatti sebességű szakaszokat (pihenők, stb), valamint a 3,5m/s feletti szakaszokat, mivel azok sebessége már inkább futást, vagy biciklizést jelent. Fontos kiemelni, hogy az általam tanulmányozott szakirodalomban egy elkalommal sem találtam példát ilyen mennyiségű túraadaton végzett vizsgálatra. A Pitman et al. [30] cikkében szereplő 360 túranaplón végzett becslés történt eddig a legnagyobb adathalmazon. A nyers adatok Kalman-filter segítségével lettek simítva, mivel jelenleg ez az általános és széles körben használt eljárás GPS-ből nyert helyzeti adatok kiigazítására. Bár néhol találunk példát arra, hogy legkisebb négyzetek módszerével, vagy mozgó átlaggal simítják a túranaplókat (ami a kanyarokat kifejezetten rosszul kezeli, szisztematikusan túlbecsülve ezzel a sebességet), a Kálmán-filter nagy előnye, hogy nem csak a helyzeti adatokat veszi figyelembe, de a sebesség adatokat is, továbbá a GPS eszköz által szintén tárolt pontossági adatokat, a GDOP-ot (Geometric Dilution of Precision) [259]. Ez GPS-ek pontatlanságát mérő mutató, mely egyrészt függ a mérésben részt vevő műholdak számától (min. 3, jobb esetben 4 darab), illetve azok egymáshoz viszonyított elhelyezkedésétől. A gyakorlatban a polgári célú GPS eszközök pontossága kb. 3 méter. A Kálmán-filter GPS adatokon történő alkalmazásáról bővebben Goh et al. [260] cikkében olvashatunk. A simított túranaplóra láthatunk példát a 9. *ábrán*, ahol a piros útvonal jelzi a nyers GPS adatot, míg a kék a Kálmán-filter által adott megoldást.

A túrázáshoz használt mobil applikációknak, valamint folyamatos GPS kommunikációnak köszönhetően már rendelkezésünkre állnak nagy pontosságú, időbélyeggel ellátott helyzeti adatok, melyek segítségével a teljes túra nyomonkövethető. Mivel az egyes felhasználók eszközei különböző frekvenciával rögzítik az adatokat, így az összehasonlíthatóság érdekében az összes általunk használt túranaplót, sztenderd módon, 20 másodperces frekvenciájú adatpontokból álló sorozattá transzformáltuk. Ezt követően a 2-dimenziós (szélesség- és hosszúság értékekhez) hozzárendeltük a Négyszög-modell alapján számolt magasság értékeket, így már 3-dimenziós adatsorokat kaptunk, melyhez rendelkezésünkre állnak a hozzájuk tartozó időbélyegek is. Fontos megemlíteni, hogy azért esett a választás a négyszög modellre, mert jóval alacsonyabb számítási igénye lehetővé teszi annak nagyobb adathalmazon történő alkalmazását. Az így nyert adataink tehát összefoglalva a következők:

- Helyzeti adatok: a szélességi és hosszúsági adatok rögzítésre kerültek a túranaplóban, melyekhez magassági adatokat rendeltünk az általunk készített földfelszín modell segítségével. A szokásos jelölésekkel a helyzeti pontok sorozata legyen $\mathbf{Q}=(Q_1, Q_2,\ldots, Q_n)$, ahol Q_i a túranapló i-edik pontjának 3-dimenziós koordinátáit jelöli.
- Idő: az egyes szakaszok kiindulási- és végpontjaihoz a GPS által rendelt időbélyegeket használtuk fel a 20 másodperces szakaszokká transzformáláshoz. Jelölje a $\mathbf{Q}=(Q_1, Q_2,\ldots, Q_n)$ lokációk és a hozzá tartozó időpontok sorozatát $\mathbf{t}=(t_1, t_2, \ldots t_n)$.
- Sebesség: az egyes szakaszokra átlagsebességet számolunk, ahol a megtett út a szakasz 3 dimenzióban meghatározott kiindulási- és végpontjainak koordinátáiból számolunk, míg az eltelt idő a kezdeti és végponti időpontok különbsége. Így az (i-1)-edik ponttól az i-edik pontig tartó szakasz átlagsebessége $v_i=\text{dist}(Q_i - Q_{i-1})/(t_i - t_{i-1})$.
- Emelkedő mértéke: két pont közötti átlagos emelkedési szöget tudunk számolni a magassági adatok különbségéből, valamint a szélességi és hosszúsági adatokbol. Itt fontos megemlíteni azt a megfontolást, ami alapján az adatok rögzítésének frekvenciáját meghatároztuk. Amennyiben a frekvencia túl alacsony, akkor az adott útszakaszon szignifikáns emelkedés és ereszkedés is lehetséges egyidejűleg, melynek mi csak a különbségét vesszük figyelembe számításunk során, illetve egyértelműen nem egyenesen közlekedünk két pont között szélességi és hosszúsági dimenziókban sem. Mi tehát az adott szakaszon a kezdeti és végpontot összekötő egyenes szakasszal közelítjük a túra útvonalát, mely nagyobb szakaszokon erősen alulbecsülheti az útszakasz valós hosszát. Ha tovább növeljük az adatpontok frekvenciáját, azzal viszont több zajt viszünk a gps pontatlanságból adódóan az adatokba. Jelen tanulmányban a fentieket szem előtt tartva tehát 20 másodperces frekvenciájú adatokon végeztük elemzéseinket. Az egyes szakaszok

átlagos meredeksége számítható a szakasz kezdeti- és végpontjának koordinátáiból, $Q_i(Q_{i,1}, Q_{i,2},\ldots, Q_{i,3})$ és $Q_{i-1}(Q_{i-1,1}, Q_{i-1,2},\ldots, Q_{i-1,3})$ pontok esetén:

$$\Theta = arctg\left(\frac{Q_{i,3} - Q_{i-1,3}}{\sqrt{(Q_{i,1} - Q_{i-1,1})^2 + (Q_{i,2} - Q_{i-1,2})^2}}\right)$$

Feladatunk tehát becslést adni az egyes szakaszokhoz tartozó v^*_i átlagsebességekre, ebből ugyanis már könnyen számolható a teljes útra vonatkozó menetidő az alábbi módon:

$$\bar{T} = \sum_{i=1}^{n} \frac{dist(Q_i - Q_{i-1})}{v_i^*}$$

A következőkben rátérünk az általunk javasolt becslési eljárások ismertetésére, azonban előtte bemutatjuk a témáját és technikáját tekintve miénkhez legközelebb álló eredményt is.

2.5.2. Többváltozós menetidő becslés

A többváltozós menetidőbecslő eljárásokra igen kevés példát találunk a szakirodalomban. Ezek közül talán a leginkább témánkhoz igazodó munka Pitman et al. [30] cikke, mely személyre szabott menetidő becslő eljárást javasol turisták számára. A cikkben felhasznált adatsor 360 túra túranaplót tartalmaz, melyet Dél-Tirolban rögzítettek 2011. Március és 2012. Március között. Minden út külön túrázóhoz és különböző útvonalhoz tartozik. Pitmanék sztenderd 5 perces szakaszokra transzformálták a nyers adataikat, és piecewise cubic spline algoritmussal simították azt. Az eljárásról bővebben Matthews és Fink könyvében olvashatunk [39]. Túranaplóikat megtisztították továbbá azon szakaszoktól, melyeket outliereknek minősítettek: ha az átlagsebsség az adott szakaszon meghaladta a 4m/s-ot (mert ekkor a túrázó vélhetően nem gyalog közlekedett, esetleg adathiba, vagy a jel időleges elvesztése állhat a kiugró érték hátterében, így a GPS pontatlanságából adódó ugrálás tűnhet nagy sebességű elmozdulásnak), vagy ha kisebb volt az átlagsebessége 2m/s-nál (hisz ekkor nagy valószínűséggel pihenőt iktatott a túrába).

A túrautak menetidejének becslésénél az alábbi tényezőket vehetjük figyelembe vizsgálataink során:

- p - az adott pontig megtett út hossza (km)
- S - a teljes út hossza (km)
- β - adott szakaszon az emelkedő mértéke (fokban)
- a - az adott pontig megtett összes emelkedő a teljes úton tervezett összes emelkedőhöz képest (%)

- d - az adott pontig megtett összes ereszkedés a teljes úton tervezett összes ereszkedéshez képest (%)
- a(30) - az adott pontig megtett összes emelkedő az előző fél órában (km)
- d(30) - az adott pontig megtett összes ereszkedés az előző fél órában (km)
- a(60) - az adott pontig megtett összes emelkedő az előző 1 órában (km)
- d(60) - az adott pontig megtett összes ereszkedés az előző 1 órában (km)

Természetesen figyelembe vehetnénk még például az időjárási körülményeket, vagy az út típusát, ám ezekről jelenleg nincsen elérhető információnk. Fontos lehet azonban az időben közeli emelkedők és eredszkedők figyelembe vétele, mert személyes tapasztalatom szerint annak igen jelentős hatása van a túrázó pillanatnyi teljesítő képességére, ezért teszteljük az a(30), d(30), stb. változók sebességre gyakorolt hatását is.

Pitmanék a tanuló adathalmazon polinomiális közelítést alkalmaztak a fenti változók egy részét szerepeltetve (p, S, a, d és β), és legkisebb négyzetek módszerével becsülték a változók hatványaiból összeállított polinomvektorhoz tartozó **α** együtthatóvektort, hogy az az alábbi alakot öltse:

$$\bar{v} = \sum_{i=0}^{n} \alpha_{pi} p^i + \sum_{j=1}^{m} \alpha_{Sj} S^j + \sum_{k=1}^{o} \alpha_{ak} a^k + \sum_{l=1}^{z} \alpha_{ld} d^l + \sum_{y=1}^{q} \alpha_{\beta y} \beta^y$$

Az eredeti polinom becslőfüggvényben a változók mind 3-adfokig szerepeltek, kivéve az adott szakaszon az emelkedő mértéke, ami 7-ed fokig hozott javulást a becslőfüggvényben. Az OLS becslés eredménye tehát egy olyan **α** együtthatóvektor, hogy az egyes szakaszokon a becsült átlagsebességek és a megfigyelt átlagsebességek különbsége minimális legyen, azaz

$$\min_{\alpha} \sum_{i=0}^{N} (v_i - \bar{v}_i)^2$$

Pitmanék cikkükben a menetidőbecslés személyre szabása érdekében bevezetnek egy plusz faktort, ami azt mutatja, sík terepen (olyan szakaszon, ahol az emelkedő -5° és +5° között van), milyen sebességgel teljesít az illető az átlagoshoz képest. Ezt a változót a mért teljesítménye és a sík terepen mért teljesítményének arányából számítják. Sajnos cikkükben nem közölték a becsült paraméterek tesztstatisztikáinak értékét, továbbá az általuk megjelenített becslő polinomban olyan változó is szerepel, melynek parciális hatását ábrázoló függvényének képe merőben eltér annak algebrai alakjától. Ettől eltekintve a szerzők jó eredményekről számolhatnak be: átlagosan

nagyjából 18%-os hibával tudják az egyes szakaszokon a hátralevő menetidőt megbecsülni modelljük segítségével, mellyel jócskán javítanak az adataikon nagyjából átlagosan 32%-os hibával teljesítő Naismith-szabályon.

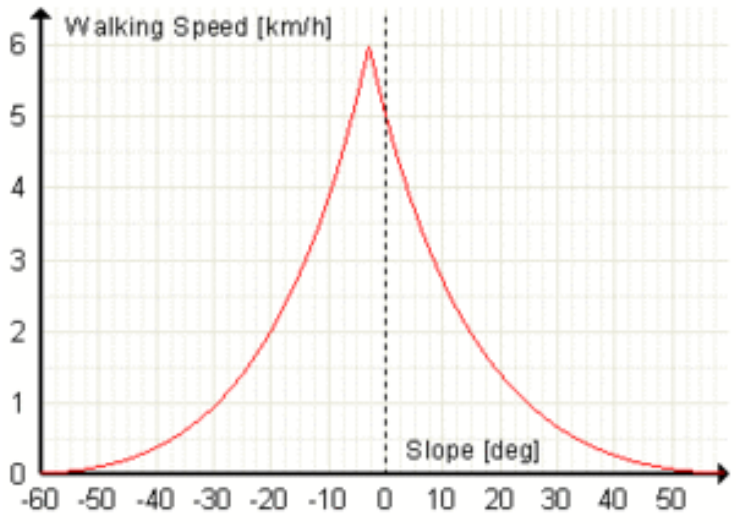
A sebesség többváltozós modellezését célzó kísérleteink akár a Pittmanék által rendelkezésünkre bocsátott 360 túranaplóból álló adathalmazon, akár a turautak.hu oldalról származó adatokon gyenge eredménnyel zárultak. Bár egyértelműen a meredekség bírt a legnagyobb magyarázó erővel az adott szakaszon mért sebességre vonatkozóan, a vizsgált modellek összességében rendre igen kis korrigált R^2 értéket adtak (< 0,1), míg sok változónk inszignifikánsnak bizonyult. Ezek teszteredményeit az *C.1 mellékletben* találjuk. Ennek tanulsága nyomán merőben más menetidőbecslő eljárást javaslunk, melyet a következőkben ismertetünk.

2.5.3. A sebesség becslése a meredekség függvényében

Ebben az alfejezetben a terep meredekségének sebességre gyakorolt parciális hatását vizsgáljuk. A Naismith által adott becslést az idők során többen is finomítani próbálták, többek között Waldo Tobler [28], aki exponenciális függvényt javasolt a sebesség közelítésére. A becsléséhez használt adatok Imhof 1950-es kartográfiai könyvéből származnak [41]. Az általa becsült sebességfüggvény az alábbi alakot ölti, (lásd a *10. ábrán*):

$$W = 6e^{-3.5\left|\frac{dh}{dx} + 0.05\right|} \qquad \frac{dh}{dx} = m = tg\beta$$

10. ábra: Tobler-görbe



ahol

- W - a becsült sebesség (km/h)
- *dh* - az emelkedési differencia
- *dx* - távolság
- *m* - meredekség (%)
- β - a meredekség szöge (fok)

Amint az látható, a becsült sebesség maximuma kb. 6km/h kis meredekségű lejtőn, míg a sebesség 0-hoz közelít ±60 fok esetén, tehát extrém meredekségű emelkedőn vagy lejtőn. Az ebből származó sebességértékek, bár nincsenek az emberi teljesítőképesség határán, de igen jó erőnlétet feltételeznek, így nem mondható átlagosnak. Másrészről nehezen indokolható a függvény csúcsossága a maximum pontjában. Mi a rendelkezésünkre álló kb. 2.400 túranapló alapján kívánjuk becsülni a túrázó sebessége és a terepszakasz meredeksége közötti összefüggést.

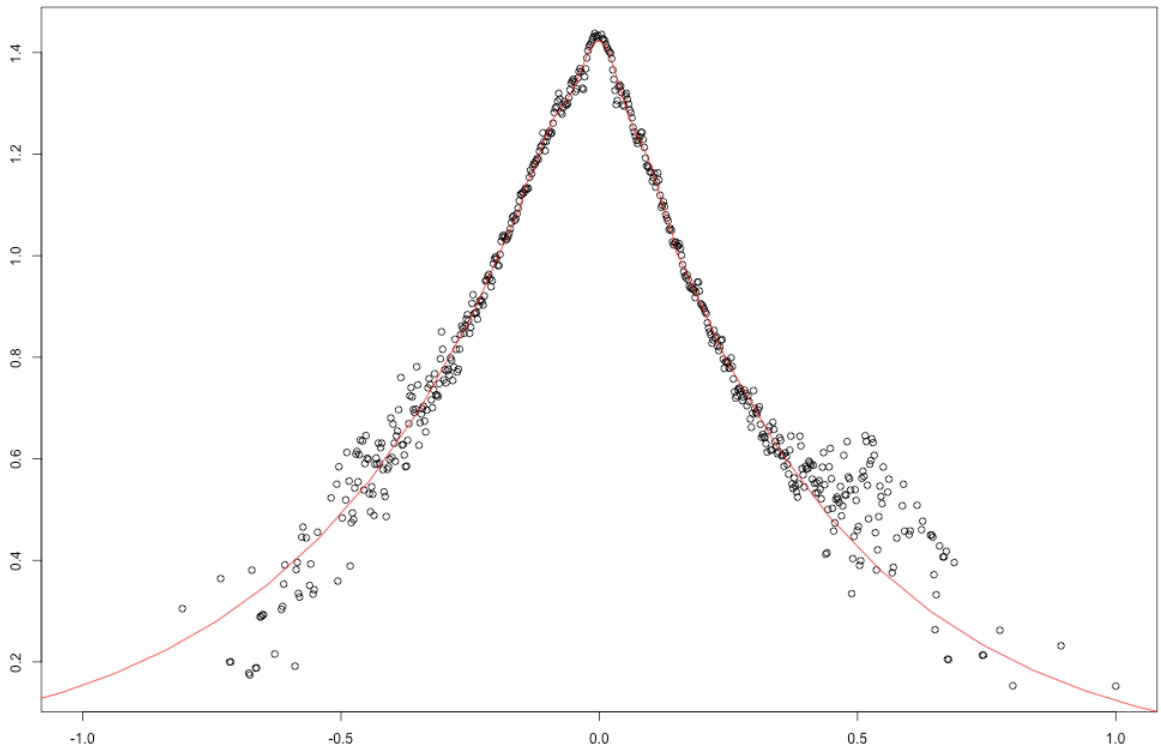
Az egyes szakaszokra vonatkozó sebesség-meredekség párokat tekintve (az outlierektől a korábbiakban leírt módszerrel való megtisztítás után) 1/4 fokonként haladva a meredekségi adatokon, az adott érték körüli ±0,125 fokos intervallumban található sebességek számtani átlagát véve számolunk egy átlagos sebesség értéket minden negyed fokhoz a meredekség skálán. Az R statisztikai szoftver lm (linear model) csomagja segítségével illesztettünk az átlagokra közelítő görbéket legkisebb négyzetek módszerével, mely QR mátrix dekompozíciós eljáráson alapszik. A módszerről bővebben Gulliksson és Wedin cikkében olvashatunk [264]. Az illesztett görbe a [-0,15; 0,15] intervallumon egy 10-ed fokú polinom, míg a széleken egy-egy exponenciális függvényt illesztettünk az átlagokra. Ennek legfőbb oka, hogy korábbi kutatási eredményeink alapján a Tobler-görbe maximum pont körül rosszul illeszkedik valós adatokra, így azt minél inkább igyekeztük lekövetni egy polinommal, másrészt a polinom a széleken rendszerint rosszul illeszkedik, ezért a gyakorlatnak sokkal inkább megfelelő (a széleken a vízszintes tengelyhez simuló) görbét kerestünk. Az illesztés eredményeként az alábbi sebességet becslő függvényt kaptuk:

$$v(m) = \begin{cases} e^{2.3203m+0.4462} \mid m \in (-\infty; -0,15) \\ p(m) \mid m \in [-0,15; 0,15] \\ e^{-2.4672m+0.3769} \mid m \in (0,15; \infty) \end{cases}$$

ahol m az adott szakasz meredekségét jelöli. A *p(m)* polinom együtthatóit és az illesztés tesztstatisztikáit összefoglaló táblázatot a *C.2 mellékletben* találjuk. A sebességet a meredekség függvényében becslő v(m) görbénket a *11. ábrán* láthatjuk. A sebességet (m/s) mérő függőleges skálán jól látható, hogy nagyjából 5km/h a becsült maximális sebesség az átlagok alapján, szemben. Tobler 6km/h-s maximális sebességével. A maximumhely, Tobler eredményéhez hasonlóan, nagyjából -2° körül található. Az illesztett görbék tesztstatisztikái alapján elmondható, hogy mindhárom szakaszon van kapcsolat az adott szakaszon mért sebességek átlaga és szakaszok meredeksége között, erre utal a magas korrigált R^2 érték, valamint a magas F-statisztika értékek (a hozzájuk tartozó rendkívül alacsony p-értékekkel). Az abszolút értékben kisebb meredekségű

szakaszokra illesztett magasabb fokszámú polinom okán fontosnak tartottuk a korrigált R^2 mutatóra hagyatkozni, elkerülendő a túlillesztést. Az illesztett *v(m)* görbe, mint azt a következő alfejezetben látni fogjuk, pontosabb menetidőbecslést tesz lehetővé, mint a Tobler-görbe.

11.ábra: Az átlagsebességekre illesztett becslés (meredekség radiánban, sebesség m/s-ban mérve)



2.5.4. Két menetidőbecslő eljárás

Az előző alfejezetben ismertettük a túrázó sebessége és a túraszakasz meredeksége között becsült összefüggést a Tobler-görbe nyomán. Ezt alapul véve a jelen szakaszban két menetidőbecslő eljárást ismertetünk, melyet a korábban bemutatott 2400 túranaplón teszteltünk.

1. Meredekség alapú eljárás

A korábbiakban bemutatott eljárás szerint illesztünk sebességet becslő görbét a teszthalmazban szereplő túranaplók szakaszainak meredekség-sebesség párjaira, majd ezt személyre szabjuk az alábbiak szerint: A túraút első 20 másodperces szakaszának meredeksége legyen m_1, ekkor ennek sebességét becsüljük az illesztett *v(m)* görbe alapján $v(m_1)$-gyel. A második szakasz sebességének becslésénél már felhasználjuk, hogy az előző szakasz becsült értékét össze tudjuk hasonlítani a

valós adattal (túra közben). A valós és becsült sebesség arányát tekinthetjük ezt egy fittségi faktornak, mely adott meredekség mellett az átlagos túrázó sebességétől vett eltérését mutatja. Legyen ennek értéke $b_1 = v_1/v(m_1)$. Ekkor a második szakasz becsült sebessége legyen $b_1v(m_2)$, ahol m_2 a második 20 másodperces szakasz meredeksége. A harmadik szakasz sebességének becslésekor már felhasználjuk a 2. szakasz megfigyelt fittségi faktorát is, és vesszük a számtani átlagukat, tehát a becsült sebessége $(b_1+b_2)v(m_3)/2$ lesz. Általánosan az n-edik szakasz sebességének a becslése az alábbiak szerint történik:

$$v_n^* = \frac{\sum_{i=1}^{n} b_{i-1}}{n-1} v(m_i)$$

Ezzel az útközben történő kiigazítással a teljes úthosszra tett becslést jelentősen javíthatjuk, hiszen olyan befolyásoló körülményeket tudunk részben leképezni, mint az időjárási viszonyok, vagy a túrázó aktuális napi erőnléti szintje. Bár az első néhány szakaszon (jellemzően az út első 10%-án) az ingadozó fittségi faktor értékek miatt még pontatlan az eljárás, a továbbiakban - mint azt látni fogjuk - igen jó becslést adhatunk a menetidőre. A kísérlet során megpróbáltuk ezt a fittségi faktort nem csupán "globálisan" meghatározni egy adott túrázó esetén, de akár meredekségi intervallumokra külön-külön. Gyakorlati tapasztalatunk szerint enélkül ugyanis figyelmen kívül hagyjuk, ha valaki sík terepen kiválóan teljesít ugyan, de az emelkedőkre rosszul reagál. Mivel azonban így több faktort is becsülnünk kell a túra során, ezért mire azok értékei "stabilizálódnak", már jellemzően igen sok szakaszt megtett a túrázó, így összességében ezzel a kiterjesztéssel rosszabb eredményeket értünk el, mintha csak egy fittségi faktort becsülnénk.

2. Az átlagsebesség alapú menetidőbecslés

Az eljárás a túra első 20%-ában az illesztett sebességgörbe alapján becsli a sebességet a teljes útra, miközben minden szakasz sebesség értékeit elraktározza. Legyen az i-edik, már megtett szakasz megfigyelt sebessége v_i, ekkor az n-edik szakasz (mely már túl van a túra első 1/5-én) sebességét az alábbiak szerint becsüljük:

$$v_n^* = \frac{\sum_{i=1}^{n} v_{i-1}}{n-1}$$

tehát egyszerűen vesszük az előző n-1 szakaszon megfigyelt sebességek átlagát. Jellemzően a túra első 1/5-ében ez az eljárás még nem ad jó sebességbecslést, ezért kell ott helyettesítenünk az illesztett sebességgörbe által adott becsléssel. A következő alfejezetben összegezzük a becslő eljárásaink jóságának vizsgálatait.

2.6. Az eredmények kiértékelése

Becslésünk pontosságát mérendő, szeretnénk azt a Tobler-görbe alapján kalkulált becslésekkel összevetni. Összehasonlítási mértékként mi is (akárcsak Pittman et al. [30]) az átlagos abszolút relatív hiba (mean absolute relative error, röviden MARE) értékét használjuk, mert egyformán bünteti az alul- és felülbecslést is. A teljes utat 100 részre bontjuk, és *p*-vel jelöljük, hogy az út hány százalékánál tartunk. Az *i*-edik útra a *p*-edik szakaszhoz tartozó, mért adatokon alapuló hátralévő időt jelölje r_{ip} míg az általunk becsült hátralevő időt r^*_{ip}.

$$MARE(p) = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{r_{ip} - r^*_{ip}}{r_{ip}}\right|$$

A számításnál a 2400 túrából álló adatbázisunkat tanuló- és teszt adathalmazra (75-25%) bontottuk véletlen módon. A tanuló adathalmaz alapján végeztük a 2.5.3-as szakaszban bemutatott görbe illesztését a sebesség átlagokra (a meredekség függvényében), majd az így kapott görbét alkalmaztuk a teszt adathalmazon a már ismertetett két módszer szerinti menetidőbecslő eljárások során. Ezt az eljárást 10-szer alkalmaztuk egymás után, és a fenti képlet alapján kalkulált MARE értékeket az 1. *táblázatban* foglaltuk össze (az out végződésű oszlopokban a vonatkozó kalkulációs eljárás értékei szerepelnek 20%-nál magasabb MARE értékek nélkül).

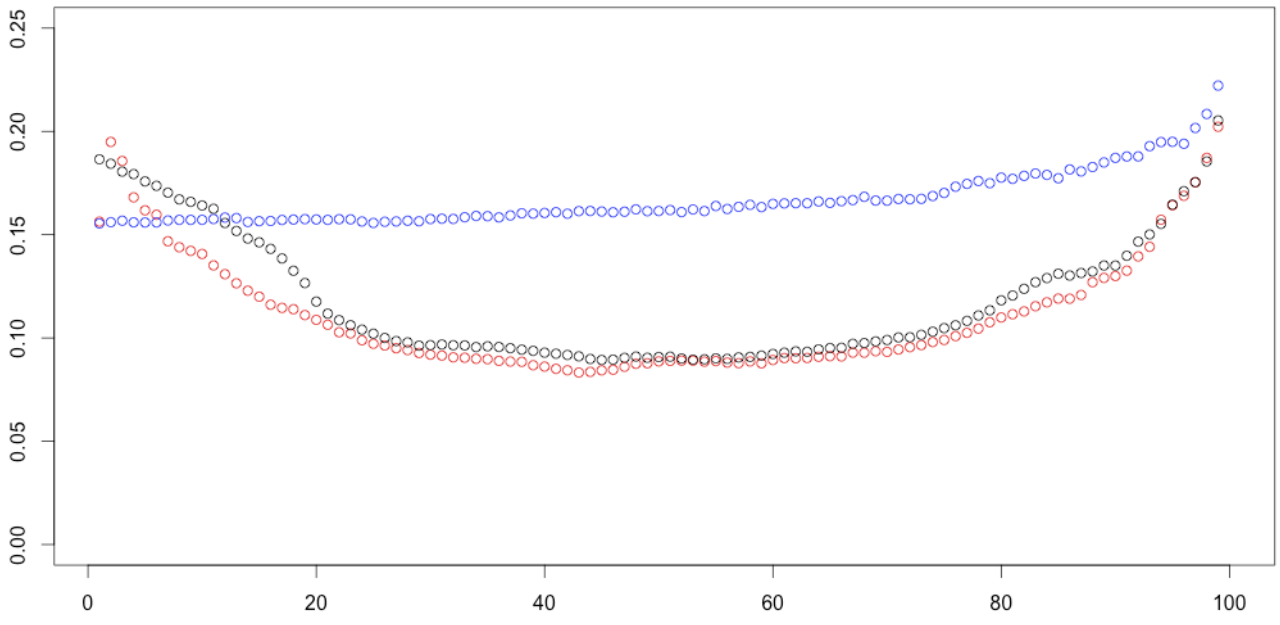
1. táblázat: A három eljárás MARE értékeinek összehasonlítása

	MARE_mer	MARE_mer_out	MARE_atl	MARE_atl_out	MARE_Tobler
Teszt_1	11,90%	9,51%	13,00%	9,47%	17,06%
Teszt_2	11,41%	9,38%	12,87%	9,56%	16,74%
Teszt_3	11,01%	9,36%	12,89%	9,66%	17,38%
Teszt_4	11,95%	9,00%	12,74%	9,98%	16,72%
Teszt_5	11,44%	9,38%	12,55%	10,03%	17,80%
Teszt_6	11,64%	9,44%	13,26%	9,60%	18,25%
Teszt_7	11,59%	9,63%	13,56%	9,92%	16,35%
Teszt_8	11,91%	9,30%	12,73%	10,24%	19,48%
Teszt_9	11,55%	9,15%	12,49%	10,18%	17,09%
Teszt_10	11,36%	9,53%	13,33%	9,85%	16,76%
átlag	**11,58%**	**9,37%**	**12,94%**	**9,85%**	**17,36%**

A MARE értékek azt mutatják, hogy az illesztett sebességgörbén alapuló eljárás teljesít a legjobban, míg azt nem sokkal lemaradva követi az átlagsebességre épülő becslésünk. Mindkét eljárás által becsült eredmények szignifikánsan jobbak a Naismith-szabály által prognosztizált menetidőknél, sőt a Pitmanék által javasolt módszer eredményeinél is átlagosan nagyjából 5 százalékponttal jobb becslést ad, bár utóbbit sajnos nem tudtuk összevetni a saját eredményeinkkel azonos adatbázison végzett tesztekkel. A teszt túranaplókon végzett menetidőbecslések MARE értékeit mindhárom vizsgált eljárásra a *12. ábrán* foglaltuk össze (a piros a meredekségen, a fekete az átlagsebességen alapuló eljárást jelöli, míg kékkel tüntettük fel a Tobler-görbe alapján becsült menetidők MARE értékeit). Ezen MARE értékek mindhárom eljárás esetében a 10 elvégzett kísérlet számtani átlaga alapján kerültek kiszámításra. Amit érdemes megemlíteni, hogy az első szakaszokon mindkét eljárásunk gyengébben teljesít, hiszen ezen szakaszok alapján becsüljük az individuális korrekciókat. A középső 60%-on jól teljesít a becslés, csupán az utolsó 20% az, ahol az eredmények romlanak, amit több okra vezethető vissza:

- A relatíve kevés hátralévő adatponton a kisebb variancia is nagyobb hatással van az eredményekre.
- Sajnos arra vonatkozóan nincsenek adataink, hogy a 360 túra közül melyek lehettek teljesítménytúrák, ahol szokás a végén hajrázni. Versenyhelyzettől vagy időkorláttól függetlenül is sokan új erőre kapnak a cél közelében.
- Nem számoltunk a túrázók kifáradásával, mely teljesítményük romlásához vezet. Ez főleg a kevésbé fitt populációt érinti.

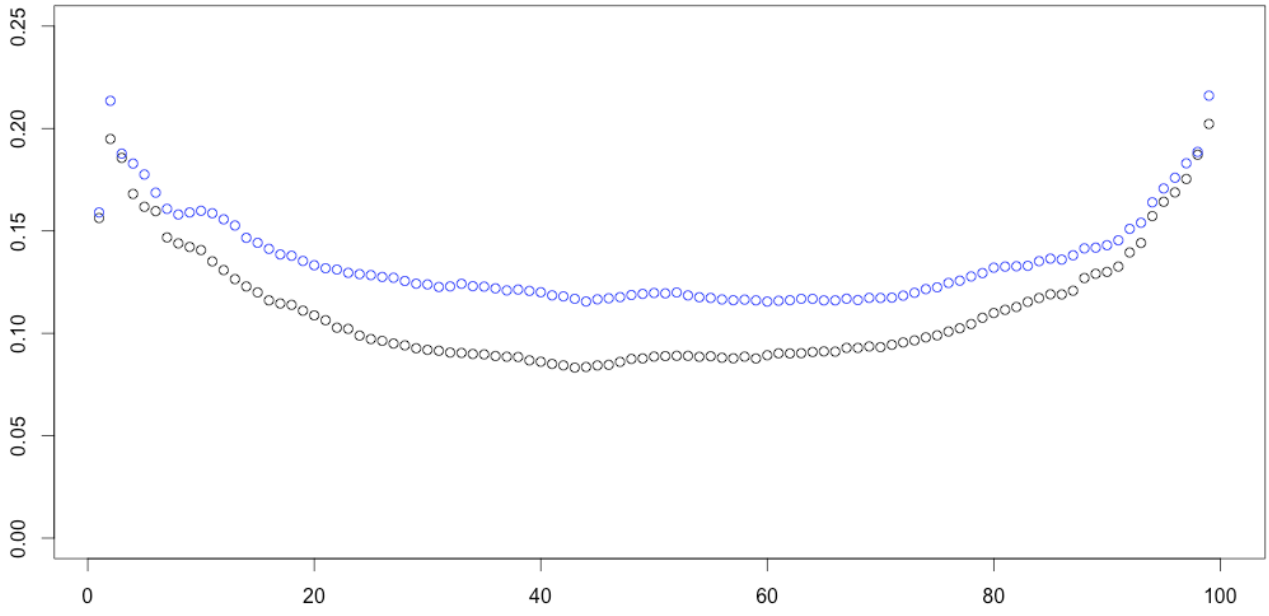
12.ábra: A három eljárás MARE értékei (a megtett út %-ának függvényében)



Eredményeink alapján tehát a Tobler-görbét alapul véve kisebb pontosságú menetidőbecslést kaptunk, mint saját becsült sebesség-meredekség összefüggésünk alapján, továbbá elmondható, hogy a meredekség alapú menetidőbecslő eljárás is szignifikánsan jobb a Tobler által adott becslésnél, de pontatlanabb, mint a meredekség alapú közelítés.

Sort kerítettünk továbbá a Toblet-görbe és az általunk illesztett *v(m)* sebességbecslő görbe összevetésére a meredekség alapú becslő eljárás segítségével is. Az egyes görbéket a meredekség alapú becsléshez felhasználva, a fittségi faktorokkal folyamatosan korrigálva, 10-10 tesztet elvégezve azt találtuk, hogy míg a *v(m)* sebesség-meredekség összefüggést alapul véve az első eljárásunk átlagos hibája 11,12% volt, addig a Tobler-görbét alapul véve ez az érték 13,48% volt. A két eljárás MARE értékeinek összehasonlítását a *13. ábrán* mutatjuk be, ahol a Tobler-görbéhez a kék, míg a mi becslő függvényünkhöz a fekete pontok tartoznak. A kétmintás t-próba alapján elutasítjuk a nullhipotézist, miszerint a két eljárás MARE értékeinek átlaga megegyezik. A t-próba eredményeit a *C.3 mellékletben* találjuk.

13.ábra: A Tobler-görbe és a *v(m)* alapján kalkulált meredekség alapú eljárás MARE értékei (a megtett út %-ának függvényében)



2.7. Konklúzió és kutatási tervek

Tanulmányunkban arra tettünk kísérletet, hogy túrautak menetidejének becslésére adjunk egy a jelenlegieknél pontosabb megoldást. A rendelkezésünkre álló túranaplók alapján elsőként adtunk egy teljes populációra vonatkozó sebességbecslést az adott szakaszok meredekségének függvényében, mely a Tobler-görbe pontosságát igyekezett javítani. Erre építve két eljárást dolgoztunk ki: az egyik a kezdeti átlagsebesség értékeket megfigyelve tisztán azok alapján becsüli a túra hátralevő részére a menetidőt, míg a másik eljárás a teljes populációra illesztett sebesség-meredekség görbe alapján becsült menetidőket szabja személyre az egyéni eredmények alapján dinamikusan becsült fittségi faktorokkal. Mindkét eljárás túlszárnyalja pontosságát tekintve az eddig ismert menetidőbecslő eljárásokat, és egyszerűségüknek köszönhetően igen alacsony a számítási igényük, így azok mobil alkalmazáson történő implementációja is indokolt. A becslések további javítására több terv is született a munka során.

A későbbiekben továbblépési lehetőség lenne a modellben szerepeltetett magyarázó változók bővítése, úgy mint a túrafelszerelés össztömege, időjárási körülmények, az út típusa vagy az adott pontig eltöltött teljes pihenőidő, melyeket most - adatok hiányában - mellőztünk. Sajnos a rendelkezésünkre álló adatok további terveink kivitelezéséhez nem elegendőek, az alábbiakat mégis fontosnak tartjuk a jövőben vizsgálat tárgyává tenni:

- Érdemes lehet a későbbiek során megvizsgálni, hogy két becslő eljárásunk pontossága javítható-e egy vagy akár több további magyarázó változó szerepeltetésével. Az általunk tesztelt többváltozós modell magyarázó ereje ugyan igen csekély volt, egy-egy új magyarázó változó szerepeltetése jelenlegi becslő eljárásainkban még javíthat annak pontosságán.
- Amennyiben a meredekségi alapú becslésünk esetén a teljes populációt fittség szerint szegmentáljuk 3 külön populációra, és azokra illesztünk egyenként sebesség-meredekségi görbéket, azzal lehet, hogy javítanánk becslő eljárásunk pontosságán. Ez esetben is korrigálnánk a becsült sebességeket a megfigyelt fittségi paraméterekkel, ám azok várhatóan már 1-hez közelebbi értékek lennének, hiszen homogénebb csoportokat hoztunk létre.
- Az azonos szakaszokon mért (átlagos) teljesítményük alapján rangsorolhatjuk/klasszifikálhatjuk a túrázókat, és ez alapját képezheti egy “legközelebbi szomszéd” alapú becslő eljárásnak. Ehhez azt kell figyelembe venni, hogyan teljesítették az egyes szakaszokat a többiek, és hozzájuk képest (hasonló terepviszonyok mellett) hogyan teljesített a vizsgált személy. Ebből szakaszonként tudunk adni egy esztimációt a menetidejére, így a teljes útjára is.

- Fontos, hogy ennek érdekében minden szakaszt tudnunk kell klasszifikálni különféle dimenziók mentén. Ha nincs a terepviszonyokra klasszifikálási lehetőség, akkor marad a meredekség, illetve az, hogy a felhasználók átlagosan hogyan teljesítettek egy olyan szakaszhoz képest, amiről tudjuk, hogy milyen nehézségű terep (pl. aszfalt), vagy legalábbis azt, hogy közel vízszintes. Ilyen módon akár az útszakaszok nehézsége is klasszifikálható lenne, melyet külön változóként kezelhetünk, és ezt is szerepeltethetjük a becslőfüggvényben. Ez ugyanis nem csak a terep meredekségét tartalmazná, hanem a extra körülményeket, mint az eddig figyelmen kívül hagyott tereptípus, mint változó. Gondoljuk csak meg, mennyiben befolyásolja a teljesítményünket, hogy minden más változatlansága mellett palás kőzeten, vagy morzsalékos talajon túrázunk.
- Pitman cikkében megállapítja, hogy a sebesség a túra teljes hosszával először nő, majd csökken (ha már a túrázók megerőltetik magukat), majd ismét nő (mivel csak gyakorlott túrázók mennek több, mint 20km-t). Ezért tartjuk fontosnak a túrázók klasszifikálását is, mert ha ezt túrázó típusonként vizsgálhatjuk, akkor várhatóan a profi túrázóknál kevésbé lesz észlelhető a fáradékonyság hatása, így extra korrekcióként szerepeltethetjük a fáradékonyságot, melyet a teljes populációra nem tudtunk kimutatni az adatsorokon.
- Kollaboratív megközelítés: a későbbiekben a klasszifikációk alapján egy második, önálló becslést is adhatunk a menetidőre, ha mások ezeken a szakaszokon történt teljesítményeiből számoljuk azt. Ennek több módja is lehet. Elegendő adatot feltételezve módunkban áll összeilleszteni a tervezett túrát a hasonló szintre klasszifikált turisták adott szakaszokon mért eredményeinek valamilyen átlagolásával. Kevesebb adatnál elég lehet azt figyelembe venni, hogy hasonló terepkörülmények esetén (lásd a fenti változók), mások hogyan teljesítettek a vizsgált személyhez képest, így a vizsgált útszakaszon a becslés előáll az előbbi alapján becsült faktor és a tesztalanyok vizsgált útvonalon adott teljesítményének szorzataként. Ilyen vizsgálathoz azonban szükség van arra, hogy egy adott felhasználónak több túranaplója is rendelkezésünkre álljon, valamint legyenek olyan útszakaszok, melyeket a vizsgált alant és hozzá hasonlóan klasszifikált felhasználók is megettek korábban, mely az összehasonlítás alapját képezi.

3. Turisztikai ajánlórendszer

3.1. Bevezetés

Mindennapi életünk során számtalan alkalommal kerülünk döntési helyzetbe, sokszor akár észrevétlenül. Mit vegyünk fel reggel, ami megfelel a napi programunkhoz? Melyik menüt válasszuk az ebédlőben? Melyik munkához kezdjünk neki előbb? Melyik iskolába irassuk gyermekünket? Ilyen és ehhez hasonló sorsdöntő, vagy éppen hétköznapi kérdések ezreire adunk választ életünk során. Gyakran ezekben a döntésekben szakértők vagy barátok segítségét kértük a múltban, ám egy ideje rendelkezésünkre állnak más lehetőségek is. A következő olvasmányunk kiválasztásában már nem csak a könyvtáros vagy a könyvesbolti eladó segíthet, hanem akár egy olyan, könyveket (is) árusító weboldal, mint az amazon. A Youtube által felajánlott videók mind a korábbi böngészéseinken alapulnak, és viszonylag nagy találati aránnyal javasol olyan audiovizuális tartalmakat, mely kedvünkre való. Mérhetetlen előnye a barátok javaslatain alapuló, hétköznapi megoldáshoz képest, hogy míg a fent említett oldal a világ legnagyobb videótárának teljes figyelembevételével teszi javaslatait, addig ismerőseink eggyüttes rálátása is ennek csupán töredéke. Ily módon például olyan együttesek dalait is megismerhetjük, akikkel nagy valószínűséggel sosem találkoztunk volna más módon. Ekkor ugyanis nem csak az ismerőseink ajánlhatnak nekünk tartalmakat, hanem a világon mindenki ezt teszi - akaratlanul - az ajánlórendszeren keresztül. Fontos azonban itt leszögezni, hogy - mint azt a későbbiekben látni fogjuk - az ajánlórendszereknek is megvannak a maguk korlátai, így vélhetően (és remélhetőleg) soha nem fognak minket olyan jól ismerni, mint barátaink és rokonaink. Ezeket is megfontolva a látszatát is szeretném annak elkerülni, hogy a hétköznapi emberi kapcsolatok ajánlórendszerekkel történő kiváltására szeretnék buzdítani bárkit is. Tekintsük ezeket sokkal inkább egy lehetőségként, mely segíthet a mindennapokban dönteni bizonyos - kevéssé fontos - kérdésekben, megspórolva ezzel magunknak némi időt, vagy ráakadni olyan élményekre, melyek talán örökre elkerültek volna minket ezen rendszerek hiányában.

A továbbiakban összefoglalom az ajánlórendszerekkel kapcsolatos definíciókat, majd a kutatási témával kapcsolatos motivációkra térek rá. A következő szakaszban sor kerül az ajánlórendszerek rövid történeti összefoglalójára, valamint az alkalmazott technikákat és az azokkal kapcsolatos kihívásokat tárgyaljuk. A fejezet további részében ismertetjük egy turisztikai helyszínekkel kapcsolatos ajánló rendszer modelljét és annak empirikus eredményeit. A fejezetet a kutatás soránt levont következtetésekkel és lehetséges továbblépési lehetőségekkel zárjuk.

3.2. Ajánlórendszerekkel kapcsolatos alternatív definíciók

Az ajánlórendszerek tárgyalásához szükségét látom néhány fogalom definiálásának. Itt erősen támaszkodnék a széles körben elterjedt és használt Wikipedia definiciókra kiegészítve néhány szakirodalmi alternatívával.

Információ: "Általánosságban információnak azt az adatot, hírt tekintjük amely számunkra releváns és ismerethiányt csökkent. Egyik legleegyszerűsítettebb megfogalmazás szerint az információ nem más, mint valóság (vagy egy részének) visszatükröződése." [42a]

Információszűrő rendszer: "Olyan rendszer, ami eltávolítja a redundáns vagy nemkívánatos információt az információs folyamból automatizált vagy számítógép által előállított módszerrel, mielőtt a(z emberi) felhasználó elé kerülne. Fő feladata, hogy kezelje az információs túlterhelést és javítsa a jel-zaj arányt." [42b]

Ajánló rendszer: "Speciális információszűrő rendszerek, amelyek felhasználói és termékprofilokat építenek tanuló algoritmusok segítésével, majd a modellek alapján ajánlanak olyan tartalmat (film, tv, zene, könyv, hír, kép, weboldal, cikk, stb.) a felhasználónak, amely nagy valószínűséggel érdekes lesz számukra." [123, p. 24.]. Abban mindenképp igaza van Riccinek és szerzőtársainak, hogy az ajánlórendszer egy speciális információszűrő rendszer, azonban a túlspecifikált a definíciójuk, mert egyrészt az eszköztárat is túlzottan leszűkítik, másrészt a "tartalomnál" is használhatnánk bővebb fogalmat (egy adott kontextusban bármilyen opciót ajánlhat a rendszer), harmadrészt a "nagy valószínűséggel érdekes" megfogalmazást nem tartom elegendően pontosnak.

Egy másik, igen hasonló megközelítés szerint, mely Melville és Sindhwani munkáján alapszik : "Az ajánló rendszerek fő célja, hogy felhasználók egy csoportjának olyan ajánlásokat tegyen bizonyos termékekre vonatkozóan, melyek valószínűleg érdeklik őket." [43]. Ennek a definíciónak az előzőhöz teljesen hasonlatos problémái vannak.

A citizendum megközelítése szerint: "Olyan szoftver program, mely megkísérli a felhaszálók számára a választékot a kifejezett preferenciájuk, a múltbeli viselkedésük, vagy a hozzájuk hasonló érdeklődéssel bíró felhasználókról gyűjtött információ alapján." [44]. Itt is feltűnő az ajánlórendszer túlspecifikálása, illetve a nehezen értelmezhető "szoftver program" okozhat némi fejtörést. A definícióban megjelenik a kollaboratív, valamint a termék és tudás alapú eljárások gondolata, ám ez sem kellően univerzális.

A fenti definíciók abban mindenképp megegyeznek, hogy a döntési helyzet során a választási lehetőségek listáját próbálják szűkíteni, illetve egyes helyeken rangsorolni is azokat. Ezeket figyelembe véve az alábbi saját definícót javaslom:

Ajánló rendszer: olyan információszűrű rendszer, mely egy adott döntési helyzetben a lehetséges opciók halmazának szűkítésével, illetve az elemeinek adott kontextusban történő rangsorolásával támogatja a felhasználót. A rangsorolás történhet a felhasználó explicit vagy implicit módon kifejezett preferenciái alapján, illetve a hozzá hasonló preferenciákkal bíró felhasználók korábbi viselkedésének figyelembe vételével.

3.3. Motiváció és a téma relevanciája

A minket körülvevő digitális világ soha nem látott módon - és egyre növekvő mértékben - zúdítja ránk az információ tömkelegét, melyből valódi kihívást jelent kiválogatni a számunkra fontos elemeket. Az ajánlórendszerek célja éppen ezen információáradat megszűrése, így segítve minket abban, hogy csak a számunkra releváns tartalmakkal tudjunk foglalkozni, minimalizálva az erre fordított időt. Az okostelefonok elterjedésével ez a segítség folyamatosan elérhetővé vált számunkra igen sok területen. Személyes tapasztalatom alapján ez a komfort még nem áll kellő mértékben rendelkezésünkre utazásaink közben. E dolgozat készülése idején nem ismert még egy olyan online szolgáltatás sem, mely figyelembe véve preferenciáinkat és lehetőségeinket - ide értve az anyagi és időkorlátokat - egy számunkra még ismeretlen városban akár több napra programot ajánlana nekünk és útvonaltervet készítene hozzá. Ezt a hiányt igyekszik pótolni ez a kutatás, melynek - reményeim szerint - gyakorlati megvalósulítására is sor kerül majd a jövőben. Célom egy olyan ajánlórendszer megalkotása, mely megkönnyíti a túristák számára a programtervezést, és általa olyan helyszínekre is eljuthatnak, melyet maguktól talán soha nem fedeztek volna fel.

Mégis ha csak egy okot emelhetnék ki, amiért e gyakorlati területtel foglalkozni érdemes, egy személyes élmény jut eszembe, melyet a Procter&Gamble néhány éve bevezetett, megújult ajánlórendszerénél tapasztaltunk: egy amerikai diáklány vásárlásai alapján a rendszer úgy észlelte, hogy a fogyasztója nagy valószínűséggel terhes, így kismamáknak szóló terméket is ajánlott számára. A lány szülei be akarták perelni a céget, ám hamar letettek ezen szándékukról, miután szembesültek a ténnyel: a rendszer nem tévedett [55].

Jelen tanulmány távlati célja nem kevesebb, mint a túristák számára minél könnyebbé és hatékonyabbá tenni a tervezést.

3.4. Az ajánlórendszerek története

A számítógépek polgári célú elterjedésével párhuzamosan egyre inkább a fejlesztő cégek és kutatók fókuszába került a felhasználói igények egyre szélesebb körű kiszolgálása. A gépek népszerűségének rohamos növekedése mögött rendkívül komoly erőfeszítések rejlenek, amit az ember és gép közötti "súrlódások" csökkentése érdekében fejtettek ki. A felhasználók számára egyre komfortosabb megoldásokkal tudtak előállni köszönhetően annak, hogy megpróbálták az emberek igényeit megérteni, és számítógép által nyújtott szolgáltatásokat személyre szabni.

Az ajánlórendszerek alapjait a megismeréstudmány [45] és az információ visszanyerés (information retrieval) [49] kutatásai alapozták meg, és az első manifesztációja a Duke Egyetem által a '70-es évek második felében megalkotott Usenet kommunikációs rendszer [105], amin keresztül a felhasználók szöveges tartalmat oszthattak meg egymással. Ezeket hírcsoportokba és alcsoportokba kategorizálták a könnyebb kereshetőség érdekében, azonban nem direkt módon épített a felhasználók preferenciáira és nem is célozta azok megismerését. Az első ilyen irányú ismert megoldás a Grundy nevet viselő számítógépes könyvtáros volt, ami a felhasználókat előbb kikérdezte a preferenciáikról, majd ezt figyelembe véve ajánlott számukra könyveket. A rendszer egészen primitív módszerrel sorolta be az összegyűjtött információ alapján a felhasználót egy sztereotípia csoportba, s így minden azonos csoportba tartozó személy számára ugyanazokat a könyveket ajánlotta. A Grundy megoldásának eredményeiről és annak népszerűségéről a felhasználók körében Rich 1979-es cikkében [45] olvashatunk bővebben. Ma már kissé idejétmúltnak tűnhet ez a megközelítés, de akkor ez egy paradigmaváltás volt az automatizált kiszolgálás terén, hiszen személyre szabottá tették azt. Fontos megjegyezni, hogy ezt a mérföldkövet, akár napjainkban sem minden internetes bolt tette meg. A Grundy megoldásának azonban gyorsan igen sok kritikusa akadt a tudományos világban. Nisbett és Wilson megfogalmazzák, hogy "az emberek igen gyengék a kognitív folyamataik vizsgálatában és leírásában" [46]. Vizsgálataik szerint az emberek gyakran olyan tulajdonságaikat emelik ki, amivel egy adott csoport többi tagja közül ki tudnak tűnni, megnehezítve ezzel a sztereotipizálási törekvéseket. Természetesen előfordulhat az is, hogy egyszerűen csak más képet szeretnénk festeni magunkról. Ahogyan Észak-Európa egyik legnagyobb bevásárlóközpontjának vezetője, Heli Vainio fogalmaz kissé sarkosan idén nyári interjújában: "a kérdőívekre úgy válaszolnak az emberek, hogy jobb színben tűnjenek fel. Nem érdekelnek a hazugságok. A tények érdekelnek." [47]. Ennek érdekében fel is szereltette bevásárlóközpontját olyan Wi-fi berendezéssel, amivel a látogatókat 2

méter pontossággal nyomon tudja követni egyénenként az épületen belül és annak közvetlen közelében. A cél, hogy beszéljenek a látogatók helyett a cselekedeteik.

Az ajánlórendszereknek alapvetően két merőben eltérő irányvonala alakult az idők folyamán: a kollaboratív szűrés (collaborative filtering) módszere és a tartalom alapú szűrés (content-based filtering). Előbbi esetén a felhasználók ízlésvilágát próbálja a rendszer feltérképezni (profilozni), majd olyan tartalmakat ajánl neki, amelyet hozzá hasonló preferenciákkal bíró felhasználók kedveltek. A tartalom alapú szűrés lényege, hogy az ajánlandó entitás dimenzióit ismerje a rendszer (zenei tartalom ajánló rendszer esetén például az alábbi dimenziók jöhetnek szóba: stílus, előadó, korszak, hangszerelés, stb), illetve a felhasználó ezekre a dimenziókra, vagy karakterisztikára vonatkozó preferenciái. Így valahányszor kedvel egy újabb dalt a felhasználó, a profilját ezekkel az új információkkal bővítik ki. Létezik még tudás alapú szűrési eljárás is, illetve a fentiek keverékéből előálló hibrid rendszerek, melyekről a későbbiekben bővebben szólunk.

A kollaboratív szűrés első példája, ahonnan egyébként az elnevezése is származik, a Xerox PARC által kifejlesztett Tapestry rendszer volt, amely a felhasználóinak lehetővé tette, hogy az olvasott dokumentumaikhoz jegyzeteket készítsenek és véleményt nyílvánítsanak azokról (kezdetben bináris formában: kedveli vagy nem kedveli). A felhasználók ezután nem csak a dokumentumok tartalma alapján tudták manuálisan szűkíteni a keresést, de más felhasználók jegyzetei és értékelései alapján is, mely megfelelő felhasználószám elérése után már igen jól tudta rangsorolni a tematikus dokumentumokat relevanciájuk, hasznosságuk alapján [48]. Az 1992-ben indulo GroupLens [105] már képes volt automatizált módon ajánlásokat tenni a Usenet cikkekre vonatkozóan, ha a felhasználó előzetesen már értékelt néhány cikket a rendszerben. Ennek mintájára a következő években megannyi tematikus ajánló oldal született, mint például az MIT-nál fejlesztett Ringo, majd később a Firefly zenei ajánló oldalak, vagy a BellCore filmajánló. Az első megoldás, mely nem csupán egy szűkebb tematikát próbált felölelni, ám nem kevesebbet, mint magát az internetet, az 1994-ben - akkor még más néven - induló Yahoo! volt. A két stanfordi diák egy tematikus weboldal katlógust készített indexelt oldalakkal, mely igen hamar népszerűségre tett szert, és milliók számára jelentett könnyebb keresést az interneten, és az Alexa-rangsor alapján ma is az 5. leglátogatottabb weboldal.

A tartalom alapú szűrés gyökereit a információ visszanyerés (information retrieval) területén kell keresnünk, melynek technikái közül is igen sokat átörökítettek. Az első dokumentált megoldás Emanuel Goldbergtől származik az 1920-as évekből (ha nem számítjuk ide az 1801-ben bemutatt Jaquard-féle szövőszéket, a Hollerith-lyukkártya elődjét), mely egy olyan “statisztikai gép” volt, ami mintákat keresve a celluloid szalagon igyekezett ott tárolt dokumentumokat automatizált

módon megtalálni [49]. Az 1960-as években a Cornwall Egyetemen Salton körül szerveződő kutató csapatnak köszönhetően közel egy évtized alatt alkották meg a szövegek automatikus indexelésére alkalmas modelljüket, mely alapját képezi a ma ismert szövegbányászati eljárásoknak [50]. Az eljárás igen egyszerű: a dokumentumok egyes előre meghatározott ismérvek (dimenziók) mentén kerülnek osztályozásra, melyeket - mint indexeket - egy vektorba gyűjtünk. Minél inkább hasonlít két dokumentum egymásra, az őket leíró vektorok által bezárt szög annál kisebb. A következő mérföldkő az 1979-ben Doszkocs Tamás által a Natinal Library of Medicine számára kifejlesztett CITE online katalógus rendszer volt, mely nem csupán azt tette lehetővé, hogy a könyveket kategóriák szerint kereshessék a felhasználók, de a keresőszavak alapján relevancia szerint rendezte sorba.

A tartalom alapú szűrés viszonylag későn, a 90-es években nyert önálló létjogosultságot az információ visszanyerés mellékágaként. A késedelem fő oka, hogy egy jól működő tartalom alapú szűrő rendszer megalkotása egy bizonyos témában is igen nagy kihívás, hiszen a feladat nem kevesebb, mint "megérteni" a vizsgálat tárgyát, és a felhasználók hozzá fűződő viszonyát befolyásoló tényezőket. Az egyik első és igen sikeres kutatás e témában a Music Genome Project 1999-ben, melynek célja a zene "megértése" és megragadása tulajdonságain keresztül. Ennek érdekében több mint 450 ilyen tulajdonságot tártak fel, és írták le azok viszonyát algoritmus segítségével. Az eljárás lényege, hogy amennyiben a felhasználó kedvel egy adott dalt, akkor annak adott tulajdonságaihoz (úgy mint stílus, korszak, előadó, hangszerelés, ütem, stb.) a rendszer pozitív értékeket rendel. A hasonló tulajdonságokkal bíró dalok ezután szintén előrébb lesznek sorolva a preferencia listán, és a felhasználó figyelmébe ajánlják. Hatalmas előnye a kollaboratív szűréssel szemben, hogy igen kevés információ is elég az indulásnál, míg az előbbinél sajnos igen sok felhasználó és sok visszajelzés szükséges, hogy hasonló ízlésvilágú embereket tudjon a rendszer azonosítani. Hátránya azonban, hogy jellemzően nehezen, vagy nem tud olyan ajánlásokat tenni, amelyek a felhasználó által hallgatott zenék köréből kivezetne, hiszen nem alapoz felhasználók közötti hasonlóságra, csak a zene, mint entitás tulajdonságainak "megértésére". A 250 millió felhasználót számláló Pandora Internet Radio működése ezen a projekten alapszik mindmáig [51].

Az első olyan megoldás, mely ötvözte a kollaboratív- és a tartalom alapú szűrési megoldásokat, az 1994-ben bemutatott, stanfordi diákok által fejlesztett Fab [52]. Kiemelik, hogy a hibrid rendszerrel az a céljuk, hogy a kétféle eljárás addigra ismertté vált hátrányait kiküszöböljék. Modelljük két alapvető folyamatból áll: először specifikus témákhoz gyűjtenek tartalmakat (például weboldalakat vagy cikkeket pénzügyi témában), majd minden adott felhasználó számára kiválogatják az egyes témakörökből azokat a begyűjtött elemeket, melyek speciálisan őt nagy valószínűséggel érdeklik, és

végül ezek a tartalmak jutnak el hozzá. A kétféle megközelítés ötvözése igen sokféle módon képzelhető el: beágyazható az egyik eljárás a másikba, ahogyan a Fab példáján láthattuk, vagy lehetséges egy közös ajánlást adni a két eljárás eredőjeként, ahogyan a Netflix teszi. A Netflix algoritmusa, a CineMatch volt a 2000-es évek elejének legsikeresebb ajánlórendszere az online film eladások területén. Igen komoly katalizátora volt az ezirányú kutatásoknak, és rohamos fejlődésnek indult az a tudományterület, mely - mint láthattuk - csak a 90-es években kapott önálló létjogosultságot. A 2006-os Netflix-díj (Netflix prize) kihívása volt, hogy az általuk elérhetővé tett 100 millió filmes értékelés alapján olyan ajánló algoritmust kellett készíteni, mely legalább 10%-kal jobb ajánlásokat tesz, mint a CineMatch eredményei. Az 1 millió dolláros fődíjat 2009-ben egy olyan megoldásért ítélték oda, mely 107 különböző algoritmust foglalt magában, és keverte azok ajánlásait a körülmények függvényében [44]. Nem hagyhatjuk ki a sorból az online ajánlórendszerek ma létező legnagyobb példáját, az amazon.com-ot, mely kollaboratív filterezési technika alapján ajánl a felhasználónak termékeket, figyelembe véve a korábban böngészett és megvásárolt termékeket, valamint azt, amit jelenleg éppen megszemlél. Ezt a technikát megannyi internetes bolt használja ma már annak érdekében, hogy eladási mutatóikat javítsák. A Gravity R&D magyar kutatócsapata, akik egyébiránt a Netflix Prize világversenyén a 2. helyen végeztek megoldásukkal, jól megfogalmazták az ajánlórendszerek lényegét: “Rendszerünk úgy működik, mintha egy hagyományos áruházban a vásárlókat jól ismerő eladók mindenki számára máshogy rendeznék el a kirakatot.” [53]

Az ajánlórendszerek mára széles körben elterjedtek és az információáradattól fuldokló felhasználók körében még akkor is nagy népszerűségnek örvendenek, ha sokan tudják, csak egy újabb terméket szándékoznak eladni nekik. Ezen megoldások sikeressége azonban vitathatatlan, és visszavonhatatlanul életünk részévé vált, gondoljunk csak a Youtube-ra vagy a Facebookra [54].

3.5. Lehetséges megközelítések

Formalizálva az ajánló rendszerek feladatát, legyen U a felhasználók halmaza és I a lehetséges termékek halmaza. A felhasználó preferenciáit leírhatjuk úgy, ha megadjuk minden termékhez rendelt értékelését, mely lehet bináris érték (tetszik - nem tetszik, vagy megveszi - nem veszi meg), de leggyakrabban egy valós számmal jelölt érték. Legyen R a lehetséges értékelések halmaza, ekkor tehát a felhasználó hasznossága leírható az alábbi módon: $Pr : U \times I \rightarrow R$. Az $U \times I$ mátrixba rendezett alakját szokás felhasználó-termék mátrixnak is nevezni (user-item matrix), melynek $r_{i,j}$ eleme az i-edik felhasználó j-edik termékre vonatkozó értékelését tartalmazza (ha nem adott a

felhasználó értékelést az adott termékre, azt 0-val jelöljük). Feladatunk olyan i^*_u terméket ajánlani minden u felhasználónak, hogy az maximalizálja a hasznosságát, vagyis

$$\forall u \in U : i_u^* = argmax_{i \in I} Pr\,(u, i)$$

Az ajánlórendszerek témakörének középpontjában az a probléma áll, hogy az U × I tér elemei hiányosan adottak, így az egyes felhasználókhoz tartozó értékelés vektor hiányzó elemeit extrapolálni kell a felhasználó-termék mátrix ismert elemeinek segítségével.
A felhasználói igények feltárásának több merőben eltérő koncepciója alakult ki. Az alábbiakban ezeket a módszereket mutatjuk be néhány példával alátámasztva és figyelmet fordítva a témával kapcsolatban megjelent szakirodalmi áttekintésre is.

3.5.1 A kollaboratív szűrésről általában

Akik korábban hasonló dolgokat kedveltek, azok a jövőben is hasonló dolgokat fognak kedvelni [44]. Talán így foglamazható meg legegyszerűbben a kollaboratív szűrés felhasználókkal kapcsolatos alapgondolata. A módszer a felhasználókat igyekszik a cselekedeteik alapján profilozni, majd az így kialakított profilok közötti hasonlóságot ragadják meg különféle eszközökkel. Egy adott felhasználó által kedvelt elemeket ezután ajánlja a rendszer a hasonlóként azonosított társainak. A megoldás hatalmas előnye, hogy nem igényli az ajánlandó termék megértését, hogy ajánlásokat tegyünk a felhasználók számára, hiszen teljes mértékben a felhasználók megértésére épít a rendszer. Egyik komoly hátránya azonban, hogy a felhasználók viselkedésével kapcsolatos - viszonylag nagy mennyiségű - kezdeti információ hiányában a rendszer működésképtelen. Ezt nevezi a szakirodalom a “hideg indulás” problémájának. Rong et al. [56] Monte Carlo algoritmust javasol a kezdeti információhiány áthidalására, melyet hatékonyan alkalmaznak a felhasználók hasonlóságának előkalkulálására és majdani értékeléseik megjóslására. Egy másik bevált módszer a felhasználók demográfiai adatainak felhasználása a javaslattétel során. A demográfiai alapú szűrést gyakran önálló eljárásként szokták besorolni, lásd [44], én mégis inkább a kollaboratív szűrés egyik aleseteként azonosítanám tekintve, hogy ez is a felhasználók közötti hasonlóságok feltárásán alapszik azzal a különbséggel, hogy itt az alapfeltevés inkább az, hogy a hasonló demográfiai karekterű emberek (kor, nem, iskolázottság, stb.) hasonló érdeklődéssel bírnak. Erre a feltevésre építeni önmagában nyilván igen kevés sikerrel kecsegtet, és sokkal inkább a kényszer - pontosabban a kezdeti információ hiánya - szüli a megoldást, ám kiegészítő információként hibrid

ajánlórendszerekben szignifikáns javulást tud eredményezni. Hasonló következtetésre jut Santos et al. [57], mikor hibrid rendszerekben vizsgálták, hogy adott érdeklődési körrel bíró csoportok között hogyan lehet a mindkét csoportba tartozó felhasználók által okozott zavarokat kiküszöbölni.

A kezdetben fellépő információs hiányt természetesen explicit (kedvencek megjelölése, keresések, választás 2 elem között, elemek rangsorolása, elemek értékelése, stb.) és implicit módon (a felhasználó szociális hálózatának elemzése, egér hőtérkép, az egyes elemek megtekintésével töltött idő, korábban megtekintett vagy akár megvásárolt elemek listája) is igyekeznek mihamarabb csökkenteni. Még a fenti technikák ellenére is igen nehéz némely területen jól működő ajánlórendszer építése, amennyiben - a felhasználók számához viszonyítva - igen nagy változatosságú termékek (elemek) piacáról beszélünk, hiszen így egy-egy terméktípus nagyon kevés értékelést kap, mely rontja az ajánlások pontosságát, illetve sok termék nem kap értékelést, és így nem is kerül majd felhasználóknak tett javaslatok listájára. Az elmondottak alapján világos, hogy szofisztikált ajánlórendszerek kialakítása (súlyosbítva a magas felhasználó- és termékszámmal) komoly kihívást jelent a jelenlegi számítógépek számítási kapacitása mellett.

3.5.2. Kollaboratív szűrés - Memória alapú megoldások

A felhasználók hasonlóságának meghatározására megannyi megoldás született az elmúlt évtizedekben. Ezek közül az egyik legelterjedtebb megoldás-család a memória alapú kollaboratív szűrés. Ezek közös vonása, hogy minden esetben hasonló felhasználókat (illetve egyes esetekben termékeket, lásd később) igyekszik keresni, majd egy aggregációs eljárással kalkulálja az ajánlandó termékek listáját, melyet a hasonlóként klasszifikált felhasználók kedveltek. Meghatározó tehát az a hasonlóságot mérő számítás, ami alapján a hasonló felhaszálók listája meghatározásra kerül. Mivel az esetek döntő többségében nem csak azt vesszük figyelembe az aggregáció során, hogy melyik k db felhasználó volt leginkább hasonló a vizsgált személyhez, hanem a hasonlóság mértékét is számításba vesszük, így ezeket a hasonlósági mérőszámokat gyakran súlyokként veszik figyelembe a szakirodalomban. A súlyok számítására az alábbi módszereket használják legtöbb alkalommal:

- Korrelációs mutatókat gyakran alkalmazzák a hasonlóság megragadására ajánlórendszerek esetében, azok közül is a Pearson-féle korrelációs mutató fordul elő leggyakrabban a szakirodalomban. Két igen korai ajánlórendszer, a Usenet [70] és a zenei ajánló aficionados [64] is ezt használta a hasonlóság mérésére. Az u és v felhasználók közötti hasonlósági súly Pearson-féle kalkulációja az alábbiak szerint alakul:

$$w_{u,v} = \frac{\sum_{i \in I}(r_{u,i} - r_u^*)(r_{v,i} - r_v^*)}{\sqrt{\sum_{i \in I}(r_{u,i} - r_u^*)^2}\sqrt{\sum_{i \in I}(r_{v,i} - r_v^*)^2}}$$

ahol I azon termékek halmaza, melyet u és v felhasználók is értékeltek, továbbá r_u* az u felhasználó által I halmazba tartozó elemekre vonatkozó átlagos értékelése. Hasonló módon számítható még két felhasználó közötti hasonlóság Pearson-féle korrelációs mutatóval, ahol viszont az átlagos r_u* helyett a mediánnal számolnak (ennek szakirodalmi megnevezése: constrained Pearson-correlation). Számítható még Spearman-féle rangkorrelációs mutató az előzőekhez hasonlóan, csak itt az értékelések sorrendiséget jelentenek, akárcsak a Kendall-féle tau-korrelációs mutatónál, ahol Spearmantől eltérően a relatív sorrendek szerepelnek [58].

- A **Jaccard-index** egy általános hasonlósági mutató [62], mely két halmaz hasonlóságát a metszetük és uniojuk arányával méri. Amennyiben két felhasználó hasonlóságát kívánjuk mérni és csak a korábbi vásárlásaik története áll rendelkezésünkre, akkor ez egy jó mérőszám lehet. Ha azonban ismert az egyes termékekre vonatkozó értékelésük is, akkor a Jaccard-index számítása során hasznos információt vesztenénk, így ekkor más mutatók használata javasolt. Legyen az u és v felhasználók által megvásárolt termékek halmaza rendre I_u és I_v, ekkor a két felhasználó Jaccard-indexe:

$$J(u,v) = \frac{|I_u \bigcap I_v|}{|I_u \bigcup I_v|}$$

- **Vektor-cosinus** alapú hasonlóság a két felhasználó azonos termékekre vonatkozó értékeléseiből összeállított vektor által bezárt szöggel ragadja meg a felhasználók ízlésvilágának hasonlóságát. Minél kisebb a két vektor által bezárt szög, annál inkább hasonlóak. Legyen tehát ismét a hasonló elemek halmaza *I*, és a rájuk vonatkozó értékelésvektor rendre *u* és *v*. Ekkor a két vektor által bezárt szög cosinusa:

$$w_{u,v} = cos(\vec{u},\vec{v}) = \frac{\vec{u} \odot \vec{v}}{||\vec{u}||\,||\vec{v}||} = \frac{\sum_{i \in I}(u_i v_i)}{\sqrt{\sum_{i \in I} u_i^2}\sqrt{\sum_{i \in I} v_i^2}}$$

Ennek azonban komoly hibája, hogy amennyiben a felhasználók nem azonos skálán pontoznak, ez a megoldás nem alkalmazható. Ilyenkor szokás a kalkulációt kiigazítani a felhasználók által adott átlagos értékeléssel, amivel visszakapjuk a Pearson-féle korrelációs mutatót. Az eljárás gyakorlati alkalmazására jó példát láthatunk Sarwar et al. cikkében [65].

- **K-legközelebbi szomszéd algoritmussal** (k-Nearest Neighbor, vagy röviden k-NN) is meghatározhatjuk az ajánlás során figyelembe vehető felhasználók körét. Ez az eljárás a gépi tanulásban (machine learning) ismert algoritmusok közül talán a legalapvetőbb. Az alkalmazott

távolság metrikától függően meghatározza a kiválasztott elemhez legközelebb eső k darab további elemet. A kiválasztott elemekhez, a korábbi eljárásokhoz hasonlóan, ismét rendelhetünk súlyokat, melyek gyakran megegyeznek a vizsgált elemtől mért távolság reciprokával ($1/d_i$). Ennek eredményessége azonban hangsúlyozottan függ a használt távolsági metrikától, mely szélsőséges esetben megtalálhatja a "legjobb barátunkat", de akár geográfiai alapon a legközelebbi lakó szomszédunkat is. Mindig érdemes az adott szituációban átgondolni, kinek a véleménye számít jobban egy adott kérdésben. Amennyiben hangfal rendszert szeretnénk vásárolni, kit kérdezzünk meg: a legjobb barátunkat, vagy a szomszédunkat? Ha jóbarátunk nagy házban lakik, úgy könnyen lehet, hogy olyan javaslatot tesz, amivel elégedetlenek leszünk kis lakásunkban, míg a szomszédunk, aki hasonló körülmények között él, mint mi, vélhetően jobb javaslatot tesz hangtechnika ügyében, ahol nem a szakértelmére, hanem a (hasonló körülmények közöti) tapasztalatára számítunk. (Attól most tekintsünk el, hogy egyébként ellenérdekelt abban, hogy komoly hangtechnikai eszközt használjanak a szomszédjában.) Ha azonban egy új filmről akarjuk eldönteni, hogy megnézzük-e, érdemesebb közeli barátainkra hallgatni, hisz ők remélhetőleg tudják, mi fog tetszeni nekünk. A k-NN algoritmusok ajánlórendszerek terén való felhasználásáról bővebben Rashid et al. cikkében olvashatunk [59].

- Az **inverz felhasználói gyakoriság** szerinti súlyozása az értékeléseknek azon a feltevésen alapszik, hogy a széles körben kedvelt termékek kevésbé alkalmasak a hasonlóság megragadására. Ennek érdekében a súlyokat az alábbiak szerint kalkulálja: $w_j=log(n/n_j)$, ahol n az összes felhasználó száma, míg n_j azon felhasználók száma, akik értékelték a j-edik terméket. Így ha egy terméket mindenki értékelt, akkor w_j értéke 0 lesz. A vektor-cosinus módszer jól alkalmazható ilyen súlyokkal, erre láthatunk példát Breese et al. cikkében [60].
- A memória alapú eljárások között is szép számmal találunk **klaszterezésen** alapuló algoritmusokat. Chee et al. [69] által javasolt *RecTree* algoritmus a skálázhatóság problémáját az "oszd meg és uralkodj" elvével kívánta megoldani: első lépésként klaszterekbe sorolja a felhasználókat (k-közép klaszterező eljárárssal), majd a már kisebb csoportok közül csupán a relevánsakkal foglalkozva egy újabb klaszterezési lépésben választja ki a leginkább hasonló felhasználókat, akiknek az értékelései alapján ajánlásokat tesz. Az eljárás a 2000-es évek elején mind futási időben, mint pontosságban felülmúlta a többi memória alapú kollaboratív szűrési megoldást.

A fentiek valamelyikével számolt súlyokat tudjuk a második lépésben felhasználni a vizsgált személy várható értékeléseinek kiszámítására. Természetesen a várhatóan legmagasabbra értékelt

darabok kerülnek majd ajánlásra. A várható értékelések kiszámítására az alábbi főbb technikákat használja a szakirodalom:

- Az értékelések súlyozott átlagát kalkulálhatjuk úgy, mint a vizsgált személy átlagos értékelése korrigálva a mások értékelésétől vett eltérések súlyozott átlagával. Adott tehát az *i*-edik értékelendő termék, amelyett nem csak a vizsgált személy, de az *U* halmazba tartozó összes felhasználó értékelt, ezek rendre $r_{i,u}$, míg átlagos értékelésük r_u^* valmint a vizsgált személy esetén ugyanezt jelölje r_v^*, továbbá legyen u és v felhasználók közötti hasonlóság $w_{u,v}$. Ekkor a *v* személy *i* termékre vonatkozó várható értékelése:

$$R_{v,i} = r_v^* + \frac{\sum_{u \in U}(r_{u,i} - r_u^*)w_{v,u}}{\sum_{u \in U}|w_{v,u}|}$$

- Ismert a fenti kalkulációnak egy egyszerűbb változata, mely a hasonló felhasználók értékeléseinek hasonlósági mértékkel súlyozott átlagával számol (jó példa ezek alkalmazására Herlocker et. al [63]):

$$R_{v,i} = \frac{\sum_{u \in U} w_{v,u} r_{u,i}}{\sum_{u \in U}|w_{v,u}|}$$

- Lehetséges a fenti kalkulációt normalizálni a felhasználók értékeléseire vonakozó szórásokkal, ezzel korrigálva az egyes felhasználók közötti különbségeket.

$$R_{v,i} = r_v^* + \sigma_v \frac{\sum_{u \in U}[w_{v,u}(r_{u,i} - r_u^*)]/\sigma_u}{\sum_{u \in U}|w_{v,u}|}$$

- Természetesen esetenként ennél egyszerűbb kalkulációt is választhatunk, Shardanand et al. [64] például a hasonló felhasználók értékeléseinek súlyozás nélküli átlagát javasolja a Ringo rendszer kapcsán:

$$R_{v,i} = \frac{\sum_{u \in U} r_{u,i}}{|U|}$$

- Az **N-legjobb ajánlat** (top-N recommendations) önmagában megtévesztő, hiszen egyrészt arra utal, hogy a már valamilyen elvek alapján sorba rendezett ajánlások közül az N-legjobbat javasolja a felhasználónak, ám a sorbarendezési eljárások egy családját is így nevezik. Ilyen például a **helyérzékeny hasítási eljárás** (Locality-sensitive hashing, vagy röviden LSH), mely adatdimenzió csökkentésével igyekszik megtalálni a hasonló felhasználókat, és azokat azonos

csoportba sorolni, szintén a k-NN algoritmust implementálva. Funkcióját tekintve leginkább a klaszterezési eljárásokhoz hasonlítható (bővebben lásd [61]).

A memória alapú eljárások csak igen kis részét fedi le a fenti összefoglaló, jobbára a leggyakrabban használt technikákra szorítkoztam, de még így is akad olyan, mi említést érdemel. Ilyen például a **súlyozott többségi alapú előrejelzés** (Weighted Majority Prediction) [94], a **súly felerősítés** (Case Amplification) [95], vagy a **feltöltéssel javított algoritmus** (Imputation Boosted algorithm) [60].

Fent tárgyalt megoldások komoly előnye, hogy nem kell a termékeket ismerni és megérteni ahhoz, hogy ajánlásokat adjunk, éppen ezért könnyen implementálhatóak, és a termékek vagy felhasználók addicionálisan könnyen beilleszthetőek a rendszerbe, azonban túlzottan növelve azok számát a fenti megoldások túlzottan számításigényessé és kényelmetlenné válnak. Ezen túlmenően, mint láttuk, meg kell küzdeni az adatok hiányával, a hideg kezdés problémájával, valamint azzal, hogy a felhasználók által adott értékelésekre a rendszer igen érzékeny.

3.5.3. Kollaboratív szűrés - Modell alapú megoldások

A modell alapú kollaboratív szűrési eljárások alapvető közös vonása, hogy céljuk olyan modellek építése, mely a tanuló adathalmazon akár összetett mintákat is azonosítani képes, feltárva rejtett faktorokat, mely befolyásolja a felhasználók értékeléseit. Az így kialakított modell feladata minél pontosabban megjósolni, mi lesz egy adott felhasználó számára érdekes termék. Az alábbiakban néhány ismertebb modell kerül bemutatásra.

- **Klaszter modellek**: a klaszterezési eljárások lényege olyan csoportok képzése (és csoportképzési ismérvek feltárása), melyek esetén a csoporton belül található elemek a lehető legjobban hasonlítanak egymásra, míg a lehető legnagyobb mértékben eltérnek más csoportok elemeitől. Az egyes módszerek jobbára abban térnek el, hogyan értelmezik ezt a hasonlóságot. Az egyik legáltalánosabb mérőszám erre a már megismert Pearson-féle korrálációs mutató, vagy a **Minkowski-távolság** (lásd M. Deza [66, p. 69.]) Legyen $u=(u_1,u_2,\ldots,u_n)$ *és* $v=(v_1,v_2,\ldots,v_n)$ két felhasználó ízlésvilágát leíró értékelésvektor. Ekkor a kettejük közötti Minkowski-távolságot az alábbiak szerint kalkuláljuk:

$$d(u,v) = \sqrt[p]{\sum_{i=1}^{n} |u_i - v_i|^p}$$

A $p=1$ érték választás mellett visszakapjuk a Manhattan-távolságot, míg $p=2$-re az Euklideszi távolsághoz jutunk. A klaszterezési eljárások legtöbbször csak a teljes sokaság csoportokra osztásának egy hatékony eszközeként használatosak, majd az így képzett csoportokon további

hasonlóság kereső eljárásokat folytatnak. Éppen ezért igen nagy előnye a klaszterező eljárásoknak a jó skálázhatóságuk, azonban ez legtöbbször a pontosságuk rovására megy. Megoldást jelenthet kisebb, jól együtt mozgó klaszterek kialakítása, melyeknek növekvő száma esetén azonban a következő lépésben végrehajtandó számítások lesznek túl nagy számításigényűek. Alapvetően 3 klaszterezési eljárást különböztetünk meg:

1. A hierarchikus klaszterezésen alapuló megoldások gyakran használatosak igen nagy adathalmazok eseten. Jó példa erre Zhang et al. [67] **BIRCH algoritmusa** (Balanced Iterative Reducing and Clustering using Hierarchies), mely az első olyan klaszterező megoldás volt, mely a zajt is kezelni tudta.
2. A daraboló eljárások közül legelterjedtebb a k-közép klaszterezés (k-means clustering) könnyű implementálhatósága és relatív hatékonysága miatt. Elsőként MacQueen [68] mutatott be **daraboló klaszterezésen** alapuló ajánló megoldást.
3. A **sűrűség alapú klaszterező eljárások** sűrű csomópontokat keresnek és igyekeznek elválasztani a ritkábban elhelyezkedő pontoktól, melyet zajnak tekintenek. Erre jó példát látunk Ester et al. [70] **DBSCAN algoritmusában**, mely tetszőleges alakú klaszterek azonosítására alkalmas nagy hatékonysággal.

A **rugalmas keverék modell** (Flexible Mixture Model) [71] több fenti klaszterező eljárást is alkalmazva egyszerre klaszterezi a felhasználókat és a termékeket, továbbá megengedi, hogy azok egyszerre több klaszterbe is kerülhessenek. Tesztadataikon hatékonyabbnak bizonyult a modelljük, mint a Pearson-féle korreláción alapuló szűrési eljárás.

- **Regressziós modellt** használnak igen gyakran olyan esetekben, mikor rendelkezésre áll az felhasználók értékeléseire vonatkozó információ, mely alapján következtetni szeretnénk még ismeretlen értékelésekre. Legyen R a felhasználó-termék n×m-es mátrix, vagyis $r_{i,j}$ az i-edik felhasználó j-edik termékre vonatkozó értékelése, $X=(X_1,X_2,\ldots,X_n)$ random változók, melyek a felhasználók termékekre vonatkozó preferenciáit írják le, $E=(E_1,E_2,\ldots,E_n)$ a felhasználók értékeléseiben észlelhető zaj. Ekkor a modell $R=\Delta X+E$ egyenlete ad becslést a Δ $n\times k$ méretű mátrix paraméterértékeivel, így közelítve R-t. Mivel azonban az esetek többségében az R mátrixban található értékelések elég ritkák, így szükséges azok értékét helyettesíteni. Canny [72] javasolja a feltöltést a mátrix sorainak vagy oszlopainak átlagából kalkulált értékekkel. Eredményeik alapján a megoldás sikeresebb, mint a **mátrixfaktorizációs** (SVD) eljárások. Vucetic and Obradovic [73] termékek közötti hasonlóságok alapján OLS becsléssel kapott lineáris regressziós modellekkel kalkulálja a felhasználók várható értékeléseit, hatékonyan oldva meg ezzel a ritka értékelések problémáját. A regressziós modellek használatakor fellépő igen nagy

számításigényből adódó problémára jó választ ad Lemire et al. [74] **"slope one"** eljárása, melyről bővebben a termék alapú szűrési eljárások alfejezetben olvashatunk.

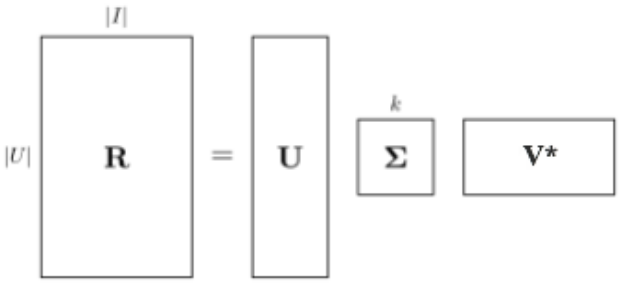
- **Bayes-i hálókon** alapuló algoritmusokat gyakran használnak klasszifikációs problémák megoldására. Legyen $\langle N, A, \Theta \rangle$ irányított körmentes gráf, ahol minden $n \in N$ csúcs egy véletlen változót jelöl, $a \in A$ irányított élek valószínűségi kapcsolatokat jelölnek a csúcsok között, Θ pedig valószínűségi mátrix, mely meghatározza, hogy az egyes csúcsok mennyire függnek azoktól a szomszédaiktól, ahonnan feléjük irányuló él fut. Az egyszerű Bayes-i algoritmus a naív Bayes-i stratégiára építve tesz ajánlásokat. Feltételezi, hogy a felhasználócsoportokon belül az egyes tulajdonságok (dimenziók) függetlenek, így kalkulálja minden csoportra azt a valószínűségi értéket, melyhez ez esetben a hasonlóságot mérjük, és a legnagyobb valószínűségű csoport alapján tesz ajánlásokat. Az eljárás gyakorlati alkalmazásának jó példáját láthatjuk Miyahara et al. cikkében [76]. Nagy, de hiányos értékelési adathalmaz esetén alkalmazzák az **NB-ELR** és a **TAN-ELR** (naïve Bayes, tree augmented naïve Bayes) algoritmusokat, melyek paramétereit **ELR** (extended logistic regression) regresszióval optimalizálják [77]. Mindkét megoldásnak jó klasszifikáló ereje van és jobban teljesít, mint az egyszerű Bayes-i algoritmus. A téma kimerítő összefoglalóját adja Pearl [75].
- A **látens (vagy mögöttes) szemantikai eljárások** (Latent semantic models) alapja, hogy egymást átfedő felhasználói csoportok képzésével rejtett változókat tár fel (akár a főkomponenselemzésnél szokás), melyekkel a felhasználók ízlésvilága leírható. A Hofmann és Puziha cikkében [78] leírt modell a feltárt értékelést befolyásoló faktorok konvex kombinációjaként írja le az egyes felhasználók értékeléseit. Előnye a korábban tárgyalt memória alapú megoldásokkal szemben a nagyobb pontossága és skálázhatósága nagy adathalmazokon. A téma jó összefoglalóját adja Symeonidis et al. [79].
- **Markov-döntési folyamatok** alkalmazása mögött az a gondolat áll, hogy a felhasználók értékeléseit nem csak előrejelzési problémaként lehet felfogni, hanem szekvenciális optimalizálási feladatként is. A Markov-féle döntési folyamatok során feltesszük, hogy a felhasználó döntései részben véletlenszerűek, részben az ő kontrollja alatt állnak [80]. A feladat definiálható a $\langle S, A, R, Pr \rangle$ négy elemmel, ahol S a lehetséges világállapotok halmaza, A a lehetséges akciók halmaza, R a kifizetésmátrix, melynek $r_{i,j}$ eleme jelöli az i-edik világállapotban j-edik akcióhoz tartozó kifizetést, míg Pr az átmenetvalószínűség mátrix, melynek $Pr_{i,j}$ eleme jelöli, hogy mennyi a valószínűsége az i-edik világállapotból a j-edikbe kerülni. A feladat tehát a kifizetések sorozatának maximalizálása az akciók sorozatán keresztül. Ez könnyen értelmezhető az

ajánlórendszerekre is Shani et al. munkája alapján [81]: az állapotok olyan n-elemű vektorok, melyek értéke 1, ha az adott termék az állapot során jelen van és 0, ha nincs. Az akciókat az egyes termékek ajánlásai jelentik. Ezután a lehetséges állapotok: felhasználó megveszi az ajánlott terméket, megvesz egy nem ajánlott terméket, vagy nem vesz semmit. A kifizetés az adott termék eladását jelenti. Feltesszük, hogy annak valószínűsége, hogy vásárol, függ az adott terméktől, attól, hogy ajánlották-e és a felhasználó pillanatnyi állapotától, de a többi terméktől nem függ. Az eljárást implementálták egy izraeli internetes könyvesboltban, melynek a forgalma ennek hatására szignifikánsan megemelkedett.

- **Mátrix faktorizációs** eljárásokat gyakorta alkalmaznak az ajánlórendszerek esetében, hiszen mind felhasználót, mind terméket igen nagy számban találunk egy-egy gyakorlati megvalósulás alkalmával. Így felhasználó-termék mátrix, mely a felhasználók értékeléseit tartalmazza, egyrészt nagy dimenziószámú, másrészt sok eleme hiányzik, tehát redundanciával állunk szemben. Mivel azonban igen sok felhasználó ízlésvilága hasonló, és sok termék is hasonlít egymásra, így természetesen adódik, hogy csoportokba próbáljuk sorolni őket, és amennyiben a termékeket és felhasználókat egy vektortérben értelmezzük, így annak dimenziószámát szeretnénk csökkenteni úgy, hogy modellünk magyarázó ereje minél kevésbé csökkenjen. Tehát például a termékek esetében szeretnénk találni k db olyan faktort (akár rejtett ismérvet), mely alapján minden elem leírható az ezen faktorok iránti érdeklődés(ek kombinációja) és az érdeklődés mértéke segítségével. Látható, hogy ez az eljárás szoros kapcsolatban van a látens szemantikai modellekkel, sőt fel is használja a mátrixfaktorizációs technikát. Az eljárás első alkalmazója Billsus és Pazzani voltak [82], akik a filmajánlók terén használták igen eredményesen a **szinguláris érték felbontást** (Singular Value Decomposition, röviden SVD) [83], és határoztak meg néhány olyan faktort, amik kombinációjával a filmek leírhatóak voltak viszonylag kevés dimenzió segítségével. Az SVD technika alkalmazására egy másik jó példa Sarwar et al. [84], aki egy internetes bolt adatain tesztelte a mátrixfaktorizáció hatékonyságát és pontosságát más kollaboratív szűrésen alapuló eljárásokkal és azt találta, hogy kellően sűrű értékelések esetén pontosabb, mint az eljárások többsége, de ritka adatok esetén konzekvensen rosszabb. Az SVD generálása igen költséges, ám nagy előnye, hogy ez a művelet a háttérben (offline) elvégezhető, így az internetes alkalmazásoknál igen jó teljesítménnyel működik, hiszen kevés dimenzióval kell dolgoznia, és csak néhány elemi műveletet hajt végre. Jelölje R az $n \times m$-es felhasználó-termék mátrixot az értékelésekkel ($|U|=n$ és $|I|=m$), U az R mátrix oszlopainak szinguláris vektoraiból összeállított n×k méretű mátrix, V^* a $k \times m$-es V mátrix transzponáltja, ahol V az R mátrix sorainak

szinguláris vektoraiból összeállított mátrix, Σ pedig egy $k \times k$ méretű háromszögmátrix, melynek fődiagonálisában az R szinguláris értékei vannak. Ekkor az R mátrix SVD felbontása: $R \sim U \Sigma V^*$, melyet a *14. ábrán* szemléltetünk. A szinguláris értékek mutatják, hogy az eredeti adat varianciájának hány százalékát sikerült a hozzá tartozó szinguláris vektornak megragadni. Az adott példában az első k legnagyobb szinguláris értéket tartottuk meg, a többi értékének 0-t választottunk, ezzel csökkentve a dimenziószámot. A k dimenzió most tehát k darab - eddig rejtett - termékfaktort jelöl, és U mátrix sorai az egyes felhasználók ezen faktorokra vonatkozó értékelését/érdeklődését tartalmazzák, míg a V^* oszlopai az egyes termékek relevanciáját jelöli minden faktorhoz. A turisztikai példánál maradva a termékek a meglátogatandó helyszínek, és a faktorok olyan kategóriák, mint a múzeum, sport, templom, zene, stb. Ezek kombinációjaként írhatóak le az egyes helyszínek súlyozva persze az adott helyszín esetében a relevanciájukkal. A Magyar Nemzeti Múzeum például leírható olyan faktorok súlyozott kombinációjaként, mint történelem, múzeum, építészet, kultúra, stb. A szinguláris értékek nem mások, mint preferencia súlyok, melyek megmutatják az egyes feltárt faktorok fontosságát az értékelésekben. Így a felhasználó preferenciája értelmezhető az egyes faktorok iránti érdeklődésének súlyozott összegének és a termék adott faktorban való relevanciájának szorzataként.

14. ábra A mátrixfaktorizációs eljárás



A mátrixfaktorizációs algoritmusok közül Kurucz et al. [85] munkáját érdemes kiemelni, mivel ők jó megoldást adnak arra a problémára is, ha a felhasználó-termék mátrixban sok hiányzó elem van. Legkisebb négyzetek módszerével minden felhasználó értékeléseire regressziót illesztenek az R mátrix elemeinek pótlására. Sarwar et al. [84] eredményei alapján érdemes helyettesíteni a hiányzó értékeket a termékekre vonatkozó értékelések átlagával (tehát az oszlopátlaggal), szemben a felhasználók értékelésének átlagával, ami rosszul teljesített a tesztadatokon. Azt is megállapították

továbbá, hogy amennyiben az értékeléseket már a faktorizáció előtt normalizálják, az növeli az ajánlások pontosságát. Ha egy adott felhasználó értékeléseit szeretnénk becsülni, vagy akár egy új felhasználót illesztenénk a modellbe, akivel korábban nem számoltunk, arra a széles körben alkalmazott behajtogatás (folding-in) eljárás használható [86]. Amennyiben adott az u felhasználó r_u értékelésvektora (ahol a nem ismert értékelések helyére 0 kerül), akkor a preferenciáit leíró vektor számolható az alábbi módon (az SVD eljárás megismétlése nélkül): $u=(\Sigma V^*)^{-1} r_i$. Az u felhasználó preferenciája ebből számítható az alábbi módon $Pr_u=u\Sigma V^*$, és az ajánlás (pl. top-N értékelés módszerét követve) az így kapott vektor N legnagyobb preferencia értékkel bíró eleméhez rendelt termék lesz. A témát átfogóan tárgyalja Ekstrand et al. [87] kollaboratív szűrőkről szóló munkájának 2.4-es fejezete, valamint Tikk et al. [93] mátrixfaktorizációs eljárásokat összehasonlító cikke, melyben az algoritmusok pontosságát és betanítási idejét vetik össze a Netflix Prize adatokon.

További modell alapú megoldások, mint például a **függőségi hálókon** vagy **asszociációs szabályokon** alapuló algoritmusok részletes összefoglalóját adja Su et al. [88].

3.5.4. Kollaboratív szűrés - Termék alapú szűrési eljárások

A hasonló ízlésvilágú felhasználók azonosításának több komoly gyengesége is akad:

- Igen nehéz olyan felhasználói profilt kialakító eljárást tervezni, mely (kevés felhasználói adat lévén) ne változtatná a felhasználói profilt igen gyorsan, ha új információhoz jutunk. A profilok tehát nem stabilak, hiszen jellemzően kevés információ alapján kreáljuk őket.
- Sok termékelem esetén (pl. amazon), termékenként csak nagyon kevés értékelésünk van, de előfordulhat azok teljes hiánya is.

Ezek kiküszöbölésére született a termék alapú szűrési eljárás, melyet elsőként Sarwar et al. [89] használ, ahol előbb a termékek közötti összefüggéseket tárjuk fel, majd kiindulva a felhasználó által értékelt termékekből tudunk javaslatokat tenni olyan termékekre, melyek hasonlóan lettek értékelve, mint a felhasználó által kedvelt darabok. A feladat megoldására egy igen egyszerű és takarékos megoldást javasolt Lemire et al. [74] arra az esetre, ha a termékekre vonatkozó értékelések rendelkezésre állnak. Ahelyett, hogy $f(x)=ax+b$ alakú lineáris regressziót illesztenének az értékelési adatokra termékpáronként, ők $f(x)=x+b$ alakot javasolnak, elkerülendő a túlillesztést és csökkentve az eljárás számításigényét. Megmutatták, hogy az általuk **"slope one"** névre keresztelt eljárás sok esetben eredményesebb, mint a lineáris regresszió. Így tehát két termék viszonyát a korábbi

értékelések kapcsán az átlagos értékelések különbségével írják le. Ha tehát *A* és *B* termék esetén a felhasználók értékelései alapján A termék rendre 2 ponttal többre lett értékelve, akkor egy adott felhasználó, ha *A*-t 3 pontra értékeli, akkor várhatóan *B*-t 1 pontra értékeli majd. Látható, hogy ez esetben a felhasználó individuális preferenciáit csak az *A*-ra tett értékelése erejéig vesszük figyelembe, a többit a "közízlés" alapján kalkuláljuk.

Gyakran azonban nem állnak rendelkezésre termékekre vonatkozó értékelések, csak olyan bináris - és a felhasználó értékítéletéről sokkal nyíltabban árulkodó - változók, mint az, hogy egy adott terméket megvásárolt-e vagy nem. Az amazon erre a célra kifejlesztett **"item-to-item" algoritmusának** alapja, hogy a felhasználókhoz egy-egy vektort rendelünk, melynek j-edik eleme reprezentálja, hogy az adott felhasználó megvette-e a j-edik terméket. Formálisan a felhasználó-termék mátrix a_{ij} eleme 1, ha az i-edik felhasználó megvette a j-edik terméket, és 0, ha nem vette meg. Két termék hasonlóságát a kettejük vektora által bezárt szöggel ragadják meg (mint azt korábban a vektor-cosicus eljárásnál láttuk), vagyis minél inkabb hasonlít két termék (megítéslése) egymáshoz, a vektoraik által bezárt szög annál kisebb, vagyis cosinusuk értéke 1-hez közeli. Így ha az amazonon megtekintünk egy könyvet, akkor az oldal figyelmünkbe ajánlja azt a másik 3 könyvet, melyek vektorai a lehető legkisebb szöget zárják be a megtekintett könyvet leíró vektorral [90].

A súlyok számítására a termék alapú szűrés esetén egy speciális lehetőség is adódik: Karypis [91] javasolja egy feltételes valószínűség alapú súlyozás bevezetését a vásárlástörténet alapján. Jelölje *B* a korábban megvásárolt termékek halmazát, ekkor az *i* és *j* termék hasonlóságát leírhatjuk azzal a feltételes valószínűséggel, hogy *j*-t megveszi valaki, feltéve, hogy *i*-t megvette korábban, azaz $w_{i,j}=Pr_B[j \in B | i \in B]$. Mivel néhány terméket igen sokan megvesznek, így azok sok termékre "hasonlítani" fognak. Ennek kiküszöbölésére Karypis-ék bevezetnek egy csillapító hatású α paramétert, mellyel ellensúlyozható, hogy a feltételes valószínűség értéke magas abból adódóan, hogy a $Pr_B[j \in B]$ értéke magas. Ezzel a paraméterrel kiegészítve a hasonlóság kalkulációja az alábbi alakot ölti:

$$w_{i,j} = \frac{Freq(i \cap j)}{Freq(i)(Freq(j))^{\alpha}}$$

A termék alapú szűrési eljárás során használt algoritmusokról Cacheda et al. [92] összefoglaló cikkében olvashatunk.

3.5.5. Tartalom alapú szűrés

Az ajánlórendszerek ezen osztályának gyökerei az információszűrő és információvisszanyerő rendszerekben keresendőek. Alapfeltevése, hogy amit a felhasználó korábban kedvelt, ahhoz hasonló termékeket a jövőben is kedvelni fog. Ennek érdekében tehát szükséges a termékek megismerhetősége és karekterük megragadása, tömörítése tulajdonságokba, kategóriákba. A gondolatmenet ismerős a mátrixfaktorizácós eljárásokból. Ahogyan ott feltártuk algoritmusok segítségével a termékeket megítélését befolyásoló - rejtett - tulajdonságokat, úgy tárjuk fel itt is, vagy adottnak feltételezzük. Míg a kollaboratív szűrés esetén ez dimenziócsökkentésre szolgáló eljárás volt, addig a tartalom alapú szűrésnél előfeltétel a termékek karakterének leírhatósága. Ez alapvetően megnehezíti a tartalom alapú szűrési eljárások alkalmazását. Mint azt korábban is tárgyaltuk, az 1999-es Music Genome Project során több, mint 400 ilyen tulajdonság került meghatározásra a zene esetében, és a tisztán tartalom alapú szűrést használó Pandora Rádió ezen a zenével kapcsolatos tudáson alapszik [51]. Természetesen sok más területen is sikeresen alkalmazzák a tartalom alapú szűrőket, ilyen például a filmajánló oldalak az IMDb és a Rotten Tomatoes. Fontos tisztázni, hogy mi a különbség a termék alapú szűrés és a tartalom alapú szűrés között: míg az előbbi esetben termékek felhasználók által adott értékelésekből összeállított vektorai közötti hasonlóságot vizsgálja, addig a tartalom alapú szűrés a termékeket tulajdonságok terében ábrázoló vektorok közötti hasonlóságokat keresi.

Az ajánlórendszert leíró modellhez az alábbi elemek szükségesek:

- Termék profil: A termékek karekterét alkotó faktorok, tulajdonságok rendezett sora, melyet leggyakrabban egy vektorral adunk meg, az egyes tulajdonságértékek helyén feltüntetett számérték pedig az adott tulajdonság termékre vonatkozó relevanciáját jelöli, ezt leggyakrabban a **TF-IDF** algoritmussal állítják elő (term frequency–inverse document frequency) [96]. Így egy-egy termék tekinthető a tulajdonságok súlyozott aggregációjának is.
- Felhasználói profil: az egyes felhasználók esetén szintén vektorba rendezzük azt, hogy az egyes tulajdonságok mennyire fontosak a számára (ezzel írva le a felhasználó preferenciáit). Ezt az információt begyűjthetjük a felhasználótól direkt módon, nyilatkoztatva az egyes tulajdonságok fontosságáról, vagy indirekt módon az egyes termékek értékelése kapcsán. Ha értékeli a terméket, akkor a terméket leíró tulajdonságvektor az - értékelést, mint súlyt figyelembe véve - hozzájárul a felhasználó profilvektorához. Íly módon kerülnek felhasználásra, és finomíthat tovább a személyre szabott ajánlás a felhasználó és a rendszer közötti összes interakció feldolgozásával, ami lehet értékelés (pontokban vagy bináris módon - tetszik, nem tetszik), a megtekintések

időtartama, vagy vásárlások alapján. Ilyen finomító eljárás a **Rocchio-klasszifikálás** [97], mely a felhasználói interakciók alapján pontosítja a felhasználói profilt, és az egyes elemekhez azt a csoportot rendeli, amely csoportátlaga a legközelebb van az adott elemhez (vagy alternatívájaként használatos a **Winnow-algoritmus** [99]). Ha például 10-ből 10 pontot adott egy filmre, aminek a tulajdonságai között szerepel a horror, akkor ezt beépítve a profilvektorába bizonyára megerősíti a rendszerben a horrorfilmek iránti érdeklődésének valószínűségét.

- **Hasonlóság kereső algoritmusok**: ha már meghatároztuk a termék- és felhasználói profilokat (mindkettőt a tulajdonságok vektorterében megadva), akkor egy hasonlóságot azonosító algoritmussal megadhatjuk a felhasználó számára azokat az elemeket, melyeket a lehető legvalószínűbb, hogy kedvelni fog.

Az első tartalom alapú szűrő eljárások szövegbányászati eljárásokon alapulnak, melyekkel akár nagyobb dokumentumokat klasszifikáltak, gondoljunk például a korábban ismertetett **Salton-féle dokumentum indexelési eljárásra** [50], vagy a **Syskill & Webert rendszerre** [98], ami a dokumentumokat a 128 legfontosabb szavuk alapján klasszifikálja. Sok eljárás létezik arra, hogy a fontosságot miként mérik, egyike a szavak gyakorisága a szövegben. A már említett **TF-IDF** algoritmus az alábbiak szerint végzi ezt a feladatot: tegyük fel, hogy összesen $|I|$ db dokumentum ajánlható a felhasználóknak, és a k_i-edik kulcsszó összesen n_i db dokumentumban fodul elő, továbbá a d_i dokumentumban $f_{i,j}$ alkalommal található. Ekkor a k_i kulcsszó d_i dokumentumban való kifejezés gyakorisága (term frequency): $TF_{i,j}=f_{i,j}/max(z)f_{z,j}$, ahol a nevezőben a d_j dokumentum maximális gyakoriságú kulcsszavának gyakorisága szerepel. Mivel azonban azok a szavak, melyek túl gyakran fordulnak elő minden dokumentumban, nem segítenek a dokumentumok klasszifikálásában, így a túl gyakori előfordulásukat az inverz dokumentum gyakoriság mutatóval (inverse document frequency) büntetik: $IDF_i=log(|I|/n_i)$, vagyis minél több dokumentumban fordul elő egy kulcsszó, annál kisebb értéket vesz fel az IDF_i. A d_j dokumentumban k_i kulcsszó súlyát a $w_{i,j}=TF_{i,j}\times IDF_i$ határozza meg, és ezekből állítjuk össze a teljes dokumentum tartalmát leíró súlyvektort: $C(d_j)=(w_{1,j}, w_{2,j}, ..., w_{k,j})$. Mivel nem csak a termékek (a példában dokumentumok) vannak ábrázolva a tulajdonságok (esetünkben kulcsszavak vektorterében) a fent kalkulált súlyokkal, hanem érdeklődésük alapján a felhasználók is, így nem maradt más, mint a felhasználó súlyvektorának összevetése a termékekkel hasonlóság szempontjából. Ennek a legelterjedtebb technikája a tartalom alapú szűrések terén a **vektor-cosinus alapú hasonlóság** keresése, melyet már láthattunk a kollaboratív szűrési eljárások sorában. Természetesen sok más hasonlóság kereső eljárást alkalmaznak a szakirodalomban, melyek nem egy heurisztikus hasonlósági mértéket alkalmaznak, sokkal inkább modell alapúak, úgy mint a döntési fák, bayes-i algoritmusok, neurális hálók,

klaszterezés, (bővebben lásd Duda et al. [100]). A **bayes-i algoritmusok** egyik nagy hibája véleményem szerint, hogy feltételezik az egyes kulcsszavak függetlenségét, holott igen gyakran a kulcsszavaknak sokkal inkább egy kombinációjának jelenléte utal egy adott témára (jó példa erre Pazzani és Billsus cikke [98]). A **neurális hálók** igen hatékonynak bizonyultak a tartalmat meghatározó látens faktorok feltárásában, van den Oord et al. [101] például a hangelemzéseknél használt **konvolúciós neurális hálót** (convolutionary neural networks [261]) használt zenei tartalmak ajánlására, mellyel jobb eredményt értek el a szokásos **szöveg alapú** elemzéseknél (bag-of words). A **döntési fák** alkalmazhatósága e területen igen korlátozott, mivel nagy adathalmazon túl nagy a számításigénye: minden felhasználóra és minden termékre fel kell építeni a fát, és egy adott felhasználó preferált termékeinek a megtalálásához az összes termékre felépített fát be kell futni a gyökértől a levelekig, hogy a valószínűségi súlyokat kalkulálni tudjuk. A témakör jó összefoglalóját adja Gershman et al. [102], míg a Li és Yamada [103] filmes tartalom alapú ajánlórendszeren mutatják be tartalmi faktorok alapján épített döntési fáik működését, ám azok precizitása rosszabb, mintha az átlagos értékelések alapján tettek volna ajánlásokat.

A tartalom alapú szűrők igazi vízválasztója leginkább az, képesek-e az egyes üzleti területek között is átmenetet biztosítani, és például a felhasználók hírolvasási szokásai alapján filmeket ajánlani nekik. A témakörök közötti komplex összefüggéseket ma még igen kevés rendszer tudja megteremteni, nagy pontossággal egyelőre egyikük sem. A tartalom alapú szűrési eljárásokat átfogóan tárgyalja Adomavicius és Tuzhilin cikke [104].

3.5.6. Tudás alapú szűrés

A korábban ismertetett eljárások komoly korlátja, hogy ha kezdetben nem áll rendelkezésre kellő információnk az értékelésekre vonatkozóan, úgy nem tud a rendszer jó ajánlásokat tenni. Olyan termékek esetében, amit ritkán vesznek az emberek, mint például autót vagy ingatlant, ez a probléma hatványozottan jelentkezik. Ezekre a termékekre alkalmazzák a tudás alapú ajánlórendszereket (Knowledge based RS) [110], ahol a felhasználóktól többlet inormációt nyernek az alábbi technikák egyikével:

- **Beszélgető eljárás** (Conversational recommendation): a felhasználó és a rendszer között egy visszacsatolási folyamat során egyre inkább pontosíthatóak a felhasználó preferenciái (korlátok beállítása, korábbi javaslatok értékelése, további szűkítések, stb.) [106].
- **Keresés alapú eljárások** (Search-based recommendation) során eldöntendő vagy kiegészítendő kérdésekkel szűkíti a rendszer a megfelelő termékek halmazát (éppen ezért gyakran nevezik

korlát alapú szűrésnek). Az első ilyen irányba tett lépéseket a Grundy rendszerénél láttuk a ‘70-es évek végén [45]. Kiegészítendő kérdések esetén is döntően alternatívákat kínál a rendszer. A Felfernig et al. [107] által bemutatott RecTurk rendszer például egyszerű feladatokra bontja a felhasználó számára a komplex értékelési feladatokat (pl. egy adott termék tulajdonságait értékelje egyenként). A **korlát alapú rendszerek** jó összefoglalását adja Felfernig és Burke [108].

- **Navigációs eljárások** (Navigation-based recommendation) esetén a felhasználói visszajelzések kritkák formájában jelnnek meg. Ilyenkor a következő ajánlott termékre vonatkozó egy tulajdonságra vonatkozó- vagy komplex változtatást fogalmaz meg a felhasználó, például ugyanilyen tulajdonságokkal rendelkező televíziók közül kérek alacsonyabb árkategóriájút. A gyakorlatban működő ilyen rendszerek kritikai összehasonlítását adja Chen és Pu [109].

A tudás alapú szűrők azonban mindmáig szenvednek a tudás megszerzésének nehézségeitől, illetve annak folyamatos frissítésétől. A téma hasznos és részletes összefoglalóját olvashatjuk Burke cikkében [110].

3.5.7. Hibrid szűrők

A korábban tárgyalt szűrési megoldások számos korláttal küzdenek, így ezek kiküszöbölésére igyekeznek azokat kombinálni az ajánlások során, hogy kihasználjanak bizonyos szinergiákat. Például kollaboratív technikák esetén ismert kezdeti információs nehézségek áthidalhatóak tudás alapú szűrési eljárással, míg a kollaboratív szűrés segítségével hasonló felhasználókat találhatunk, akik segítenek finomítani a javaslatokat, vagy akár olyan ajánlásokat tehetünk ezen keresztül, amire egy tudás alapú megközelítéssel sosem juthattunk volna. Ugyanígy megoldható a stabilitás és formálhatóság problémája, vagyis hogy a rendszer ajánlásait egyrészt ne billentsék ki esetlegesen felmerülő igények, de ne is ragadjon bele tartósan olyasmibe, amit a felhasználó igényei idővel túlhaladtak. Erre adhat megoldást a tudás alapú és a tartalom alapú eljárások kombinálása [60]. Az egyes ajánlórendszerek információs forrásait összefoglalva az 2. *táblázatban* olvashatjuk.

Burke tanulmányában [112] átfogó leírást ad hibrid rendszerekről és 7 stratégiát állapít meg, ahogyan az egyes eljárásokat kombinálni lehet, és az általa vázolt 4 önálló ajánlási eljárásra - kollaboratív (KO), demográfiai (DE), tartalom alapú (TA) és tudás alapú (TU) - összesen 53 lehetséges eljárást ismertet (mivel sok helyen a felhasznált technikák sorrendje is számít, így több megoldás adódik, mint 7×4). A lehetséges megoldások összefoglalása a 3. *táblázatban* látható.

2. táblázat: Ajánlási technikák a felhasznált információk alapján

	Kollaboratív szűrők	Termék alapú szűrők	Demográfiai szűrők	Tartalom alapú szűrők	Tudás alapú szűrők	Hibrid szűrők
Adott felhasználó értékelései	×	×	×	×	×	×
Felhasználó-Termék mátrix	×	×	×	×		×
Demográfiai adatok			×			×
Termék adatbázis				×	×	×
Felhasználói igények, kérdések					×	×
Korlátozások					×	×
Üzleti terület ismerete					×	×

3. táblázat: A hibrid rendszerek lehetséges kombinációi

	Súlyozott	Kevert	Váltogató	Tulajdonság kombináció	Elmélyítő	Tulajdonság kiterjesztés	Közbeékelő
KO/TA	[113]	[115]	[116]	[117]		[118]	
KO/DE							
KO/TU	[114]						
TA/KO							[52],[119]
TA/DE							
TA/TU							
DE/KO							
DE/TA							
DE/TU							
TU/KO					[112]		
TU/TA							
TU/DE							
		Redundáns		Nem létezik		Van ismert példa	

Az alábbiakban áttekintjük a 7 létező hibrid stratégiát:

- **Súlyozott** (Weighted) eljárás során az egyes ajánló technikák párhuzamosan futnak, és az általuk kalkulált megoldásokat numerikusan aggregálják, majd juttatják el a felhasználóhoz végeredményként. Claypool et al. [113] cikkében tartalom alapú és kollaboratív eljárásokat használnak online hírek ajánlására. Fontos mozzanata az eljárásnak annak megállapítása, hogy a két külön ágról érkező ajánlást milyen súlyokkal kombinálva érdemes a felhasználó elé tárni. Például Mobasher et al. [114] szemantikai tudás alapú eljárást ötvözött kollaboratív szűrűvel, és a két eredmény 60-40 arányú lineáris kombinációját véve listázták ki az ajánlott filmeket az érdeklődőknek.
- **Kevert** (Mixed) eljárás esetén a fentivel szemben nem cél az eredmények aggregálása, ilyenkor egyes eljárások alapján kalkulált megoldásokat egyszerre közlik a felhasználóval egy közös listában, így döntésekor ezeket mind figyelembe veheti. A kihívás sokkal inkább az, hogy milyen sorrendben állítsák össze a listát. Ekkor gyakran a kalkulált feltételezett értékelések alapján teszik meg a sorbarendezést. Ezen alapszik például a PTV ajánlórendszere, mely tartalom alapú- és kollaboratív technikákat keverő megoldást alkalmaz [115].
- **Váltogató** eljárás (Switching): A rendszer több ajánló technikával is operál, és bizonyos elvek mentén dönt, hogy melyik eljárás által kalkulált ajánlást osztja meg végül a felhasználóval. A NewsDude rendszere (lásd Billsus és Pazzani [116]) például 3 eljárást próbál ki sorrendben. Ha a tartalom alapú legközelebbi szomszéd eljárás nem sikeres (ezt konfidencia intervallum alkalmazásával döntik el), akkor kollaboratív algoritmust használ, és ha az is sikertelen, akkor végső esetben egy naive-Bayes-i tartalom alapú eljárás ad ajánlást.
- **Tulajdonság kombináló** eljárások (Feature-combining) esetén nem több technikát alkalmazunk, mint a korábbi stratégiáknál láthattuk, hiszen itt végig egy ajánlórendszer van működésben, de az adatok beszerzése más forrásokból is történhet. Például Basu et al. [117] a Ripper nevű filmajánló rendszert tartalom alapon működteti, de a felhasználók értékeléseit kollaboratív technikákkal kapcsolják a film tulajdonságait tartalmazó súlyvektorokhoz.
- **Elmélyítő** eljárás (Cascade): két ajánló technika hierarchikus alkalmazását jelenti. Amennyiben az elsődleges eljárás bizonyos elemekre azonos értékelést ad eredményül, úgy egy másodlagos eljárással, csak a döntetlen elemeken, egyértelmű sorrendet állít fel. Ezt alkalmazta nagy hatékonysággal az eredmények finomítására Burke [112] az Entree étteremajánló oldalnál, melynek elsődleges rendszere tudás alapú ajánlást ad, melyet döntetlen esetén egy kollaboratív megoldás finomít.

- **Tulajdonság kiterjesztő** eljárás (Feature-augmenting) a tulajdonság kombináló eljárással szemben nem csak az adatforrásokat kombinálja, hanem minden ajánlható elem értékeléseit egy másik technikával minden termékre új tulajdonságvektort alkothat, melyet az alapeljárás során, a kibővített információval együtt felhasználunk. Melville et al. [118] a hiányos értékeléseket tartalom alapú szűrési eljárással pótolja, majd ezeket az értékeléseket használják fel a kollaborációs szűrési folyamat során. Tőlük számazik a **tartalommal javított kollaboratív szűrés** (content-boosted collaborative filtering) elnevezés.
- **Közbeékelő** eljárások (Meta-level) esetén az egyik technika outputja lesz a másik technika inputja, például Pazzani [119] étteremajánló rendszerében a felhasználói preferenciákra tartalom alapú modellt illesztett, majd ennek eredményeit egy kollaboratív szűrési eljárás során felhasználva hasonló felhasználókat keresett. Hasonló eljárást használ a már korábban ismertetett Fab rendszer is [52].

A Burke [112] által végzett tesztek alapján rosszul teljesítettek az alábbi eljárások:

- Súlyozott: legjobban a KO/TA kombináció volt megbízhato, ha Pearson-féle korreláción alapuló hasonlóságot használunk. Az alapvető probléma, hogy ezek a rendszerek eltérő ajánlásokat tesznek, melyeket nehéz jól súlyozni.
- Váltogató: hibája, hogy nem tud “univerzális” konfidencia értéket megjelölni, mely a váltásokat irányítja, és nem nyújt egyforma teljesítményt különböző típusú felhasználók esetén. Legjobb kombinációja a TU/KO.
- Tulajdonság kombináló eljárások: egyik kombináció sem emelhető ki. Jellemzően minden pár esetén inkonzisztenciát észleltek.
- Közbeékelő eljárás: a kipróbált 6 kombináció egyike sem mutatott szinergiát, és az első lépésben betanult modellek egyike sem bizonyult megbízható inputnak a második lépéshez.

Jól teljesített azonban az elmélyítő és a tulajdonság kiterjesztő eljárás. Előbbi esetén a KO/TU és KO/TA kombinációk teljesítettek messze jobban a többinél, utóbbi esetében pedig a KO/TA kombináció volt az, ami a teljes vizsgálat során a legjobb teljesítményt nyújtotta. Általánosságban elmondható, hogy adott feladat esetén a jó hibrid kombinációk és azok helyes sorrendjének megtalálásához figyelembe kell venni az egyes eljárások konzisztenciáját és pontosságát, hogy szinergiát hozzunk létre.

3.6. Az ajánlórendszerek jóságának mérése és kihívásai

A felhasználók számára tett ajánlások pontosságának mérésére megannyi mutatószám létezik. Ezek közül a szakirodalomban leggyakrabban használtak kerülnek az alábbiakban bemutatásra.

Érdemes bevezetni a következő jelöléseket: ha egy felhasználó egy adott terméket érdekesnek talál és a rendszer ajánlja, az igaz-pozitívként (*IP*) kerül megjelölésre, míg ha nem ajánlja, az hamis-negatív (*HN*). Ha nem tartja érdekesnek, de ajánlásra került, az hamis-pozitív (*HP*), de ha nem ajánlja, akkor igaz-negatív (*IN*). Ezek összefoglalását a *4. táblázatban* láthatjuk.

4. táblázat: Az ajánlások pontossága

	Ajánlja	Nem ajánlja
Érdekes	Igaz-Pozitív	Hamis-Negatív
Nem érdekes	Hamis-Pozitív	Igaz-Negatív

- Az értékelések pontosságának mérésére leggyakrabban használt mutató az **RMSE** (Root-mean-square Error) [120]. Ezt alkalmazták többek között a Netflix Prize kiértékelésénél is. Legyen az *(u,i)* felhasználó-termék párok halmaza *H*, valamint $r^*_{u,i}$ az *u* felhasználó *i* termékre vonatkozó kalkulált értékelése, míg $r_{u,i}$ a valódi értékelés. Ekkor az ajánlások pontossága:

$$RMSE = \sqrt{\frac{1}{|H|} \sum_{(u,i) \in H} (r_{u,i} - r^*_{u,i})^2}$$

- **MAE** (Mean absolute Error) szintén a becsült és valós értékek közötti eltéréseket méri, de kevésbé bünteti a nagy hibákat, mint az RMSE [121].

$$MAE = \frac{1}{|H|} \sum_{(u,i) \in H} | r_{u,i} - r^*_{u,i} |$$

- A **Rand-index** annak mérésére szolgál, hány százalékban találja el jól a rendszer a felhasználó ízlését. Kalkulációja *RI=(IP+IN)/(IP+IN+HP+HN)*. Ennek hátránya, hogy az *IP* és *IN* értékeket egyforma súllyal veszi figyelembe, mely bizonyos esetekben félrevezető lehet. Ezt igyekszik korrigálni az F-érték.

- **Precizitás**: Az ajánlott termékek hány százaléka bizonyult érdekesnek a felhasználó számára, azaz $P=IP/(IP+HP)$, melyet más néven **Wallace-indexnek** is neveznek (B^I)
- **Emlékezés** (recall): vagyis az igaz-pozitív ráta, $R=IP/(IP+HN)$, más néven a Wallace-index (B^{II}).
- **F-érték**: Az R és P értékek súlyozására bevezetve egy nemnegatív β paramértert az alábbi formulával számolható: $(\beta^2+1)PR/(\beta^2P+R)$, ami $\beta=0$-ra megegyezik a P értékkel.
- **Jaccard-index** már a korábbiak során bemutatásra került. Itt is a hasonlóság mérőszámaként alkalmazható, az alábbi módon: J=IP/(IP+HP+HN). [62]
- **Fowlkes-Mallows-index** a klasszifikáló algoritmus pontosságát méri. Minél nagyobb az értéke, annál inkább hasonlít az ajánlásban szereplő klasszifikáció a felhasználók ízlésvilágához.

$$FM = \sqrt{\frac{IP}{IP + HP}}\sqrt{\frac{IP}{IP + HN}}$$

A fenti mérőszámokról bővebb összefoglalót olvashatunk Armstrong és Collopy cikkében [121]. Az ajánlórendszereket azonban nem csak ajánlásaik pontossága alapján mérhetjük, hanem számtalan más tulajdonságukat is, úgy mint robosztusság, alkalmazkodókészség (adaptivity), megbízhatóság, skálázhatóság (scalability), hasznosság (utility), sokszínűség (diversity), lefedettség (coverage), stb. Ezek mérhetőségéről bővebben Ricci et al. [123, pp. 258-293.] könyvében.

Bár az ajánló technikák bemutatása során már igen sok korlátot és kihívást, illetve azok kiküszöbölésére tett kísérletet megismerhettünk, az alábbiakban a legfontosabbakat ismertetjük.

- Az **adatok hiányossága, ritkasága** (sparsity): Amennyiben a termékek számához mérten igen kevés értékelés áll rendelkezésünkre, a hasonló felhasználók azonosítása a Pearson-féle korrelációs együtthatóhoz hasonló technikákkal igen nehéz. Ennek kiküszöbölésére javasolja Su et al. [57] az **IBCF** eljárást (Imputation-boosted Collaborative Filtering method), mellyel előbb feltöltik a felhasználók által a termékekre adott értékelésekből összeállított mátrixot, majd erre alkalmazzák a Pearson-féle korrelációs mutatót. Az értékelések kiegészítésének rengeteg módja lehet: lineáris regresszó, átlag beillesztése, **BMI** (Bayesian Multiple Imputation) eljáráson alapuló megoldást (lásd Rubin [131]), vagy gépi tanulásnál alkalmazott klasszifikáló eljárások, mint a neurális hálók, döntési fák, naive Bayes-i klasszifikáló eljárás (lásd [130]) és az SVM (bővebben Duan et al. cikkében [129]). Átfogó vizsgálat után úgy találták, hogy a legoptimálisabb megoldás, ha viszonylag sűrű értékelési adat esetén érdemes az IBCF eljárás naive Bayes-i klasszifikációra építve tölti ki az üres helyeket, míg ritkább adatok esetén átlaggal érdemes helyettesíteni).

- A **szürke bárányok** (grey sheep) elnevezéssel azokat illeti a szakirodalom, akiket döntéseik alapján nem tudnak besorolni egyetlen csoportba sem, mondhatni, nincs jól leírható karakterük. Mivel nem találunk egykönnyen hozzájuk hasonló felhasználót, így a kollaboratív szűrési eljárások náluk csődöt mondanak, de a tartalom alapú szűrés is rosszabb találati aránnyal működik esetükben, mint másoknál. A problémát és annak lehetséges megoldásait részletesen tárgyalja Ghazanfar és Prugel-Bennett [128].
- A **nehéz indulás** (cold start) problémája talán a legtöbbet emlegetett kihívása az ajánlórendszereknek. Ez leginkább a kollaboratív technikákra igaz. Rong et al. [56] Monte Carlo algoritmust javasol a kezdeti információhiány áthidalására, melyet nagy sikerrel alkalmaznak a felhasználók hasonlóságának előkalkulálására és az értékelések kiszámítására. Mint láttuk korábban, Canny [72] javasolja a feltöltést a mátrix sorainak vagy oszlopainak átlagából kalkulált értékekkel. Vucetic and Obradovic [73] a ritka értékelések problémáját igen nagy hatékonysággal oldja meg: a termékek közötti hasonlóságok alapján OLS becsléssel kapott lineáris regressziós modellekkel kalkulálják a felhasználók várható értékeléseit. A tartalom alapú szűrők ezzel szemben akár egy kedvelt termék alapján is tudnak hasonlót javasolni feltéve, hogy rendelkezésükre áll a termék megértéséhez szükséges mennyiségű értékelés, hogy a látens faktorokat, tulajdonságokat fel tudják tárni ezek alapján. A tudás alapú megközelítések jól veszik fel a harcot ezzel a problémával, ám a nehézség itt rendszerint **tudás megszerzésekor** (knowlsedge acquisition) merül fel.
- A folyamatosan növekedő felhasználó- és termékbázis állandó kihívást jelent az ajánlórendszereknek, melyeknek frissíteni kell az új elemeket figyelembe véve a személyre szabott ajánlásokat. Hagyományosan jól küzdenek a **skálázhatóság** problémájával a klaszterező eljárások, ám ez gyakran a precizitásuk rovására megy. A memória alapú kollaboratív eljárások között jó példa a Chee et al. [69] által javasolt *RecTree* algoritmus, mely első lépésként klaszterekbe sorolja a felhasználókat (k-means clustering eljárárssal), majd a már kisebb csoportok közül csupán a relevánsakkal foglalkozva egy újabb klaszterezési lépésben választja ki a leginkább hasonló felhasználókat.
- A **szinonímák** okozta probléma főként a tartalom alapú szűrőknél jelentkezik, melyek szövegbányászati eljárásokkal operálnak. Ennek leküzdésére gyakran használt eljárás, hogy a problémát okozó szót nem önállóan, hanem más szavakkal kombinálva keresik a szövegben. Például Blei et al. [124] látens Dirichlet allokációs eljárást (Latent Dirichlet Allocation) javasol a szinonímák kezelésére, mely feltételezi, hogy a dokumentumok kisebb témák összességéből állnak elő, és minden szó ezek egyikéhez köthetően van jelen a dokumentumban.

- Ahol pénzt lehet keresni, ott a **szélhámosok támadásával** (shilling attacks) is számolni kell. Gyakori jelenség, hogy egyes online áruházakban a gyártók igyekeznek saját termékeik átlagos értékelését mesterségesen feltornázni, míg a konkurenciát rossz színben feltüntetni. Chirita et al. [126] olyan algoritmust javasol, mely a szokatlan értékelési szokásokat azonosítja és a rosszhiszeműnek minősített felhasználók értékeléseit megsemmisíti.
- A **sokszínűség** (diversity) kérdése igen könnyen csorbul tekintve, hogy az eljárások döntő többsége hasonlóság keresésén alapul. Ennek következtében a népszerű termékek mindinkább előtérbe kerülnek, míg az új termékek nehezen jutnak figyelemhez, és így értékeléseket sem kapnak. Ennek leküzdése érdekében Adamopoulos és Tuzhilin [127] olyan algoritmust javasolnak, mely esetenként "váratlan" ajánlásokkal áll elő, ám igyekszik kerülni, hogy a felhasználónak csalódást okozzon.
- A **stabilitás és formálhatóság** dilemmája abban rejlik, hogy a rendszernek egyszerre kell annyira "jól emlékeznie" a korábbi értékelésekre, hogy ajánlásait ne billentsék ki esetlegesen felmerülő egyszeri igények, de ne is ragadjon bele tartósan olyasmibe, amit a felhasználó igényei idővel túlhaladtak. Tehát viszonylag gyorsan különbséget kell tenni az ízlésben beálló változás és az eseti kilengések között. A szakirodalomban gyakran hozott példa erre az a felhasználó, aki vegetáriánus étrendre állt át, ám még mindig kapja az értesítéseket a hús akciókról. Erre adhat megoldást a tudás alapú és a tartalom alapú eljárások kombinálása [60].

A fentieken túl további kihívásokat és megoldási kísérleteket tárgyal Adomavicius és Tuzhilin [104] összefoglaló cikkében.

3.7. Turisztikai helyszínek ajánlórendszerének modellezése - Szakirodalmi áttekintés

3.7.1. A turisztikai ajánlórendszerekről általában

A mai online és mobil eszközökre készített ajánlórendszerek már nem csupán elektronikus változatai a papír alapú túristakönyveknek. Az utóbbi években tucatszám jelentek meg olyan oldalak, melyek célja utazásaink megkönnyítése, és az ismeretlen terepen való eligazodás. A 2000-es alapítású Tripadvisor elektronikus túrakönyvként kezdte pályafutását azzal a különbséggel, hogy lehetőséget adott a felhasználóknak, hogy értékeléseket adjanak, de saját bevallásuk szerint sem azzal a szándékkal, hogy ajánlórendszert alakítsanak belőle [148]. Jópár évig tartott, míg felismerték ennek lehetőségét, és csak 2012-ben kapcsolódtak a Facebookhoz. Ugyan 2014-ben már

az elektornikus turisztikai piac legbefolyásosabb szereplőjeként tartották számon hatalmas látogatóközönségének köszönhetően, ám mindmáig igen kevéssé innovatív módon közelíti meg az ajánlásokat, és inkább a nagy tömegek által értékelt helyszínek ranglistáját bocsátja rendelkezésre. A Tripadvisor népszerűsége mögött nagyon lemaradva találunk több turisztikai ajánlórendszert is, melyből kiemelnék három különböző technikán alapuló megoldást:

- A TripSay kollaboratív szűrésen alapuló rendszer, mely lehetőséget ad látványosságok, szolgáltatások és tevékenységek keresésére, és ajánlásai során nem csak hasonló felhasználókra támaszkodik, hanem a Facebook integráltságán keresztül a felhasználó személyes közösségi hálójára is. Sajnos a TripSay jelenleg hivatkozott helyén sajnos már nem elérhető [149].
- A Heracles [150] tartalom alapú rendszer, mely a működéséhez szükséges tartalmakat szövegbányász eszközökön keresztül gyűjti be megannyi turisztikai oldalról, majd tárja a felhasználók elé.
- A DieToRecs tudás alapú rendszer, mely kérdéseken keresztül tárja fel a felhasználói igényeket és szűkíti az érdekesnek vélt tartalmak körét. Sajnos a DieToRecs jelenleg hivatkozott helyén sajnos már nem elérhető [150].

Felmerülhet a kérdés: ha vannak pontosabb ajánlást adó rendszerek, miért választják mégis a Tripadvisort? Mint sok piacon, itt is az döntött, ki volt jelen előbb, és kinek sikerült akkora felhasználói bázisra, és ezzel együtt halmas információtömegre szert tenni, ami jóval csábítóbb, mint a kisebb oldalak szűkös információkészlete. A továbiakban azt a néhány tipikus tartalmi elemet járjuk körül, amit az ajánlórendszerek szolgáltatásként nyújtanak:

- A mobil turisztikai ajánlórendszerek legfőbb szolgáltatása a látványosságok ajánlása, mely történhet útvonal köré csoportosítva (lásd Vansteenwegen et al. [142]), tematikusan, közelség alapján listázva (Horozov et al. [137]), vagy kollaboratív szűrési eljárással számított várható értékelések alapján, lásd Brown et al. [143]. Egyéb körülményeket is figyelembe vehet (context-based recommender systems), mint például az időjárás (Gavalas és Kenteris [140]), időablakok (Cheverstet al. [135]), utazási mód (autó, tömegközlekedés, gyalogos), lásd Savage et al. [138].
- Turisztikai szolgáltatásokkal igen sokan egészítik ki ajánlásaikat, legyen szó akár szállodáról, étteremről vagy közlekedési lehetőségekről (pl.: Horozov et al. [137] és Savage et al. [138]). Bizonyos esetekben adott a lehetőség korlátok beállítására is, hogy szűkíteni lehessen a szolgáltatások körét például az árkategóriájuk szerint (lásd Yu és Chang [144]).
- A kollaboratív szűrésen alapuló megoldások gyakran szorgalmazzák, hogy felhasználóik új látványosságokat, éttermeket osszanak meg a többiekkel, így bővítve a választékot, és kielégítve a sokszínűségre vonatkozó elvárásokat (például [138], [139], [140], [143]). Egyes esetekben

lehetőség nyílik akár új barátokat vagy utitársakat szerezni, és javasolják hasonló ízlésvilágú emberek együtt utazását, példa erre Zheng és Xie [141] alkalmazása.

- Lucchese et al. [253] az alapján készít turisták számára útvonaltervezése során helyszínekre vonatkozó ajánlásokat, hogy az adott városban a Flickr-en vagy más közösségi oldalon található fotók milyen gyakorisággal készülnek egy adott helyszínről. Ezt kiegészítve a helyszín Wikipedia oldaláról nyerhető információkkal, könnyen pontozhatóak fontosságuk szerint a helyszínek. Lim [254] a Flickr fotók és bejegyzések alapján javasol négy különböző útvonaltervező eljárást, melyek profit függvényeik kalkulációs elveiben különböznek, de mind felhasználják a turista korábban látogatott helyszíneinek listáját, mellyel javítani kívánják az ajánlások pontosságát. Kimutatta, hogy a legnépszerűbb (és lehető legtöbb) helyszíneket ajánló eljárása vezetett a legnagyobb felhasználói elégedettséghez.
- Sok alkalmazás nyújt útvonaltervező megoldást is a felhasználók számára. Cheverst et al. [135] korai megoldásában a legrövidebb útvonalat kalkulálja a jelenlegi hely és a legközelebbi meglátogatandó pont között. Shiraishi et al. [145] az ismert utazóügynök problémát alkalmazza városi útvonaltervezésre. Néhány megoldás képes akár többnapos túrák tervezésére is, például Vansteenwegen et al. [142], ahol néhány paramétert is beállíthat megának a felhasználó, hogy személyre szabhassa az ajánlást, ilyen például látogatás napjainak száma, a kezdő és végpontja a túrának, a helyszínekre vonatkozó preferenciák, a séta üteme, stb. Garcia et al. [146] 2013-as megoldása volt az első olyan többnapos útvonaltervező algoritmus, mely a meglátogatott helyszínek nyitvatartási idejét is figyelembe veszi csakúgy, mint a tömegközlekedési alternatívákat a gyaloglás mellett. Sajnos akkori jelentésük szerint az algoritmus számításigénye még lehetetlenné tette gyakorlati alkalmazhatóságát.

A mobileszközökre készült vagy tervezett turisztikai ajánlórendszerek kiváló összefoglalóját adja Gavalas et al. cikke [147].

3.7.2. A helyszínek integrált adatbázisának kihívásai

Rodrigues et al. [132] munkája alapján tudjuk, hogy a látványosságok, helyszínek jó klasszifikálásának egyik legfőbb gátja lehet, ha a kategóriák és alkategóriák szerkezete túlzottan bonyolult és szofisztikált. Amennyiben egy ilyen rendszerben a klasszifikálást nem szakemberek végzik, hanem a felhasználókra bízzuk, vagy megengedjük önkényes kategóriák bevezetését, a klasszifiklás nem lesz koherens. Az ilyen körülmények nehezítik továbbá a nem azonos szerkezetben osztályozott helyek adatbázisainak integrálását, ahol a kategóriákat és alkategóriákat

össze kell párosítani. Rodrigues-ék cikkükben többezer POI-ból álló adathalmazon tett kísérletet két különböző kategorizálási rendszer elemei közötti párosításra. A két adathalmaz részleteit az *5. táblázatban* rögzítettük. Az egyik rendszer a NAICS (North American Industry Classification System), ahol 6 szinten összesen 2332 alkategória szerepel, míg a másik a Yahoo!, ahol 3 hierarchia szinten találunk nagyjából 1300 kategóriát.

5. Táblázat: A NAICS és Yahoo! adatbázisának összevetése

Dataset	A	B
NAICS source	D&B	InfoUSA
Total POIs	7289	44634
Distinct NAICS	504	689
Distinct Yahoo! categories	802	1109
Distinct Yahoo! category combinations	569	1002
Category combinations that appear only once	136	92
Categories that appear only once	181	107
NAICS that appear only once	115	96

A JaroWinkler TF-IDF algoritmust használva meghatározzák az egymáshoz közel álló neveket, figyelmen kívül hagyva az írásjel hibákat és rövidítéseket (bővebben az eljárásról Cohen et al. [133] cikkében olvashatunk). A manuálisan ellenőrzött minták alapján 98%-os egyezést sikerült elérniük. Tesztelésre került megannyi gépi tanuló eljárás (machine learning): szabály alapú algoritmus (rule based), döntési fák (tree based), példa alapú elárás (instance based) és bayes-i háló (bayes-network). Az összes kipróbált eljárás közül az IBk algoritmus (k-nearest neighbor with distance weighting) volt a legsikeresebb, mely 86,6%-os pontossággal működött, de nem sokkal maradtak el a döntési fa alapú algoritmusok sem. A tesztadatok eloszlását figyelembe véve (a helyszínek nem csupán néhány klasszifikációs kód körül koncentrálódnak, hanem inkább egyenletesen), a döntési fák várhatóan gyengébben teljesítenek. A vizsgálat során fény derült arra is, hogy a klasszifikálás fő akadálya a túlszofisztikált (főként 5-ös és 6-os hierarchia szintű) alkategóriák szerepeltetése, mely zavarossá teszi a helyszínek besorolását. Ennek érdekében néhány egyszerűsítést tettek, megszüntetve az inkonzisztencia egy részét, és “szuperkategóriákat” hoztak létre. Az így keletkezett adathalmazon újból futtatva a már tesztelt algoritmusokat ismét az IBk került ki győztesen, ám már 94,1%-os eredménnyel. Elmondható továbbá, hogy az összes klasszifikáló eljárás jobban teljesített a megváltozott körülmények között. Az itt bemutatott átfogó tanulmány tanulságaként elmondható, hogy az ajánló rendszerekkel szemben támasztandó egyik fő elvárás a termékek (esetünkben a helyszínek, vagy látványosságok) lehetőleg egyszerű klasszifikálása.

3.7.3. A helyszínek értékelésének lehetőségeiről

Az alábbiakban néhány, a gyakorlatban működő turisztikai ajánlórendszer pontozási eljárásait mutatjuk be.

1. Aurigo

Az Aurigo alkalmazása nem szándékozik több napos útvonaltervezési problémát megoldani, melyekről majd a 4. fejezetben szólunk. A Yahi et al. cikkében [250] szereplő ajánlórendszer egy adott pontból kiindulva választja ki a következő célpontot úgy, hogy a lehetséges pontok listáját a jelenlegi helyzetének *r* sugarú körén belülre szűkíti, ahol *r=d(2i + 1)*, ahol *i*=1, ha könnyed sétát tervez a felhasználó, *i*=2, ha átlagos, és *i*=3, ha hosszú sétát tervez. Ha az *r* sugarú körben bármely lépésben nincs legalább 5 pont, akkor *r'=d(2i+3)* sugarú kört nézünk. A *d* faktort alapesetben 60m-re állítják be. A körön belül található pontok között egyfajta mohó algoritmussal választja ki a következő pontot az alábbi célfüggvény alapján:

Összesen 100 pont gyűjthető: 80 pont az értékelésekből és 20 a megtekintésekből. Ezt "népszerűségi indexet" az alábbi módon kalkulálja egy adott *p* pontra:

$$pop(p) = 20(s(p) - 1) + 20\frac{v(p,j)}{v_{max}(p,j)}$$

ahol *s(p)* a látványosság átlagos értékelése a felhasználóktól (1-5), *j* azt a kategóriát jelöli, amibe a *p* pont tartozik (pl. múzeum), *v(p,j)* a látványosság összes megtekintésének száma, míg $v_{max}(p,j)$ a *j*-edik kategóriájába tartozó összes pont közül a legnagyobb a látogatottság érték. Ezután azt a pontot választja következő lépésben, mely egyrészt az r sugarú körbe esik, valamint az alábbi mutatószám értékét maximalizálják:

$$Score(p, \lambda, w, path) = t(p, path) - \lambda pop(p)a(j)w$$

ahol

- *a(j)* a felhasználó j-edik kategóriára adott értékelése, melyet inputként meg kell adni. Összesen 4 kategória létezik (múzeum, park, emlékmű és filmes helyszín), melyet 1 és 5 közötti egész számmal pontozhat.
- λ empirikus koefficiens a távolság és az értékelések súlyozására, melyet a szerzők 0,0002-nek választottak
- *pop(p)* a korábban ismertetett népszerűségi index

- *t(p, path)* a *p* pont eddig megtervezett útvonaltól (*path*) való legkisebb távolsága
- *w* a "távolság faktor", melynek értéke 1, ha a pont az úthoz közel van, 2 ha átlagos távolságra, és 3 a távoli pontokra. A távolság-kategóriákra vonatkozó intervallumokat cikkükben nem közölték.

Sajnos értékelési rendszerük (már csak a kategóriák alacsony száma miatt is) nem nevezhető kifinomultnak, és a tervezett útvonaluk bár a keresési rádiusz alkalmazása okán nem ad egymástól távol eső pontokat, mohó tulajdonsága miatt esetleges a globális optimum elérése.

2. City Trip Planner

Souffriau és Vansteenwegen [240] túratervező weboldalának célja akár többnapos túrautak tervezése egyéni értékelések alapján. A felhasználónak az alábbi bemeneti adatokat lehet megadnia:

- A célváros
- Ott töltött napok száma
- A túra tempója: gyors, közepes, lassú
- Látványosság kategóriák értékelése (1-5 közötti egész számmal), 8 kategóriát (főbb látványosság, tevékenység, templom, emlékmű, múzeum/művészet, természet, vásárlás, utca/tér)

Minden látványosságot megfeleltetnek egy kategóriának, és a pontszáma a felhasználó kategóriára adott pontszáma és az adott látványosság korábbi felhasználók által adott értékelése (szintén 1-5 között) összegeként áll elő. Az adott helyszín értékelése az összes felhasználó által adott pontszám átlagaként áll elő, és nem súlyozzák azokat aszerint, mennyire vannak közel az adott felhasználó ízlésvilágához. Ezt könnyedén számolhatnák például a 8 kategóriára adott értékeléseikből előálló vektorok által bezár szögből. Az így megállapított pontszámokat tekintik az adott helyszín meglátogatásával begyűjthető profitnak, és céljuk ezek összegének maximalizálása a napok során. Az útvonaltervezés a 2009-es cikkükben [233] bemutatott ILS algoritmus alapján történik, mely igen hatékony heurisztikus megoldást ad a TOPTW feladatra (lásd *A.4. melléklet*), ám célfüggvényük (azaz a profitösszeg-maximalizálás) csak esetlegesen szolgálja a személyre szabott ajánlások jóságát. Erről bővebben a 4. fejezetben szólunk.

3. Tripadvisor

A világ jelenleg legnagyobb turisztikai ajánló oldala, ám - számomra érthetlen okból - nem tesz személyre szabott ajánlásokat, hanem sokmillió emberből álló felhasználóbázisára építve az általuk adott értékelések alapján rendezik sorba a látványosságokat. Ezzel tulajdonképpen a közízlésre alapozva tesznek ajánlásokat, figyelmen kívül hagyva az egyéni preferenciákat, melynek kiszolgálása érdekében csupán szűrési lehetőségeket kínálnak fel: geográfiai dimenzióban, valamint

a látványosságokat leíró 18 kategória alapján. Ezen kívül felkínálnak néhány sablon útvonaltervet, melyet sok felhasználó értékelt már. Noha csak Budapesten több mint félmillió értékelést adtak le a felhasználük, mégsem élnek az ajánlórendszerek adta lehetőségekkel, hogy ajánlásaikat személyre szabják, elkövetve ezzel a piacvezetők gyakori hibáját.
További turisztikai ajánlórendszerek is elérhetőek az interneten, például a Tripomatic, de fölösleges volna bővíteni a sort, mert bár többé-kevésbé kiterjedt adatbázissal rendelkeznek, és kategorizálták a látványosságokat, valódi ajánlásokat azonban nem tesznek, sőt a legtöbb esetben még értékeléseket sem gyűjtenek a felhasználóktól a látogatást követően.

3.8. A turisztikai ajánlórendszer megalkotása

3.8.1. A minimális információ problematikája

Egy potenciálisan sikeres turisztikai alkalmazás egyik előfeltétele, hogy a lehető legnagyobb pontossággal tudja a felhasználók számára releváns, érdeklődésükre számot tartó célpontokat kínálni. Ez azonban koránt sem olyan egyszerű, amennyiben nem áll rendelkezésünkre kellő információ az adott felhasználó ízlésvilágáról. Adatok hiányában nem tehetnénk mást, mint a széles körben kedvelt célpontokat ajánlani számukra, reménykedve abban, hogy preferenciáik nem térnek el nagyon az ¨átlagostól¨. Az ajánlások pontosítása érdekében azonban mégis jobb információt gyűjtenünk, melyet két forrásból szerezhetünk meg: egyrészt egy előzetes kérdőívvel, melyben a felhasználó általános érdeklődési köreit probáljuk feltárni (tudás alapú szűrés), másrészt a már meglátogatott helyekkel kapcsolatban tudunk visszajelzést kérni, hogy az mennyire volt élvezetes számára (kollaboratív technika). Az így kapott felhasználói profilokból kialakul egy kép a felhasználó ízlésvilágára vonatkozóan, melyet idővel - folyamatos visszajelzések gyűjtésével - tovább finomíthatunk, sőt a preferenciák időbeli változását is nyomon tudjuk követni.
A rendelkezésre álló információ hatékonyabb felhasználása érdekében kézenfekvőnek tűnik, hogy a felhasználók preferencia profiljait összevetve keressünk egymáshoz hasonlókat, hiszen ezzel is tudjuk javaslataink minőségét javítani. Példának okáért így módunkban állna a felhasználónak olyan látványosságokat ajánlani, melyet egy hozzá hasonló preferenciákkal bíró társa korábban már látott, és pozitív visszajelzést adott róla, akárcsak az amazon.com teszi könyvek esetében.
A preferenciák feltérképezésének egyik fő kihívása, hogy ezt a felhasználók idejének lehetőleg minimális igénybevételével oldjuk meg, hiszen célunk éppen az lenne, hogy időt és energiát takarítsunk meg számukra a tervezés során. A legtöbb turisztikai ajánlórendszer általában a teljes felhasználói csoport értékelése alapján kialakított ranglistát tekinti irányadónak minden

felhasználója számára, és nem fordít figyelmet az egyéni preferenciákra. Szerencsére azonban nem vagyunk egyformák, mint ahogy az ajánlórendszerekről sem gondolkodik mindenki úgy, ahogyan a Tripadvisor alkotói. Payne et al. [134] nyomán tudjuk, hogy a felhasználók nem ismerik saját preferenciáikat igazán, és sokkal inkább tudnak nyilatkozni konkrét szituációkban, mikor látványosságok egy listájáról kell dönteniük. Világos, hogy a felhasználót minél kevésbé kellene terhelni értékelések adásával és döntési szituációkkal, ám ha nem tesszük, az az ajánlások pontosságának a rovására megy. Ezt a dilemmát járja körül a tudás alapú ajánlórendszerek kapcsán Felfernig és Burke is [108].

A Vansteenwegen és társai által készített, jelenleg is működő CityTripPlanner [142] az alábbi indormációkat gyűjti be a tervezéshez:

- A túra alapadatai: mely városban, mikor érkezik és hány napot tölt ott.
- Lassú, közepes vagy gyors tempóban közlekedik-e a túrázó (ennek megfelelően kalkulálnak majd a két pont közötti menetidő becslésekor, illetve, hogy mennyi időt tölt az egyes helyeken)
- 5-ös skálán értékelhetőek az alábbi kategóriák: természet, emlékművek, múzeumok/művészet, templomok, vásárlás, utcák/terek, közkedvelt túrista csomópontok, szórakozás/¨látnivalók¨.

A fenti adatlista nem tekinthető hosszúnak, két helyen mégis kritikával illetném:

- Az 5-ös, és általában a páratlan, értékelési skálát nem tartom jónak, mert akik hajlamosak elkerülni a döntési szituációkat, vagy nem szeretnek véleményt formálni, arra készteti, hogy a középső értéket vállasszák, kivonva magukat ezzel a döntés alól. Ehelyett jobbnak látom egy 4-es skálát bevezetni: 0 - egyáltalán nem érdekes, 1 - ha van rá időm, megnézem, 2 - kifejezetten érdekel, 3 - látnom kell/nagyon érdekel
- Érdemes a helyszínek klasszifikációs kategóriáit átgondolni, ha nem is azért, hogy tovább szofisztikáljuk, sokkal inkább annak érdekében, hogy olyan kategóriák alakuljanak ki, melynek mentén jól elkülöníthetőek a különféle érdeklődési körű emberek. Ennek megfelelően az általunk javasolt 17 kategóriát a *6. táblázatban* foglaljuk össze:

6. Táblázat: A látványosságokat leíró faktorok

Kiemelkedő látványosság	Templom/vallási témájú hely	Múzeum/művészet
Történelem/kultúra	Építészet/épület/homlokzat	Történelmi helyszín/emlékmű
Utcák/terek	Kilátópont	Természet/park
Egyetem/tudomány/technológia	Család/gyermek program	Fürdő/sport/rekreáció
Piac/helyi ételek	Kávézó/étterem	Színház/mozi/szórakozás
Éjszakai élet/zene/bár	Vásárlás/divat	

A fenti kategóriák jó tematikus lefedését adják a helyszíneknek, és kombinációjukkal az összes látványosság jól leírható eddigi tapasztalataink alapján. A jövőben szükségét látom egy olyan vizsgálat elvégzésének, ahol a felhasználók egyes látványosságokra adott értékeléseik alapján mátrixfaktorizációs eljárással tárjuk fel azokat a faktorokat, amelyek kombinációjaként az egyes helyszínek leírhatóak, akárcsak Sarwar et al. cikkében láttuk [84]. Mivel az SVD eljárás csak akkor ad jó megoldást, ha az értékelések viszonylag sűrűek, így ezt adatok hiányában most nem tehetjük meg, és helyette a fenti saját faktorok kerültek meghatározásra.

3.8.2. A felhasznált adatok

Empirikus vizsgálatunk során 3 városra állítottunk össze látványosságokat tartalmazó, részletes adatbázist, mely az ajánlórendszer tartalom alapú modulját alapozza meg.

Az ajánlórendszer teszteléséhez 3 város (Budapest, London és Párizs) összesen 500 helyszínéből álló adathalmazt hoztunk létre, mely az alábbi változókat tartalmazza:

- POI_Name: a helyszín neve
- Category_i: a helyszín kategóriája (minimum egy, legfeljebb 3 ilyen kategória lehetséges), melyek a 3.8.1-es alfejezetben meghatározott 17 kategóriából kerültek kiválasztásra, (például a Nagytétényi kastély esetén ezek: Építészet/épület/homlokzat, Múzeum/művészet, Történelem/ kultúra)
- Relevance_i: az egyes kategóriákhoz tartozó relevancia értékek, vagyis, hogy mennyire írja le jól az adott kategória a helyszínt, értéke 0-3 közötti egész szám, (a Nagytétényi kastély példájánál maradva ezek a relevancia értékek rendre: 3, 3, 2)
- Importance: a helyszín fontossága, azaz, hogy mennyire számít ismertnek, kiemeltnek egy adott látványosság, (ez mérhető például a látványosságot leíró wikipedia oldal hosszában, vagy hogy mennyi egyedi találatot ad a google keresője, illetve később a látogatók számából is következtethetünk rá), értéke 1, 2 vagy 3 lehet (3 jelöli a legfontosabb helyeket)
- City: a város, ahol a látványosság található
- WikiLink: a látványosság wikipedia oldala, mely a kísérlet során tájékoztatást ad az alanyoknak a látványosságról, ha azt esetleg nem ismerik.

Az elkészült adatbázis egy részletét a *7. táblázatban* láthatjuk.

7. Táblázat: Az ajánlórendszer teszteléséhez használt adatbázis

POI_name	category1	category2	category3	relevance1	relevance2	relevance3	Importance	wiki link	City_country
Clarke Ádám tér és alagút	Architecture/Facade	History/Cultu	Street/Squar	3	1	3	2	https://hu	Budapest,
Batthyány tér	Architecture/Facade	History/Cultu	Street/Squar	2	1	2	2	https://en	Budapest,
Budai vár	Architecture/	History/Cultu	Top sight	3	3	3	3	http://en.	Budapest,
Budapest Történeti Mu	Museum/Art	History/Cultu	Architecture/	2	2	1	2	https://hu	Budapest,
Keleti pályaudvar	Architecture/	History/Culture		2	1		2	http://en.	Budapest,

3.8.3. Az empirikus vizsgálat ismertetése

Célunk egy olyan ajánlórendszer megtervezése, mely alkalmas a felhasználók széles körét kiszolgálni. Ez szerves részét képezi majd annak a célfüggvénynek, melyet a 4. fejezetben igyekszünk maximalizálni a felhasználó igényeit kielégítő, egyéni útvonaltervezéssel, hiszen a látványosságok értékelései ebben a szakaszban kerülnek meghatározásra. A felhasználók helyszínekre vonatkozó értékeléseit minél pontosabban leíró kalkulációs eljárások vizsgálatát ebben a szakaszban mutatjuk be.

Az általunk vizsgált eljárások alapja az imént ismertetett adatbázis, mely azon a feltételezésen alapszik, hogy a turisztikai látványosságok jellemezhetőek különböző faktorok segítségével, ahogyan például a zene tulajdonságainak feltárása történt a Music Genome Project során [51]. Ezt bővítjük a felhasználók ezen faktorokra vonatkozó preferenciáinak begyűjtésével kérdezéses módszerrel. Így a vizsgálat alá vont eljárásunk egy hibrid ajánlórendszer, mely az alábbi egységekből áll:

- Tudás alapú modul: adatgyűjtési eljárás során a felhasználóktól tudás alapú kérdező eljárással kapjuk meg a megállapított 17 faktorra vonatkozó értékelésüket, (illetve a későbbi szakaszban egyéb paramétereket is, például a napi költségkeret összegét, és az időkorlátjukat is, mely az útvonaltervezéshez szükséges majd).
- Tartalom alapú modul: a látványosságok faktorok kombinációjaként történő leírása (ahol a súlyok a faktorokra vonatkozó relevancia értékek) a tartalom alapú szűrési technika előfeltétele. Ez lehetőséget nyújt olyan ajánlások készítésére, mely során a felhasználónak korábbi visszajelzései alapján olyan helyszíneket ajánlunk, melyhez hasonlóakat korábban már pozitívan értékelt. Ekkor a 17 faktort reprezentáló 17 elemű vektorba rendezzük sorben a hozzájuk tartozó relevancia értékeket, és vektor-cosinus eljárással [90] keresünk a korábban a felhasználó által pozitívan

értékelt látványosságokat reprezentáló vektorokhoz hasonlóakat, (vagyis olyanokat, melyek kis szöget zárnak be azokkal).

Ez a technika a korábban ismertetett hibrid eljárások osztályozásában a *tulajdonság kiterjesztő eljárások* (Feature-augmenting) közé tartozik, hiszen minden ajánlható elem értékeléseit egy másik technikával minden látványosságra új tulajdonságvektort alkothat, melyet az alapeljárás során (a kibővített információval együtt) felhasználunk. Nagy előnye az eljárásnak, hogy a tudás- és tartalom alapú megközelítések ötvözésének köszönhetően minimális információval el tud indulni a rendszer, gyakorlatilag csak a felhasználó 17 faktorra vonatkozó értékelésére és a célvárosra van szükség kezdeti információként. Cserébe az induló adatbázis előkészítése időigényes, hiszen a látványosságokat leíró faktorokat és azok kezdeti relevancia értékeit meg kell adni. Ezek a későbbiekben a felhasználók értékelései alapján finomíthatóak, ahogy egyre több információhoz jutunk a velük történő interakciók során. A rendszer a továbbiakban 3 típusú ajánlást lesz képes előállítani, ám ebből 2 csak kellő információ esetében lehetséges majd:

- A hibrid eljárásunk tisztán a felhasználó faktorokra vonatkozó értékelései és a látványosságokat leíró faktorok relevancia értékei alapján kalkulált mutatószámok alapján pontozza turisztikai célpontokat, és az ez alapján felállított rangsort közli a felhasználóval.
- A tartalom alapú szűrési technika alkalmazhatóvá válik, amint az adott felhasználó kellő sűrűségű értékelést adott le helyszínekre, ekkor ugyanis a rendszer már képes lesz olyan látványosságokat ajánlani, melyhez hasonlóakat a felhasználó korábban pozitívan értékelt.
- A kollaboratív szűrési eljárás előfeltétele, hogy kellő mennyiségű felhasználója legyen a rendszernek, és azoknak lehetőleg elég sok egyéni értékelése a látványosságokra vonatkozóan. Így lehetőség nyílik egy adott felhasználóhoz hasonló ízlésvilággal bíró embereket találni (akik a 17 faktorra hasonló értékelést adtak), és az azok által jóra értékelt látványosságokat ajánlhatjuk neki.

Mint az látható, adatok hiányában jelenleg csak az első típussal van lehetőségünk foglalkozni.

Ennek az ajánlórendszernek sikeressége - az adatbázis mellett - azon a függvényen múlik, amellyel a felhasználó által az egyes faktorokra adott értékelések alapján a látványosságokat pontozzuk. A kutatás jelen szakaszában az alábbi 4 kalkulációs eljárás által adott ajánlásokat kell értékelniük a résztvevőknek:

- *KPI1*: A felhasználó által az egyes faktorokra adott értékelések szorzata az adott faktor helyszínre vonatkozó relevanciájával összegezve az összes faktorra, majd ezt szorozzuk a helyszín fontosságával, (*imp* a helyszín fontossága, e_i az i-edik faktorra adott értékelés, és r_i az i-edik faktor relevanciája). Továbbá minden olyan látványosság értékelését 50%-kal csökkentjük,

aminek legalább 1 faktorából, melyhez legalább 2-es relevancia érték társul, már legalább 5 szerepel a kiválasztott látványosságok listáján. Ezzel azt kívánjuk elkerülni, hogy túlzottan egysíkú ajánlásokat tegyünk, és idővel büntetjük a hasonló elemeket.

$$imp \sum_{i=1}^{17} e_i r_i$$

- *KPI2*: Az első eljárás során kapott pontot megduplázzuk a 2-es fontosság érték esetén, ezzel próbáljuk előnyben részesíteni a kevésbé ismert helyszíneket, (legyen tehát *h*=1, ha *imp*=2, és 0 különben). Ezt elosztjuk a 0-tól különböző értékelést kapott faktorok számával, amit jelöljön *k*. Ezzel igyekszünk azoknak a helyszíneknek a hátrányát lefaragni, melyek bár bizonyos szempontból relevánsak és izgalmasak, de nem tartozik hozzájuk több faktor. Például a Postatakarék Bank épülete 2 faktor segítségével leírható, és a *KPI1* pontszámítás esetén hátrányt szenvedne azokkal a látványosságokkal szemben, amik 3 faktort is tartalmaznak. Az eljárás ellen szól, hogy ugyanakkor figyelmen kívül hagyja, hogy adott esetben sokkal inkább érdekes lehet a felhasználó számára egy olyan látványosság, mely akár 3 dimenzióban is élményt nyújt neki. Sőt, az átlagolás egy gyengébb faktor jelenléte esetén a pontszámítás végeredményét károsan befolyásolja. Ezt kiegészítjük még azzal, hogy a helyszín pontszámának értékét felezzük, amennyiben van olyan 3-as relevanciájú faktora, mely faktorhoz a felhasználó 0 értékelést rendelt.

$$\frac{imp \sum_{i=1}^{17} e_i r_i}{k}(1+h)$$

- *KPI3*: A *KPI2* eljáráshoz képest annyit módosítunk, hogy növeljük a kifejezés értékét annyiszorosával, ahány 9-es értéket találunk a relevancia-faktor értékelés szorzatok között (vagyis ahány esetben valami kiemelten érdekes a felhasználó számára), ezek számát jelölje *l*:

$$\frac{imp \sum_{i=1}^{17} e_i r_i}{k}(1+h)(1+l)$$

- *KPI4*: Az utolsó eljárás első 7 eleme a *KPI1* eljárás listájának első 7 eleme, míg további 8 elemét a *KPI2* eljárás szerint kalkulált listából vesszük sorrendben úgy, hogy duplikáció ne forduljon elő.

A fenti 4 eljárás által adott ajánlások vizsgálata érdekében létrehoztunk egy weboldalt (www.travelschedule.org), ahol a korábban ismertetett három város látványosságait értékelhetik az alábbi lépések szerint:

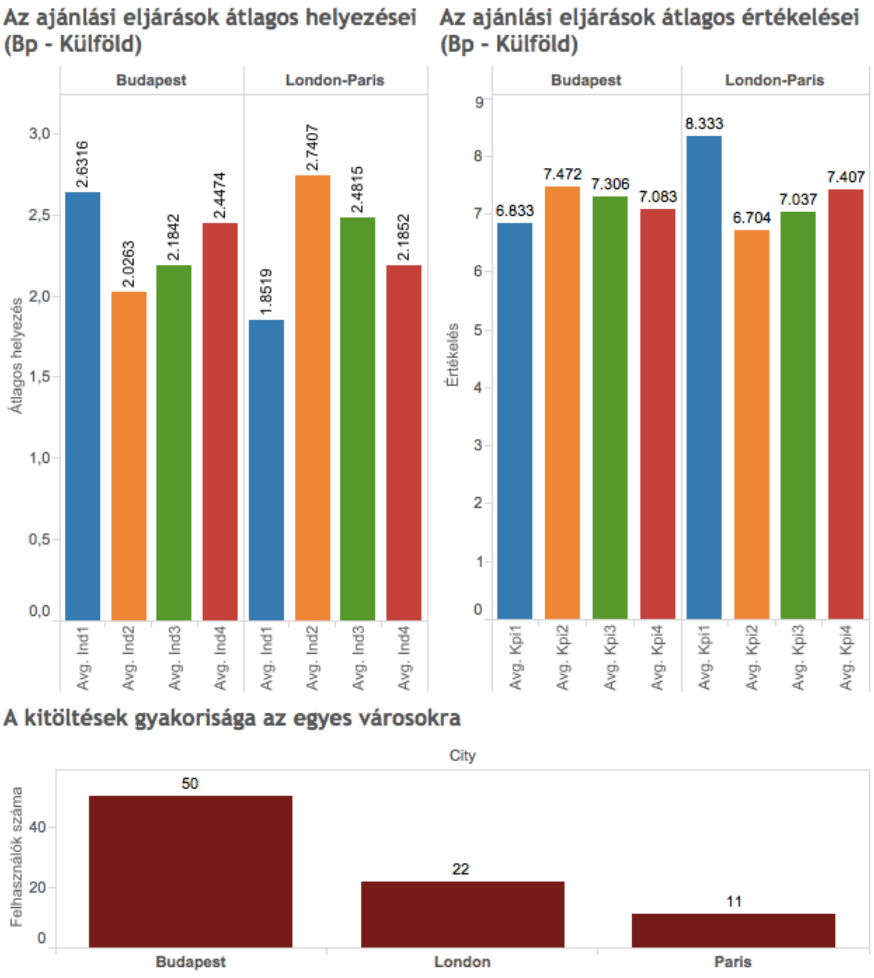
1. Regisztráció (csak egyedi felhasználónév szükséges, hogy meg tudjuk különböztetni a kísérlet résztvetőit).
2. A korábban meghatározott 17 faktor értékelésének leadása (0-3 közötti egész szám).
3. Döntés, hogy mely város látványosságairól szeretne ajánlást kapni (Budapest, London vagy Párizs).
4. A továbbiakban a rendszer az adott városhoz tartozó összes látványosságra kiszámítja a felhasználó 17 faktorra adott értékelése alapján a látványosságokhoz tartozó pontszámokat (a *KPI1-2-3* és *4* eljárásra egyenként), majd a listát csökkenő sorrendbe rendezi, és a 15 legjobb pontszámú látványosságot adjuk ajánlásként a felhasználónak. Pontegyezés esetén külön figyelembe vesszük azt, hogy az adott helyszín legrelevánsabb faktora(i)t mennyire értékelte a felhasználó. Így tehát mind a 4 kalkuláció végeredményeként egy 15 helyszínből álló ajánlást teszünk. A kísérletben résztvevő alanyok a 4 eljárás által adott értékeléseikkel adnak visszajelzést a kalkuláció jóságára vonatkozóan, vagyis arra, mennyire képes jól megragadni a felhasználó ízlésvilágát. Az ajánlások értékelése során a 15 elemű lista sorrendjét nem kell figyelembe venniük az alanyoknak.
5. A felhasználó kiértékeli 4 különböző módon kalkulált ajánlást (1-10 közötti egész számmal). Amennyiben a válaszadó nem ismeri az egyik ajánlott helyszínt, a nevére kattintva elnavigálja a felhasználót a látványosságot ismertető wikipedia oldalra, ezzel segítve őt a döntésben.
6. Opcionálisan más város látványosságairól is kérhet további ajánlásokat, melyeket értékelhet.
7. A látványosságokat egyesével is értékelheti (1-10 közötti egész szám). Ennek a kutatás egy későbbi fázisában még jelentősége lesz az adatbázisban megadott kezdeti relevancia értékek pontosításában, valamint egy majdani kollaboratív szűrési technikán alapuló ajánlórendszer kialakításában. Erről bővebben a kutatási tervek szakaszban szólunk.

A vizsgálat alá vont 4 eljárás közül azt értékeljük a legjobbnak, melyre a felhasználók szignifikánsan magasabb érétkelést adtak, mint a többire. A fenti vizsgálat eddigi eredményeit a következő szakaszban ismertetjük.

3.9. Empirikus eredmények értékelése

Az internetes kérdőívet az összegzésig 59-en töltötték ki legalább egy várost értékelve. A kitöltések városok közötti megoszlását, valamint az egyes kalkulációs eljárások átlagos értékeléseit (Budapest - külföld bontásban) a *15. ábrán* láthatjuk. A vizsgálatot 2016.02.02. és 2016.02.10. közötti időszakban végeztük.

15. ábra: A kitöltések városok közötti megoszlása, valamint az egyes kalkulációs eljárások átlagos értékelései



A Budapestre külön- (50), valamint a Londonra és Párizsra együtt (33) adott értékelések száma alapján elérjük a statisztikai értelemben vett nagyminta követelményét, ám kevéssel haladják meg azt. Érdekes jelenség, hogy míg Budapesten nem tűnik egyértelműnek, hogy mely eljárás a legsikeresebb, addig a Londonra és Párizsra adott értékelések alapján kijelenthetjük, hogy az első eljárás által adott ajánlásokat értékelték a legjobbnak, hiszen az érte el a legjobb átlagos helyezést is (1,85) valamint a legmagasabb átlagos értékelést is (8,33). Mindeközben Budapesten a legutolsó

helyen végzett a *KPI1*, és egymással szoros versenyben ugyan, de a *KPI2* bizonyult a legjobbnak. A jelenség hátterében vélhetően az áll, hogy a kitöltők szinte kivétel nélkül magyarok voltak, így számukra Budapesten a kiemelt látványosságok helyett (melyet jellemzően az első eljárás ajánl) sokkal érdekesebbek lehetnek a kevésbé ismert látnivalók, így jobban értékelték a 2. és 3. eljárást, melyek célja, hogy akár a helyi lakosok számára is valami újat ajánljanak. Azok, akik Párizsra és Londonra (is) kitöltötték a kérdőívet, az esetek többségében az első eljárás ajánlásai voltak a legszimpatikusabbak, hiszen a listában javarészt olyan helyszíneket láttak viszont, melyet már nem voltak ismeretlenek számukra, és szívesen megnéznék azt, míg a 2. és 3. eljárás sok esetben ajánlott ismeretlen helyeket. Noha az összes ajánlott helyszín wikipedia oldala elérhető volt számukra a kérdőívből egyetlen kattintással, tartok tőle, hogy ezzel a lehetőséggel kevesen éltek, így az ismeretlen helyszíneket ajánló eljárásokat lepontozták. Összegezve tehát a teljes populációra nézve Budapest esetén az eljárások sorrendje (értékelés és rangsor alapján számolva egyaránt): 2-3-4-1, míg küldöldön ez éppen ellentétes: 1-4-3-2.

8. Táblázat: Azonosított turista típusok

Turisztikai faktorok	Kultúra kedvelő	Természet kedvelő	Családos	Fiatal	Mondén	Gurmé
Múzeum/művészet	x					
Természet/park		x				
Építészet/épület/homlokzat	x					
Történelem/kultúra	x					
Vásárlás/divat				x	x	
Kilátópont		x				
Kiemelkedő látványosság				x		
Éjszakai élet/zene/bár				x	x	
Piac/helyi ételek				x		x
Utcák/terek				x		x
Tört. helyszín/emlékmű	x					
Fürdő/sport/rekreáció				x		x
Színház/mozi/szórakozás			x	x	x	
Kávézó/étterem				x	x	x
Templom/vallási témájú hely	x					
Egyetem/tudomány/techn.			x	x		
Család/gyermek program			x			

A faktorokra adott értékelések alapján azonban tovább finomíthatjuk az eredményeinket, és lehetőségünk nyílik megérteni az egyes turisták motivációit.

A faktorok közötti összefüggések megértése érdekében képezzük azok korrelációs mátrixát, melyet a *D.1*-es *mellékletben* közlünk, (zölddel jelöltük a 0,4 fölötti korrelációs együtthatókat, és sárgával a 0,3-0,4 közöttieket). Az egymással csoporton belül korreláló faktorok alapján 6 turista típust tudtunk azonosítani, melyeket a *8. táblázatban* összegeztünk, jelölve a velük szorosan összefüggő faktorokat. Az együtt mozgó faktorok, illetve az ezek alapján kialakított csoportok nagyrészt plauzibilisek, bár az elnevezések nem mindenhol elég találóak, hiszen nehéz egyetlen szóban egybesüríteni a mögötte rejlő koncepciót. Meglepőnek mondható talán, hogy az általunk fiatalok elfoglaltságaiként azonosított 9 faktor szinte kivétel nélkül relatíve erősen korrelál a másik 8-cal. Néhány között azonban erősebb kapcsolatban áll fent, ennek szemléltetésére alakítottuk ki az utolsó két típust, mely a 9 faktor egy-egy részhalmazából épül fel. Az így klasszifikált turista típusok összsokaságon belüli megoszlását a *D.2*-es *mellékletben* láthatjuk. Valakit egy adott csoportba tartozónak minősítettünk, ha az adott csoport faktoraira adott átlagos pontszáma elérte a 2,2-et (a természet kedvelőknél ez a küszöbérték 2,5, mert csak 2 faktor szerepel benne). Mivel az általunk képzett 6 csoport bármelyikébe való tartozás nem zárja ki, hogy egy másikba is tartozzon az illető, ezért akadnak olyanok, akik kettő, vagy akár három csoportba is tartoznak. Az egyes csoportok tagjai által az eljárásokra adott értékelések viszont jól elválnak egymástól. A pontozások részleteit a *D.3*-as *mellékletben* adjuk meg, és a 9. *táblázatban* foglaljuk össze, milyen sorrendet állíthatunk fel az eljárások között az egyes csoportok esetében.

9. Táblázat: Azonosított turista típusok eljárásokra vonatkozó értékeléseinek sorrendje

Turisztikai faktorok	Eljárások	Kultúra kedvelő	Természet kedvelő	Családos	Fiatal	Mondén	Gurmé
Bp	KPI1	3	4	2	4	4	4
	KPI2	1	1	1	3	3	2
	KPI3	2	2	4	2	2	3
	KPI4	4	3	3	1	1	1
London - Párizs	KPI1	1	1	1	1	1	1
	KPI2	4	4	4	3	4	3
	KPI3	3	3	3	4	2	2
	KPI4	2	2	2	2	3	4

Jól látható, hogy a külföldi értékelések körében minden esetben az első eljárás volt a legnépszerűbb, míg a budapesti értékelések esetén 3-3 csoport értékelte legjobbnak a 2. és 4. eljárást. A csoportok között előforduló átfedések miatt azonban esetenként nehéz eldönteni, mely eljárással adható a legjobb ajánlás. Szűkös mintánkon végzett szimulációink alapján az alábbi mechanizmus bizonyult a legsikeresebbnek: Külföldi helyszín esetén válasszuk az 1. eljárást. Budapest esetén ha valaki összes leadott értékelése meghaladja a 2,2 pontot, akkor adjuk ismét az 1. ajánlást. Ha művészet kedvelőként klasszifikáltuk (akár egyebek mellett), adjuk a 2. ajánlást. Ha nem művészet kedvelő, de fiatalként klasszifikáltuk, adjuk a 4. ajánlást. Ha a fentiek közül egyik sem, adjuk a 2. ajánlást. Ezzel a mechanizmussal a 88 esetből 76-ban adtuk a legjobbra értékelt ajánlást, és csak 2 esetben nem a 2. legjobbat. Ha a priori minden esetben a móduszt ajánljuk (tehát Budapest esetén a 2. eljárást, míg külföld esetén az 1. eljárást), akkor 88 esetből csak 36 esetben adtuk volna a felhasználó számára legjobb ajánlást. A döntési mechanizmus ábráját a *D.4*-es *mellékletben* láthatjuk. Fontos megjegyezni, hogy ez nem egy döntési fa.

3.10. Konklúzió és kutatási tervek

Jelen fejezetben bemutattuk az ajánlórendszerek kiterjedt szakirodalmát, és néhány gyakorlati példát azok turisztikai célú felhasználására. Az általunk készített tulajdonság kiterjesztő hibrid ajánlórendszer tudás alapú- és tartalom alapú moduljainak köszönhetően igen kevés kezdeti információ alapján is képes ajánlást tenni, így megbirkózik az ajánlórendszerek legnagyobb kezdeti nehézségeivel. Az empirikus vizsgálat során gyűjtött értékelések alapján sort kerítettünk a turisták típusok szerinti klasszifikálására is, melynek segítségével pontosítani tudtuk ajánlásainkat, ám kutatásainkat koránt sem tartjuk lezártnak.

A kutatás egy későbbi szakaszában, a látványosságokra adott egyedi értékelések bővülése esetén szükségét látom a korábban már röviden ismertetett, két további eljárás vizsgálatának. Ez egyik kollaboratív ajánló eljárás lenne, mely elsőként az adott felhasználóhoz hasonló ízlésvilágú személyeket keres. Ezt könnyen megtehetjük például az amazonnál látott vektor-cosinus eljárással [90], ha a 17 faktorra adott értékeléseket adott sorrendben vektorba rendezzük, és ezeknek a vektoroknak a hajlásszögét tekintjük a hasonlóság alapjának. Minél kisebb szöget zár be két felhasználó vektora, annál inkább hasonlít egymásra ízlésviláguk. Ezután a felhasználónak adott ajánlásokat az alapján állíthatjuk össze, hogy hozzá leginkább hasonló felhasználók a látványosságokra adott egyedi értékeléseik közül melyek voltak átlagosan a legjobbak. Ezzel egy

kollaboratív technikát tudnánk tesztelni, szemben a fenti 4, tudás alapú, hibrid eljárással. Ennek előfeltétele, hogy igen nagy felhasználótömeg álljon rendelkezésre a kísérleti fázisban, mert ennek hiányában nehéz hasonló felhasználókat találni. Amennyiben ez adott esetben nem teljesül, az ajánlások adhatóak az átlagosan jó értékelést kapott helyszínek köréből, vagyis az egyéni ízlést - jobb híján - a közízléssel igyekszünk közelíteni.

A jelenleg használatos 17 faktort kellő mennyiségű adat esetén érdemes lenne felülvizsgálni, és az értékelések alapján mátrixfaktorizációs eljárással feltárni esetleges új faktorokat, illetve a köztük lévő kapcsolatot.

A másik eljárás, melyet vizsgálat alá vonnánk a jövőben, a tartalom alapú szűrés, mely során a felhasználó által korábban jónak értékelt látványossághoz hasonlóakat keresünk, és azt ajánljuk neki. Hasonló látványosságok alatt ismét az azokat reprezentáló, 17 faktorhoz rendelt relevancia értékek vektorait értjük, melyek viszonylag kis szöget zárnak be egymással.

Természetesen nem csak a jelenlegi vizsgálatba bevont 4 eljárás eredményeit versenyeztethetjük a fent körvonalazott másik kettővel, hanem akár egy azokból előállított hibrid eljárás megalkotása is a további vizsgálat célkitűzései között szerepel. Előfordulhat ugyanis egyrészt az is, hogy bizonyos felhasználói szegmensekre az egyik - általában nem kiemelkedően jól teljesítő - eljárás jobb eredményt a többinél, és ez egy *váltogató hibrid* eljáráshoz vezethet, vagy akár megalkothatunk egy *súlyozott* vagy *kevert hibrid* eljárást is a fentiek felhasználásával.

4. Útvonaltervezés

4.1. Bevezetés

A dolgozat 3-as tagolásának utolsó pilléreként az útvonaltervező algoritmus kerül megalkotásra, felhasználva az előző fejezetekben kapott eredményeket. A valamilyen szempont szerint optimális útvonal(ak) algoritmizált keresése jellemzően az úthálózatot és csomópontokat leképező irányított vagy irányítatlan gráfon történik, ahol a csomópontokban (vagy éppen az éleken) profitokat érhetünk el azok felkeresésével, míg a csúcsok között megtett utaknak élköltségei vannak. A 2. fejezet eredményei alapján képesek vagyunk az egyén teljesítményéhez igazodó menetidőbecslére, melyből a gráfunk élköltségeit származtatjuk. Továbbá a 3. fejezetben megalkotott ajánló rendszerünk képessé tesz minket arra, hogy személyre szabott profitokat rendeljünk minden csúcshoz a gráfon. Ezeket a paramétereket a fejezet további részében adottságnak tekintjük.

Ebben a fejezetben a továbbiakban olvashatnak a szerző személyes motivációjáról, ami a témaválasztásban befolyásolta, majd a 3. szakaszban bemutatjuk a témához kapcsolódó korábbi eredményeket és köztük fennálló összefüggéseket. A 4. szakaszban kerül megalkotásra az útvonaltervező algoritmus, melyet az eredmények kiértékelése követ, végül a 6. szakaszban összefoglaljuk eredményeinket, és kijelöljük a kutatás további lehetséges irányait.

4.2. Motiváció és a téma relevanciája

Mint azt már korábban említettem, az évek során volt alkalmam igen sok idegen országot, ismeretlen várost bejárni eddigi életemben. Az idő szűke miatt gyakorta volt ez sietős, és tűnt - a magam hibájából - inkább feladatnak, semmint kedvtelésnek. Útjaim során, amikor csak tehettem, a DK Eyewitness könyveit használtam a szükséges információk begyűjtésére, mert logikus struktúrájában könnyű eligazodni, a helyszíneket pedig geográfiai- és téma szerinti csoportokba sorolják. Nagy hátránya ezzel együtt, hogy - ha csak nem áll rendelkezésre elektronikus formában - cipelni kell az út során, és még segítségével is csak térképen tudunk útvonalakat rajzolni, hogy merre is menjünk a helyszínek látogatása során. Ha ez mind nem lenne elegendő, még azt is észben kellene tartanunk, mi mikor van nyitva, nehogy egy olyan napon és napszakban érjünk majd oda, mikor a kérdéses hely éppen zárva van. Egy 3 napos út megtervezése papíron, akár csak egy Budapest méretű városban is komoly kihívást jelent, és nem kevés munkaórát. Egy ilyen utat minimális személyes információ megosztása árán automatikusan generálni képes algoritmus, vagy a későbbiekben alkalmazás, nyilvánvalóan tervezgetéssel töltött órák százezreit takaríthatja meg az

emberek számára, akik azon fáradoznak, hogy azt a néhány pihenésre szánt napot a számukra leginkább élvezetessé tegyék. Mindezek mellett, ha jobban meggondoljuk, egy 3 napos úton is legfeljebb napi 10 órát tölt aktívan városnézéssel valaki. Ebből a 30 órából akár csak 2 órát megspórolva, amit fölösleges kitérőkkel, és tévutakkal töltünk, értékes, új élményekhez juthatunk. Nem is beszélve arról, hogy ismeretlen helyszínek egész sorát kellene átnézni egy utikönyvben vagy az interneten ahhoz, hogy a számunkra leginkább relevánsakat megtaláljuk, holott - ahogy ezt az ajánlórendszereknél láthattuk - csekély információt kell kiadnunk preferenciáinkról, és máris egy kész, egyéni igényekre szabott listát kaphatunk. Jelen fejezet célja egy erre a feladatra alkalmas algoritmus megtervezése.

4.3. Kapcsolódó szakirodalom

4.3.1. A legrövidebb út problémája

Az útvonaltervező algoritmusok szakirodalma messzire nyúlik vissza, hiszen már az őskorban is foglalkoztatta elődeinket, akárcsak az állatokat, hogyan tudnak a leggyorsabban, vagy leginkább energiatakarékos módon eljutni az élelem- vagy vízforráshoz. Első említést érdemlő mérföldköve a labirintusból való kijutást megoldó mélységi keresés algoritmusa (**Depth-first search**), mely Trémaux nevéhez fűződik. Az eljárás lényege, hogy adott pontból úgy járjunk végig egy gráfot, hogy addig megyünk előre a csomópontokon, míg lehetséges, majd visszelépünk az első olyan csomópontig, ahol elágazás volt, stb. Ez tehát egy mohó algoritmus, mely lokális optimumokon keresztül reméli elérni a globális optimumot. A mélységi bejárás módszeréről Wienernél olvashatunk először 1873-ban [185]. A legrövidebb utakra adott megoldások további történeti áttekintése előtt következzen egy definíció.

Definíció (legrövidebb út irányítatlan gráfon): Legyen $G(V,E)$ irányítatlan gráf, V a csúcsok, E az élek halmaza, míg $P=(v_1,v_2,\ldots,v_n) \in V\times V\times\ldots\times V$ úgy, hogy v_i szomszédos v_{i+1}-gyel $\forall\ 1\leq i<n$, így P egy n hosszú út v_1 és v_n között. Legyen $e_{i,j}$ élköltség v_i és v_j csúcsok között, valamint az éleken legyen adott $f : E\rightarrow\mathbb{R}$ élköltség függvény. Ekkor v_0 és v^* között (ahol $v_0=v_1$ és $v^*=v_n$) a legrövidebb út az, ami minden lehetséges n-re minimalizálja az alábbi kifejezést:

$$\sum_{i=1}^{n-1} f(e_{i,i+1}).$$

Az irányított gráfok esetén csupán annyi a különbség a definícióban, hogy irányított $e_{i,j}$ éleket követelünk meg a szomszédos v_i és v_j csúcsok között. Irányított gráfokra az 50-es években két megoldás is született. Ezen eljárásokban közös az alábbi, Ford [186] által leírt általános forma:

Legyen adott $G(V,E)$ irányított gráfon az $f : E \rightarrow \mathbb{R}$ élköltség függvény, és két csúcs közötti távolságot definiáló függvény, $d : E \times E \rightarrow \mathbb{R}$. Ekkor egy adott s csúcsból egy másik csúcsig tartó út hosszát az alábbiak szerint kalkuláljuk: legyen $d(s)=0$ és $d(v_i)=\infty \ \forall \ v_i \in V/s$. Válasszuk (v_j, v_i) élt, ahol $d(v_i) > d(v_j) + f(v_j, v_i)$ és legyen $d(v_i) := d(v_j) + f(v_j, v_i)$, majd folytassuk ezt addig, amíg már nem találunk ilyen élt. A két módszer közötti különbség ott van, ahogyan az iterációban a következő élt kiválasztjuk:

- A **Bellman-Ford algoritmusban** minden iterációban végigmegyünk az éleken, míg el nem fogynak, összesen maximum $|V|$ darab iterációban. Ezt a módszert (vagy ezzel ekvivalens módszert) írt le egymástól függetlenül Shimbel 1955-ben [187], Bellman 1958-ban [188] és Moore 1959-ben [189]. Shimbel telefonhálózatok mátrix reprezentációján igyekezett megoldani a legrövidebb út problémját.
- A **Dijkstra** által 1959-ben közzétett algoritmusban [191] mindig a legkisebb $d(v_j)$ értékhez tartozó (v_j, v_i) élt választjuk, így minden él legalább egyszer kiválasztásra kerül, ha nincsenek negatív élköltségek. Ezzel ekvivalens megoldást írtak le Leyzorek et al. [190], a Case Institute of Technology kutatói is 1957-es riportjukban, és Shimbel korábbi eredményének komplexitásán is tudtak javítani javaslatukkal. Hasonló, és csak kicsit lassabb algoritmus az 1958-ban Dantzig cikkében megjelent módszer [192], amely szerint azt az élt kell választani a következő lépésben, amelyre a $d(v_j) + f(v_j, v_i)$ érték minimális.

Mint látható az évszámok közelségéből is, a korszak igen termékeny volt, a megoldások pedig kis túlzással egyszerűek, hiszen több kutató egymástól függetlenül is ekvivalens eredményre jutott. A korszakról bővebben Schrijver cikkében olvashatunk [193].

Rövid kitérő erejéig meg kell említenünk a témával kapcsolatosan a *minimális feszítőfa problémát*, mely egy összefüggő, irányítatlan gráfban a legkisebb összélköltségű feszítőfát keresi. (Feszítőfa alatt azt a fát értjük, amely a gráf összes csúcsát tartalmazza, élei a gráf eredeti élei, és minden csúcsból, minden csúcsba pontosan egy út vezet). A problémára már 1926-ban adott egy megoldást Boruvka [257], melynek egy egyszerűsített változatát írta meg Jarník 1929-es levelében Boruvkának, majd 1930-ban cseh nyelven cikk formájában is megjelent [194]. Ám ez feledésbe merült, és tőle függetlenül Prim 1957-ben [195], valamint Dijkstra 1959-ben ismét megalkották az eljárást [191], ezzel sikerült javítaniuk **Kruskal** 1956-ban megjelent megoldásának számításigényén

[196], melyet Boruvka nyomán írt. Az eljárás igen egyszerű (**Prim-algoritmus**): legyen *G(V,E)* összefüggő, irányítatlan gráf, valamint jelölje *A* a keresett feszítőfa csúcsainak halmazát, míg *B* az élek halmazát. Válasszunk tetszőleges csúcsot *V*-ből, töröljük *V*-ből, és kerüljön *A*-ba. Válasszuk ki a legkisebb élköltségű (v_j, v_i) élt úgy, hogy $v_i \in V$ és $v_j \in A$. A kiválasztott (v_j, v_i) élt tegyük át *B*-be, és v_j-t töröljük *V*-ből, és tegyük *A*-ba. Ha már *G* gráf minden csúcsa *A*-ban van, akkor megkaptunk egy olyan feszítőfát (tehát nem feltétlenül egyértelmű megoldáshoz jutunk), melynek éleit *B* tartalmazza. Ennek az eljárásnak igen nagy szerepe van többek között közüzemi hálózatok telepítésében.

Visszatérve a legrövidebb út problémához, Dijkstra algoritmusa után sok heurisztikus megoldás szü1etett a teljesítmény javítására. (A heurisztika minden esetben egy függvény, mely rangsorolja a lehetséges megoldásokat az elérhető információk alapján, ezzel segítve a továbblépésnél a gyorsabb döntést). Talán a legismertebb útkereső algoritmus, az $\mathbf{A}^*$ (A-star) is ekkor született 1968-ban a Stanford Research Institute-ban, mely a best-first search [198] eljárást használja heurisztikaként minden iterációban, hogy a lehető leghamarabb megtalálja az optimális utat, lásd Hart et al. [199]. Egyéb heurisztikus megoldások, mint például a $\mathbf{B}^*$ [200] vagy a kétirányú keresés (**bi-directional search**) [201] után, 1987-ben sikerült a számításiigény terén áttörést elérnie Fredmannak és Tarjannak [197], Fibonacci-halmokon (*F-heaps*) alapuló, új adatstruktúrájuknak köszönhetően. A hálózatok bonyolultságának növekedésével nehezen tudta az informatika fejlődése tartani a versenyt, így 2005-ben a 9. alkalommal megrendezett Dimacs Challange [202] nevű tudományos verseny a legrövidebb út témájában írta ki pályázatát, és mintaadatként rendelkezésre bocsátották az USA akkori teljes úthálózatának gráfját. A verseny igen sok új eredményt generált, közülük is kiemelkedő a Karlsruhe Institue of Technology csapata által publikált cikkek sora. A rövidség kedvéért csak egyet, Geistberger et al. [203] cikkét emelném ki, akiknek érdeme abban rejlik, hogy a korábbi eredményeket javítani tudták azzal, hogy a keresés során nem preferált elemeket előzetesen eltávolítják a gráfbol. Eljárásuknak a rövidítési rangsor (**contraction hierarchy**) nevet adták. Gyorsabb megoldásuknak azonban igen komoly előkalkuláció az ára. Ennek kiváltására tett kísérletet Delling et al. [204] **RAPTOR** nevű algoritmusa, mely egyáltalán nem igényel előkalkulációt, és mivel nem Dijkstra-algoritmusán alapszik, minden útat maximum egyszer vesz figyelembe iterációnként. Az előkalkulációk elhagyásával az algoritmus alkalmassá vált online alkalmazásokban való felhasználásra, hogy Pareto-optimális utakat kalkuláljon tömegközlekedési hálózatok felhasználói számára. 2013-as cikkében Dibbert et al. [205] közzétettek **Connection Scan** nevű algoritmusukat, mely bár nem sokkal gyorsabb, mint a RAPTOR, de lényegesen

egyszerűbb, mindamellett képes kezelni komplex eseteket is, például a várható késéseket, és ezt figyelembe véve kalkulálja a felhasználók várható érkezési idejét. A témakörben a 90-es évek közepéig megalkotott algoritmusokat részletesen tárgyalja Cherkassky et al. [206], különös figyelmet fordítva azok számításigényére.

A legrövidebb út problémára adott megoldások történeti áttekintése után térjünk rá az útvonaltervező eljárások gyakorlati problémákon való alkalmazásaira.

4.3.2. Útvonaltervező eljárások

Az egyik első útvonaltervező alkamazás az utazóügynök probléma (*Traveling Salesman Problem,* röviden *TSP*), melyet először az 1930-as években Karl Menger formalizált, és adott rá megoldást [1]. Lényege, hogy az ügynöknek adott telephelyeket kell felkeresnie, és dönteni csak arról tud (az élköltségek ismeretében), milyen sorrendben teszi ezt, hogy a lehető legkisebb költséggel járja körbe a telephelyeket. Tehát minimális összköltségű Hamilton-kört keresünk a gráfon. Birkhoff [257] munkájának köszönhetően lehetővé vált a hozzárendelési feladatok megoldása lineáris programozási feladatként, melyet Dantzig, Fulkerson és Johnson alkalmazott elsőként a TSP megoldására [3]. 1954-es cikkükben olyan módszereket vezetnek be, mely ma kombinatorikus optimalizálás alapját képezik, mint például a metszősíkok módszere. Fontos megemlítenünk, hogy a kombinatorikai és gráfelméleti alapok megteremtéséből olyan magyar tehetségek vették ki részüket, mint Kőnig Dénes a páros gráfok ekvivalencia tételével [4], majd tanítványa, Gallai Tibor független- és lefogó halmazokról szóló tételével [5], és Egerváry Jenő, aki általánosította a Kőnig-tételt [6], majd később a szállítási feladat kapcsán is elért önálló eredményt [7]. A magyar gráfelméleti iskola jelentőségét az is jól mutatja, hogy Kuhn Magyar-módszernek nevezte el az Egerváry munkája nyomán megalkotott, ma is alapvető eljárását a hozzárendelési feladat kombinatorikai megoldására [8]. A témakör tudománytörténeti hátterét bővebben Schrijver dolgozta fel [9].

A későbbiekben is javarészt ipari és gazdasági motivációk vezérelték a kutatások fókuszát, így alakult önálló témakörré a szállítás tervezését segítő jármű útvonaltervezési probléma (*Vehicle Routing Problem,* röviden *VRP),* mely egy teherszállító flotta járműveinek telephely központú körútjainak optimalizálását célozza idő- és kapacitáskorlátok mellett. A probléma első formalizálására Dantzig és Ramser 1959-es cikkében került sor [10]. Később ennek több változata alakult ki: jármű útvonaltervezési probléma időablakokkal (*Vehicle Routing Problem with Time Windows,* röviden *VRPTW*), a korlátozó feltételek kibővültek a meglátogatandó célállomások nyitvatartási idejével vagy éppen a *kapacitáskorlátos jármű útvonaltervezési probléma* esetén a

szállítóeszköz kapacitás korlátjával (*Capacitated Vehicle Routing Problem,* röviden *CVRP*), de több példát láthatunk a feltételek könnyítésére is: a *többutas jármű útvonaltervezési probléma* esetén a teherautók akár több körutat is tehetnek (*Vehicle Routing Problem with Multiple Trips,* röviden *VRPMT*), vagy nem feltétlenül szükséges az út végén a telephelyre visszatérniük a nyílt *jármű útvonaltervezési problémában* (*Open Vehicle Routing Problem,* röviden *OVRP*). Mivel a probléma NP-nehéz, így az idők során megannyi közelítő módszer született, ezek egyik jellemző iránya a heurisztikus megoldások köre:

- genetikus algoritmusok (**genetic algorithm)**, melyek utánozzák a mikrobiológusok által megfigyelt DNS-lánc javításának mechanizmusát, és az első fázisban - jellemzően mohó algoritmus segítségével - elkészült utakat variálják cserék és eltolások sorozatával. Az algoritmus futási ideje erősen függ attól, milyen megállási értéket állítanak be az algoritmusban (vagyis hány olyan random próbát tehet az algoritmus egymás után, ami nem javította a célfüggvény értékét, mielőtt új helyen próbál javulást elérni), lásd Chang és Chen [11].
- a hangya kolóniák módszere (**ant colony system**) a hangyák "motivációs eljárását" igyekszik utánozni: tudvalevő, hogy a hangyák feromonok segítségével kommunikálnak egymással. Amennyiben egy hangyának hosszú utat kell megtenni az élelem forrásáig, úgy egyre gyengül a feromon jel, amit maga után hagy. Ha azonban sikerül rövid utat találnia, ez a jel erős marad, így mind többen járnak majd a megtalált rövid úton. Ezt a logikát alkalmazták Bullnheimer et al. [12] VRP feladat megoldására.
- szimulált lehűlés (**simulated annealing**) egy sztochasztikus technika, mely minden lépésben dönt - megfelelő kritériumok mellett -, hogy egy másik állapotba lépjen-e át, vagy helyben maradjon. A kohászatból vett kifejezés arra utal, ahogyan a fémet ellenőrzött körülmények között felhevítik, majd visszahűtik, hogy a szerkezetét erősítsék, és a benne található zárványokból minél több eltűnjön. Ezzel az eljárással keres globális optimumot VRPTW feladatra Czech és Czarnas [13].
- a tabu keresés (**tabu search**) megoldások a memóriában tárolják azokat a megoldásokat, melyek korábbi iterációkban tesztelve lettek és valamilyen előre megállapított szabály miatt a tiltó listára kerültek (egy időre). Az eljárást például Bräysy és Gendreau alkalmazta VRPTW megoldására [14].
- az **2-opt** általában más algoritmusokkal kombinálva jelenik meg a megoldásokban. Lényege, hogy olyan út, mely keresztezi saját magát, úgy legyen átrendezve, hogy ne legyen benne kereszteződés. Az algoritmus leírását elsőként Croes adta 1958-ban a TSP megoldására [213].

- az **3-opt** olyan helyi keresési (**local search**) algoritmus, mely a gráfon vagy úton 3 szomszédos csúcsot töröl, majd ezeket minden lehetséges módon újra rendezve igyekszik az optimális utat vagy utakat megtalálni. Az algoritmust elsőként Lin formalizálta 1965-ben [214].
- a **Lin-Kernighan-algoritmus** az 2-opt és 3-opt eljárások általánosítása, melyben mindkét algoritmust adaptívan alkalmazzuk az útvonalakon. A Lin és Kernighan [215] által 1973-ban alkotott algoritmus az egyik leghatékonyabb eljárás a TSP megoldására.

Mindemellett egzakt algoritmusok is születtek, mint

- korlátozás és szétválasztás (**branch and bound**) egy kombinatorikus optimalizációs eljárás branch szakaszában a keresési halmazt diszkrét halmazokra bontja bizonyos szabályok alapján, majd a bound szakaszban az egyes halmazokat "ritkítja", ezzel gyorsítva fel a keresést a brute-force megoldásokhoz képest, lásd Bektas et al. [15]
- a vágás és szétválasztás (**branch and cut**) eljárás egészértékű lineáris programozási (Integer Linear Programming, röviden ILP) feladatok megoldására szolgál, melynek keretében először a branch and bound algoritmust használjuk az LP feltételeinek könnyítésére, majd metszősíkok módszerével szűkítjük azokat, hogy az optimumhoz közelebb jussunk. Jó példa ennek alkalmazására VRP feladat megoldásában Pessoa et al. [16].
- az egzakt algoritmusok számításigénye gyakran csökkenthető olyan eljárásokkal, melyek egyszerű megfontolások alapján az irreleváns csúcsokat, vagy csúcs kombinációkat eleve törlik. Erre jó példa Lu et al. [246] Trip-Mine algoritmusa, ahol a csúcsok költség-profit alapú rendezésével, valamint már időben el nem érhető csúcsok törlésével lerövidítik a vizsgálandó esetek számát. Így a vizsgált "brute force" algoritmus (mely 12 csúcs kalkulálása esetén már majdnem 1 órás futási időt produkál) helyett javasolt eljárás néhány ezred másodpercre csökkenti annak futásidejét.
- mivel a VRPTV formalizálható egyenletrendszerként, így a probléma LP feladatként való megoldása is lehetséges, lásd Rousseau et al. [17].

Az utazóügynök problémából kifejlődő másik ág a tájfutó problémája (*Orienteering Problem,* röviden *OP)*, vagy más néven a szelektív utazóügynök probléma (*Selective Traveling Salesman Problem,* röviden *STSP)*, ahol az egyes ügyfelekhez már profitot rendelnek, és az ügynököt szorító időkorláton belül a legnagyobb összprofitot kell begyűjtenie az útja során az ügyfelek meglátogatásával. Az elnevezés 1996-ban Chao et al. [208] cikkében szerepel, de már 1984-ben megjelent Tsiligirides-nél [209], ahol a TSP-ben az ügynöknek nincs elég ideje, hogy az összes várost meglátogassa egyedül. Cikkében olyan sztochasztikus algoritmust alkalmaz az optimális útvonal közelítő megoldására, mely minden iterációban Monte-Carlo-módszerrel keresi a következő

csúcsot, a távolság és a begyűjthető profit függvényében. A problémát már formalizálta Kataoka és Morito 1988-ban [210], ám ők még maximális gyűjtési probléma (*Maximum Collection Problem*) néven hivatkoztak rá. A témáról bővebben Feillet et al. összefoglaló cikkében olvashatunk [18]. Az OP megfogalmazását az *A.1. melléklet* alatt találjuk. Már a kezdetektől ismert volt ennek a technikának a természetjárásban és általában a turizmusban való alkalmazhatósága, hiszen az OP elnevezés is a tájfutásból ered, ahol a versenyzőknek egy térkép és egy iránytű segítségével kell felkeresni az előre kijelölt pontokat a lehető legrövidebb időn belül. Innen datálható a tudományág sport és turizmus területén történő hasznosítása, és terjedt ki nem csak a természetjárásra, de a városnézésre is. Ennek jó példája Wang et al. [211], ahol a legérdekesebb látványosságokat látogatja végig a turista a szállodából indulva, és a nap végén oda érkezik vissza. Golden, Levy és Vohra megmutatták, hogy az OP NP-nehéz [19], így az erre adott egzakt megoldások csak viszonylag kis számú csúcs esetén lehetséges. Ramesh et al. [216] branch-and-bound algoritmust használ, mellyel egzakt megoldást ad akár 150 csúcsot tartalmazó gráfra is, míg Fischetti et al. [217] cikkükben brach-and-bound eljárással akár 500 csúcsra is egzakt megoldást tudnak adni. Ramesh és Brown [218] 4 fázisból álló heurisztikus megoldást adnak az OP-re, melyben az 2-opt és 3-opt eljárásokat alkalmazzák. Ennél jobb eredményeket ad Chao et al. [208] 5 lépésből álló megoldása, mely mohó algoritmust, sztochasztikus eljárást és 2-opt algoritmust ötvözve építi fel az útvonalat. A fenti heurisztikus megoldások egy komoly hátránya, hogy könnyen be tudnak ragadni egy lokális optimumba, melyet Gandreau et al. [219] tabu search megoldása hatékonyan hidal át. Mivel az eredmények turisztikában történő felhasználása igen nagy figyelmet kap, így cikkek sora foglalkozik azok térinformatikai beágyazásával is (mobil applikációk formájában), erre jó példát találunk az OP esetére Souffriau et al. 2008-as cikkében [212]. A tájfutó problémája időablakkal (*Orienteering Problem with Time Windows,* röviden OPTW) az OP általánosítása, ahol a csúcsokhoz nyitvatartási időket rendelünk. Az OP leírását a *A.2. melléklet* alatt adjuk meg. Elsőként Kantor és Rosenwein [220] adtak rá megoldást 1992-ben. Első lépésben úgy illesztenek be az útvonalba új csúcsokat, hogy ne ütközzön időkorlátba, és az egységnyi időköltségre eső fajlagos profitja a lehető legnagyobb legyen. Ezután mélységi keresési algoritmussal állít elő útszakaszokat, majd fűzi őket össze, elhagyva a nem megvalósítható elemeket. Mivel az időablakok miatt a OP-nél hatékonyan alkalmazható 2-opt és 3-opt algoritmusok OPTW esetén nem használhatóak, így annak egzakt megoldására más eljárásra van szükség. Az azonban igaz, hogy az OPTW megoldására használt eljárás alkalmazható az OP megoldására. Ezt megmutatja Tricoire et al. [231] 2010-es cikkükben. Righini és Salani 2009-es cikkében [221] kétirányú dinamikus programozási megoldást

javasol: a kezdő- és végcsúcstól egyszerre kezdik el az út felépítését, végig ellenőrizve, hogy megvalósítható-e az egyes lépésekben javasolt megoldás, ha a két szakaszt összekapcsolnánk.

Az OP egy természetes kiterjesztése a tájfutó csapat probléma (*Team Orienteering Problem,* röviden *TOP)*, ahol a turista "feladata", hogy P nap alatt a rendelkezésére álló időben a lehető legtöbb (számára érdekes) látványosságot meglátogasson, és minden nap végén visszatérjen a szállodájába, (ez igen hasonlít a VRPTW-ben megfogalmazott feladathoz). Ezt először Butt és Cavalier formalizálta 1994-ben [20], ahol egy toborzási feladat megoldására alkalmazták. A TOP megfogalmazását a *A.3. melléklet* alatt találjuk. Az egzakt megoldások közül igen hatékonyan működnek az oszlop generáló algoritmuson [222] alapuló eljárások. Ekkor LP feladatként oldjuk meg a feladatot, de redukáljuk a dimenziók számát a gyorsabb futási idő érdekében, melyre jó példa Butt és Ryan 1999-es cikke [223], ahol akár 100 csúcsra is egzakt megoldást kaphatunk viszonylag rövid idő alatt. Később Boussier et al. [224] alkalmazta az oszlopgeneráló algoritmust, de már kombinálva a branch-and-bound eljárással, hogy javítsanak az algoritmus teljesítményén. A heurisztikus megoldások közül a legkorábbi a már az OP kapcsán ismertetett Chao et al. [208] cikkében szereplő 5 lépcsős eljárás kis átalakítással: itt az első P legjobb utat listázzuk ki eredményül [225]. Tang és Miller-Hooks [226], valamint Archetti et al. [227] is tabu search eljárást alkalmaz az TOP megoldására, míg Ke et al. [228] hangya kolóniák módszerét javasolja cikkében. Az első lépésben 4 eljárást is teszteltek, amivel egy megvalósítható eljáráshoz lehet jutni. Közülük az utakat szekvenciálisan felépítő algoritmus bizonyult a leghatékonyabbnak. Az egyes iterációkban elkészült megoldást 2-opt algoritmussal javítják, majd kiegészítik annyi csúccsal, amennyi az időkorlátba belefér. Vansteenwegen et al. két heurisztikus eljárást is kifejlesztett. Mind az irányított lokális keresés (**Guided Local Search**, röviden GLS) [88], mind a ferde változó szomszéd keresés (**Skewed Variable Neighborhood Search**, röviden SVNS) [230] eljárások ugyanazokon a lépéseken alapulnak: egy kezdeti eljárásbók kiindulva "gyengébb" útszakaszokat törlünk, illetve kisebb útszakaszokat illesztünk össze, majd az így kapott út összprofitját igyekszik javítani cserékkel, illetve a menetidőket csökkenteni, és új pontokat beilleszteni a megtakarított idő terhére. Az SVNS más sorrendben variálja ezeket a lépéseket, és így jóval megelőzi a GLS-t. A TOP időablakokkal általánosított változata a *Team Orienteering Problem with Time Windows (TOPTW)*, melynek leírását a *A.4. mellékletben* adjuk meg. A TOPTW-re adott megoldások közül Vansteenwegen et al. [233] iterált lokális keresés algoritmusa (**Iterated Local Search**, röviden ILS) algoritmusa messze a leggyorsabb, bár akadnak eljárások, melyek átlagosan kicsivel jobb megoldást adnak. Ilyen például Gambardella et al. [263] hangya kolóniák módszerén alapuló eljárása. Tricoire

et al. [231] a TOPTW egy általánosítására, a többperiódusos, több időablakos tájfutó problémájára (*Multi-Period Orienteering problem with Multiple Time Windows,* röviden MPOPMTW) ad heurisztikus megoldást változó szomszéd kereső eljárással (**Variable Neighborhood Search**, röviden VNS*)* [232] eljárással, míg az útvonal megvalósíthatóságának ellenőrzésére egzakt algoritmust javasolnak. Ez esetben az egyes telephelyeknek napok között változó lehet a nyitvatartási ideje. Kísérleteik alapján 100 csúcs és 2 megtervezendő út esetén nagyjából 1 perc alatt jut megoldásra, míg Vansteenwegenék ILS algoritmusával ez 1 másodperc.

4.3.3. Az útvonaltervező eljárások néhány kiterjesztése

A fent ismertetett modelleknek több lehetséges általánosítása létezik, melyek közül a teljesség igénye nélkül néhányat megemlítünk az alábbiakban:

- Az időfüggő tájfutó probléma (*Time-dependent OP,* röviden *TDOP)* lényege, hogy az egyes élköltségek időben változnak. Jól írja le azt a gyakorlati problémát, hogy napszakonként eltérő a városi közlekedés minősége: változik a forgalom és a tömegközlekedési eszközök járatsűrűsége is. Ez talán akkor érint bennünket legkevésbé, ha csak gyalogosan közlekedünk a városban, bár a lámpák beállításai még így is időben változó módon befolyásolja menetidőnket, lásd Fomin és Lingas [237]. Verbeeck et al. [256] hangya kolóniák módszerét kombinálta lokális kereső eljárásokkal a TDOP megoldására. Korábban Abbaspour és Samadzadegan [244] adnak közelítő megoldást a TDOPTW-re genetikus algoritmus segítségével. Az ILS jó kompromisszumot nyújt gyorsaság és pontosság között, de minden csúcsot külön kezel. A időfüggő, időablakos tájfutó csapat problémája (*Time-dependent Team Orienteering Problem with Time Windows,* röviden *TDTOPTW)* megoldása a hagyományos ILS módszerrel már nem lenne hatékony, így García et al. [245] előkalkulációs eljárással visszavezeti TOPTW feladatra, majd ILS algoritmussal oldja meg azt. Egy másik módszert is kidolgoztak, mely nem él az időbeni függés eliminálásával, ám helyette a tömegközlekedés menetrendjére tesznek periodicitási feltevéseket (mely koránt sem realisztikus). Gavalas et al. [262] javasolja az egymáshoz közel eső pontok együtt kezelését a probléma egyszerűsítése érdekében, melyhez k-közép klaszterezés (k-means clustering) eljárást alkalmaznak. Athéni helyszíneket és tömegközlekedést modellező kutatásukban klasztereken alapuló heurisztikus eljárásukat tovább fejlesztve 3 algoritmust is adnak a TDTOPTW közelítésére, melyek az időablakok mellett kezelni tudják az időben változó utiköltségeket és a tömegközlekedési menetrendet is [247]. Az eljárásaik hátránya, hogy nem veszik figyelembe az újabb csúcsok útvonalba történő beillesztésénél a következő helyszín várakozási idejében okozott változást, mikor a beillesztésről döntenek.

- Az általánosított tájfutó probléma (*Generalized Orienteering Problem,* röviden *GOP)* abban különözik az OP-től, hogy célfüggvénye nem pusztán a csúcsokban begyűjthető profitok összessége, hanem általánosabb, nemlineáris összefüggés a pontok között. Lehet például az egyes helyszínek változatosságát extra profittal értékelni (például a negyedik múzeum meglátogatása helyett egy park felkeresése esetén), vagy bizonyos kiegészítő helyszínek megtekintése, például Glasgow-ban Mackintosh múzeummá alakított házának meglátogatása után érdemes felkeresni az általa tervezett Willow Tearooms enteriőrjét. Schilde et al. [234] cikkében a turisták különleges igényeit próbálja leírni nemlineáris célfüggvényekkel. Az Aurigo nevű alkalmazás [250] útvonaltervező algoritmusa igen egyszerű, hiszen csak az épp adott tartózkodási hely egy *r* sugarú környezetében keresi a következő, legnagyobb profitú pontot, de a profitok adaptív módon, dinamikusan kerülnek meghatározásra a felhasználó ízlése, valamint a már meglátogatott pontok függvényében.
- Cikkek sora foglalkozik olyan modellekkel, ahol az egyes élekhez profitok vannak rendelve. Amennyiben a csúcsokhoz nincs, csak az élekhez, azt a szakirodalomban él útvonaltervező probléma (*Arc Routing Problem,* röviden *ARP* vagy *Arc Orienteering Problem, AOP)* néven találjuk. A feladat, hogy két adott pont között a lehető legtöbb profitot begyűjtve haladjunk át éleken, melyeknek költség vonzata is van. Souffriau et al. [249] például az észak-flandriai úthálózaton tesztelte biciklis útvonaltervező mohó véletlenszerű adaptív keresési eljárásukat (**Greedy Randomized Adaptive Search Procedure**, röviden GRASP) eljárását, mely előbb mohó algoritmussal jut egy kezdeti megoldáshoz, majd azt javítja a következő lépésben lokális keresési eljárással. Muyldermans et at. [239] az OP-t kiegészítve élekhez rendelt profitokkal formalizálta az általuk általános útvonaltervezési problémának (*General Routing Problem,* röviden *GRP)* nevezett feladatot, majd adott rá egzakt megoldást 2-opt és 3-opt algoritmusok felhasználásával. A feladat gyakorlati jelentősége a turisztikai célú útvonaltervezésben az lehet, hogy ezáltal a szebb, látványosabb útvonalakat, mint például a sugárutak vagy folyópartok, előnyben részesíthetjük.
- Ha az OP-ben egyes csúcsokat kötelezővé teszünk, az a GOP egy szélsőséges alesetének tekinthető (végtelenül nagy profitokat rendelve bizonyos csúcsokhoz). Gendreau et al. [235] ilyen eljárással biztosítja, hogy a legfontosabb látnivalók minden egyedileg tervezett túraútban benne legyenek.
- Amennyiben az egyes csúcsoknál begyűjthető profitok értéke előre nem ismert, csupán azok eloszlásáról van tudomásunk, az OP-ben megismert feladatunk annyiban módosul, hogy az összprofitunk várható értékét kell maximalizálnunk, melyet sztochasztikus profitú tájfutó

probléma (*Orienteering Problem with Stochastic Profits,* röviden *OPSP)* néven találunk a szakirodalomban. Például Ilhan et al. [236] genetikus algoritmust adott az optimum közelítésére, valamint egy egzakt megodást is, melyben a sztochasztikus célfüggvényt vele ekvivalens, determinisztikus célfüggvényre cserélik, majd súlyozott összeg eljárással (**weigthed sum method**) [237] oldják meg a feladatot.

- A csúcsoknál gyűjthető profitok értéke lehet időben változó, de ismert érték. Ez főleg szállítási feladoknál fordul elő, ahol a késedelmes kiszállítás büntetéssel járhat. Erre adott eljárást Erkut és Zhang [241], ahol a szállítási feladatot időfüggő díjazású maximális gyűjtési probléma (*Maximum Collection Problem with Time Dependent Rewards,* röviden *MCPTDR)* modellel írta le, és a profitok időben lineárisan csökkentek. Ezt egészértékű programozási feladatként kezelték, melyre branch-and-bound algoritmussal és egy mohó algoritmussal adtak közelítő megoldást. Ennek több útra felírt változatára *(Multiple Tour Maximum Collection Problem with Time-Dependent rewards, röviden MTMCPTD)* ad megoldást Tang et el. [242], akik hibaelhárító szerelőcsoportok kiszállásait optimalizálására tabu search algoritmust adtak közelítő megoldásként. A turizmusban olyan gyakorlati esetekben fordulhat elő, mikor egy kiállítás valamely részlege csak szűkebb látogatási időben érhető el, és annak zárva tartása esetén a csúcsnál gyűjthető profit értéke kisebb, vagy mint Erdogan és Laporte cikkében [243], ahol az adott ponton töltött időtől függ a beszedhető profit.
- A TOPTW egy másik általánosítása a szelektív jármű útvonaltervezési probléma időablakokkal (*Selective Vehicle Routing Problem with Time Windows,* röviden *SVRPTW)*, ahol két új korlátot vezethetünk be: a járművek nem csak időkorlátokkal bírnak, de távolságkorláttal is, valamint a rakterükből adódó kapacitáskorláttal. Ezt tetszőlegesen értelmezhetjük turistákra is, akik egy bizonyos távolság megtétele után elfáradnak, valamint anyagi lehetőségük is véges, így nem tudnak naponta egy adott összegnél többet elkölteni a nevezetességeknél megváltandó belépőjegyekre. Boussier et al. [224] korábban említett egzakt algoritmusa erre a problémára is megoldást ad akár 100 csúcs és 10 megtervezendő út esetére is.
- Ennek egy speciális változata a kapacitáskorlátos tájfutó csapat probléma (*Capacitated Team Orienteering Problem,* röviden *CTOP)*, ahol csak egy extra kapacitáskorláttal (pénzügyi korlát) egészítjük ki a TOP modelljét, lásd Archetti et al. [238].
- A turizmusban előforduló gyakorlati problémából fakad a szálloda választó tájfutó probléma (*Orienteering Problem with Hotel Selection,* röviden *OPHS)*, ami a TOP feladat kibővítve azzal, hogy egy adott halmazból szállást kell választani az utakhoz (ahol azok kezdődnek és végződnek), lásd Divsalar et al. [248]. Castro et al. [255] a TSP-t egészíti ki szállodaválasztással

(TSPHS), melyre ILS és egy speciális genetikus algoritmus kombinációjából álló heurisztikus megoldást adnak cikkükben.

- Külön említést érdemel még az útvonaltervező feladatok egy speciális családja, mely a turisták gyakorlati útvonaltervező feladatait kívánja megoldani, és gyakran köthető mobil alkalmazásokhoz, és ebből adódóan kis számításigényű eljárást kíván. Elnevezése, a turistaút tervezési probléma (*Tourist Trip Design Problem,* röviden *TTDP)*, Vansteenwegen és Van Oudheusden 2007-es cikkéből származik [240]. A TTDP legegyszerűbb modellje az OP, és gyakorlati jelentőséget tulajdoníthatunk annak minden kiterjesztésének. A TTDP megoldások részletes áttekintését olvashatjuk Gavalas et al. [35] összefoglaló cikkében. A mobil eszközökre készült alkalmazások jó példája Sylejmani és Dika cikke [24], ahol Bécs turisztikai látványosságain tesztelték tabu search alapú heurisztikus algoritmusukat. García et al. [23] a TDTOPTW megoldására tesznek javaslatot heurisztikus algoritmusukkal, mely személyre szabott profitokkal látja el az egyes csúcsokat a felhasználó preferenciáinak megfelelően. A TDTOPTW mobil alkalmazásokra tervezett megoldások közül Souffriau et al. [21] ILS algoritmussal adott közelítése az egyik leghatékonyabb.
- Az útvonal tervező eljárások egy máshova kevéssé beilleszthető példája De Choudhury et al. [251] cikke, akik "közösségi kenyérmorzsáknak" (social breadcrumbs) nevezett információ alapján építenek túraútvonalakat. Az interneten (Facebook, Flickr, stb.) megosztott fotók és egyéb bejegyzések gyűjtése és szisztematikus válogatása alapján, összeegyeztetve a felhasználó előre kinyilvánított preferenciáival. Mivel a fotókhoz időbélyegek (timestamp) is tartoznak, így Popescu és Grefenstette [252] korábbi munkája alapján már lehetőség nyílt az egyes helyszínek látogatási idejének, illetve a köztük megtett út menetidejének becslésére is. Hasonlóan közösségi adatokon alapszik Letchner et al. [36] munkája, akik helyi lakosok autós GPS adatai alapján jobb útvonalat tudtak javasolni az átutazóknak, mint amit bármilyen útvonaltervező adott, mert ők egy eddig fel nem használt információt építettek a tervezésbe: a tapasztalatot.

Az útvonaltervező algoritmusokról bővebb összefoglalót Vansteenwegen et al. cikkében olvashatunk [22], ahol külön kitérnek az egyes eljárások számítási igényére is. A dolgozatban bemutatott útvonaltervezési problémák közötti kapcsolatot a *E.3-as mellékletben* szemléltetjük.

4.4. Az útvonaltervező algoritmus

Az alábbiakban bemutatásra kerül a dolgozat magját képező útvonaltervező algoritmus, melyhez felhasználjuk a korábbi fejezetek eredményeit is, így a tervezés alapjául szolgáló gráf élköltségeit az útvonaltervező algoritmus segítségével határozzuk meg, míg a helyszíneket jelképező csúcsok értékelései a 3. fejezetben leírt ajánlórendszer segítségével kalkulálhatóak. Elsőként bemutatjuk a tervezéshez felhasznált adatokat, majd a probléma megfogalmazása után egy heurisztikus eljárást adunk annak megoldására.

4.4.1. A felhasznált adatok

Az útvonaltervező algoritmus teszteléséhez létrehoztunk egy adatbázist, mely 150 budapesti turisztikai látványosságot tartalmaz az alábbi adatokkal:

- Helyzeti adatok: a látványosságok szélességi- és hosszúsági koordinátái, 3 méter pontossággal.
- Az adott hely látogatásához szükséges idő percben
- Az egyes helyszínek költségei (belépő díjak), Euroban
- A felhasználó által az egyes helyszínekre adott értékeléseket az útvonaltervezés során adottságnak tekintjük, és feltételezzük, hogy a 3. fejezetben adott eljárás alapján kalkuláltuk, így leírják az adott turista prefenciáit.
- A szálloda (pontosabban annak koordinátái), melyből a turista a nap elején elindul, és a nap végén oda érkezik vissza.

A fentieken túl az OpenStreetMap alkalmazás segítségével, mely tartalmazza a város teljes útvonalhálózatát, kiszámoltuk az összes pont többitől vett távolságát, melyet egy távolság mátrixban foglaltunk össze. Ennek a_{ij} eleme az i pontból a j pontba való leggyorsabb eljutásához szükséges időt jelenti (másodpercben). A két pont közötti legrövidebb utat Dijkstra-algoritmussal számoltuk. Így tehát a várost egy olyan gráffal modellezzük, melynek csúcsai a meglátogatható látványosságok (ide értve a fix szállodát is), valamint az azokat összekötő, időben legrövidebb utak, mint a gráf élei. Az élek költségei az él kezdő- és végpontja közötti menetidők, a csúcsokban pedig a látogatások során gyűjthető profitok (a turista adott csúcsra vonatkozó értékelései), valamint a csúcsnál eltöltendő idők, és belépő díjak jelentik a költségeket.

4.4.2. A turista célfüggvénye

A turisták maximalizálandó célfüggvényéről azért érdemes szót ejteni, mert még a lefrissebb és igen haladó megközelítésekben is, lásd Gavalas et al. [247], vagy Vansteenwegen et al. [230], a feladat nem más, mint a TSP-ben is meghatározott csúcsoknál gyűjthető profitok összegének maximalizálása. Ennek megértése érdekében egy pillanatra tegyük fel, hogy a csúcsoknál begyűjthető profitok lehetséges értékei legyenek az [1;10] intervallumba eső egész számok. Ekkor, ha egy meglátogatott ponthoz igen közel eső, de kis profitú pont meglátogatása mégis jó ötletnek tűnik, hiszen annak az útvonalba történő beillesztése nagy fajlagos profittal kecsegtet. Azonban a gyakorlati problémára koncentrálva ez mégsem jó ötlet, hiszen egy 10-es skálán 2-esre értékelt látnivaló általában nem nevezhető a túrista ízlésvilágával összeegyeztethetőnek. Ez a megközelítés még a TSP megfogalmazása óta része az útvonaltervező algoritmusoknak, ahol pénzben mérhető profitról lévén szó, összeadható volt, és reális elvárás, hogy az összprofitot maximalizáljuk. Helyszínekre adott értékelések esetén azonban ez már nem igaz. Javaslom tehát, hogy ne "pontgyűjtő akcióként" kezeljük a feladatot, és ennek érdekében egy olyan célfüggvényt alakítsunk ki, mely igyekszik garantálni a felhasználót leginkább kielégítő útvonal megtervezését. Az alábbi megfontolásokat tesszük a céfüggvény megalkotása során:

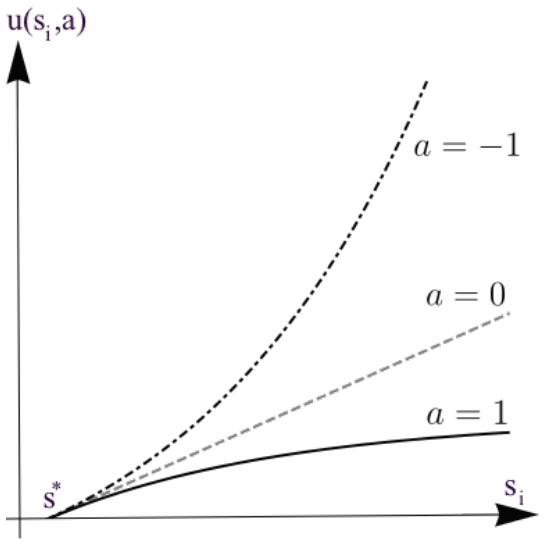
- A turistát a kialakult pontok alapján kevésbé érdeklő nevezetességeket töröljük a listából. Ez egyrészt csökkenti a feladat számítási igényét, másrészt garantálja, hogy csak valóban személyre szabottan kurrens helyszíneket veszünk számításba. A továbbiakban jelöljük s^*-gal azt a minimális értékelést, amit el kell érnie egy csúcsnak a bent maradáshoz, ellenkező esetben töröljük a gráfból.
- A lehető legtöbb időt töltse a turista a helyszíneken, tehát igyekezzünk minimalizálni a csúcsok közötti közlekedésre fordított időt. Ezt egy α paraméterbe építjük be a célfüggvénybe: minél érzékenyebb erre a turista, annál nagyobb büntetést számol fel a látványosságok közötti távolságok megtételéért.
- A turisták különbözhetnek egymástól sétára vonatkozó hajlandóságukban is. Céltalanul hosszú utakat (két pont között) az egyéni preferenciáktól függően sújtsuk külön büntetéssel. Úgy vélem, kevés turista örülne egy két órás sétának két helyszín között. Amennyiben lehetséges, vegyünk fel egy látogatható pontot a hosszabb utakat megtörve. Ennek érdekében a célfüggvényben ne az utazással töltött idők összegét szerepeltessük, hanem azoknak egy 1-nél nagyobb hatványát

szerepeltessük (β "lustasági" paraméter), és azokat adjuk össze. Ezzel ösztönözzük az útvonaltervezésben indokolatlanul hosszú utak felvételét.

- Ne hagyatkozzunk pusztán az egységnyi összköltségre (menetidő + látogatási idő) eső profitra a döntsénél, hiszen így sok időintenzív látnivalót hagyunk ki az útvonaltervezésből: például Párizsban nem javasolnánk meglátogatni az Eiffel-tornyot, mert annak látogatási ideje - hozzávetőlegesen - 2 óra, míg profitja bár igen magas lehet, de egy kicsivel alacsonyabb profitú pont meglátogatása fél óra alatt nagyobb fajlagos haszonnal kecsegtet. Itt javaslom olyan kategória létrehozását, ami az úgynevezett kötelező látnivalókat tartalmazza, melyeket fel kell venni a meglátogatandó helyszínek listájába függetlenül attól, mennyire időintenzívek. Ezek személyre szabottan kerülnek meghatározásra, például az egyén értékelése alapján maximális pontszámot kapott látnivalók lehetnek ezek.
- Legyen az értékelések figyelembevételekor személyre szabható, mennyivel értékel többre az adott felhasználó például egy 9-es értékelésű helyet egy 8-ashoz képest. Az általunk javasolt $u(s_i, a)$ hasznossági függvényt úgy alkottuk meg, hogy a=0 mellett az adott látványosság eredeti s_i értékeléseit adja vissza (illetve azok s^* küszöbértékkel csökkentett értékét), míg $a < 0$ esetén progresszíven nő az értékelések hasznossága. Az $a > 0$ esetben csökkenő határhasznossággal bír az értékelés egységnyi növekedése. Ezek figyelembevételével a következő hasznossági függvény formát javasoljuk (*16. ábra*):

$$u(s_i, a) = \begin{cases} \frac{1-e^{-a(s_i-s^*)}}{a} \mid a \neq 0 \\ s_i - s^* \mid a = 0 \end{cases}$$

16. ábra: Hasznossági függvény



Az *a* elméletben (-∞;∞) intervallumom bármilyen értéket felvehet, gyakorlati megfontolások alapján [-2;2] intervallumban vizsgáljuk majd az útvonaltervre gyakorolt hatását. Mivel *u(s*,a)=0*, így tehát azok a helyszínek, melyek értékelése küszöbértéken van, 0 profitot hoznak, és csak az ennél jobb értékelés helyszínek jelentenek pozitív profitot, melyek növelik a célfüggvény értékét. Ez függvényforma természetesen csak javaslat, ám a kutatás jelen fázisának eredményei alapján reményt keltő annak alkalmazása.

Ezeket figyelembe véve a célfüggvényünk, mely alapján az újabb pontokat veszük fel az útvonalba:

$$C(\alpha,\beta,a,R)=\left(\frac{\sum_{p=1}^{P}\sum_{i=1}^{N}\theta_{ip}v_i}{\sum_{i=1}^{N-1}\sum_{j=2}^{N}\tau_{ijp}t_{ij}^{\beta}}\right)^{\alpha}\times\left(\sum_{p=1}^{P}\sum_{i=1}^{N}\theta_{ip}u(s_i,a)\right)^{1-\alpha}$$

ahol *P* a rendelkezésre álló napok száma, *N* a gráf csúcsainak száma, s_i, v_i, t_i rendre a következő pont értékelése, látogatási ideje és az odaút menetideje, *s** a felhasználó értékeléseinek azon küszöbértéke, ami alatt nem kerülhet be látványosság a potenciálisan látogatható helyszínek közé, α a célfüggvényben szereplő két szempont (a hasznosság és az egységnyi menetidőre eső látogatási idő) súlyozására szolgál, *β* a "lustasági" paraméter. A τ_{ijp} értéke legyen 1, ha a *p*-edik útnál az *i*-edik csúcs után a j-edik következik az úton, és 0 különben. Legyen θ_{ip} értéke 1, ha a *p*-edik úton az *i*-edik csúcsot meglátogatják, és 0 különben. A *P* napra tervezett útvonalak összességét *R* jelöli. Az *a* paraméter hivatott tükrözni, mennyivel értékel többre a felhasználó egy *s*-re értékelt látványosságot egy (*s-1*)-re értékelthez képest.

A választott célfüggvény forma tehát alapvetően két törekvést szolgál: egyrészt az egységnyi megtett útra jutó fajlagos látogatási időt igyekszik növelni, másrészt a legnagyobb hasznossággal bíró csúcsok meglátogatását szorgalmazza. Az α paraméterrel ezek súlyát szabályozhatjuk. Fontos látni, hogy a célfüggvény a mások által széles körben használt profitösszeg-maximalizálás egy kiterjesztése, hiszen α=*a*=0 választással éppen ezt kapjuk.

4.4.3. A feladat formalizálása

Legyen adott egy *G(V,E)* gráf, amelynek minden c_i csúcsához egy s_i nemnegatív értékelés van rendelve, mely a turista számára *u(s_i,a)* hasznossággal bír, ha meglátogatja a c_i csúcsot. A c_i és c_j csúcsok közötti e_{ij} élhez t_{ij} élköltséget rendelünk, ami a turista számára a távolság megtételéhez

szükséges idő. Az c_i csúcs meglátogatása v_i időt vesz igénybe (látogatási idő). Jelölje továbbá h_{ip}, hogy a p-edik útnál az i-edik csúcs hanyadik lépésben kerül sorra az úton, valamint τ_{ijp} értéke legyen 1, ha a p-edik útnál az i-edik csúcs után a j-edik következik az úton, és 0 különben. Legyen θ_{ip} értéke 1, ha a p-edik úton az i-edik csúcsot meglátogatják, és 0 különben. A turistának P napja van a látványosságok megtekintésére, és naponta T_{max} perc ideje. Az i-edik látványosság megtekintése b_i költséggel jár (belépődíj), melyet a napi B költségkeretéből fedezhet. Fontos, hogy a költségvetési korlát tetszés szerint átcsoportosítható a napok között, így összességében a P napra BP költségkerettel rendelkezik. Ez nem vonatkozik az időkorlátra, mely minden napon betartandó. Minden nap elején a szállodából indul, és a nap végén oda érkezik vissza. A modellben ezt az általánosság jegyében külön kezeljük (az *1*-es és *N*-nel jelölt csúcs), de ezek megegyezhetnek egymással. A feladat, hogy P nap alatt olyan P darab utat bejárni a $G(V,E)$ gráfon, hogy maximalizáljuk a célfüggvény értékét, miközben betartjuk az idő- és költségkorlátokat, és minden csúcs legfeljebb egyszer látogatható meg. Ekkor a feladat megfogalmazható a következőképpen:

$$max_{\tau_{ijp}} \left(\frac{\sum_{p=1}^{P} \sum_{i=1}^{N} \theta_{ip} v_i}{\sum_{i=1}^{N-1} \sum_{j=2}^{N} \tau_{ijp} t_{ij}^{\beta}} \right)^{\alpha} \times \left(\sum_{p=1}^{P} \sum_{i=1}^{N} \theta_{ip} u(s_i, a) \right)^{1-\alpha}$$

$$\sum_{p=1}^{P} \sum_{j=2}^{N} \tau_{1jp} = \sum_{p=1}^{P} \sum_{i=1}^{N-1} \tau_{iNp} = P$$

$$\sum_{p=1}^{P} \theta_{kp} \leqslant 1; \forall k = 2, ..., N-1$$

$$\sum_{j=2}^{N} \tau_{kjp} = \sum_{i=1}^{N-1} \tau_{ikp} = \theta_{kp}; \forall k = 2, ..., N-1; \forall p = 1, ..., P$$

$$\sum_{i=1}^{N-1} \sum_{j=2}^{N} \tau_{ijp} t_{ij} + \sum_{i=1}^{N} \theta_{ip} v_i \leqslant T_{max}; \forall p = 1, ..., P$$

$$\sum_{p=1}^{P} \sum_{i=1}^{N} \theta_{ip} b_i \leq PB$$

$$h_{ip} - h_{jp} + 1 \leqslant (N-1)(1 - \tau_{ijp}); \forall i, j = 2, ..., N; \forall p = 1, ..., P$$

$$2 \leqslant h_{ip} \leqslant N; \forall i = 2, ..., N; \forall p = 1, ..., P$$

$$\tau_{ijp}, \theta_{ip} \in \{0, 1\} \forall i, j = 1, ..., N; \forall p = 1, ..., P$$

Az egyes kifejezések jelentése a következő:

1. A maximalizálandó célfüggvény
2. Minden út az *1*-es csúcsnál kezdődik, és az *N*-ediknél ér véget, (ezek a korábbiak alapján megegyezhetnek).
3. Minden csúcsot csak legfeljebb egyszer látogatunk meg.
4. Minden út egyenként összefüggő.
5. Betartjuk az időkorlátot: a napi látogatási- és menetidők összege nem lehet több, mint T_{max}.
6. Betartjuk a költségkorlátot: a belépődíjak összege a *P* napra együttesen nem lehet több *BP*-nél.
7. és 8. együtt garantálja, hogy ne legyenek körök az útban, Miller–Tucker–Zemlin javaslata alapján [207].
9. A τ_{ijp} és θ_{ip} értékkészlete 0 vagy 1.

A kitűzött feladatra adott heurisztikus megoldásunkat a következő alfejezetben ismertetjük.

4.4.4. Az útvonaltervezés

Mivel a megoldandó feladatunk megfeleltethető a TOP egy speciális esetének, így az NP-nehéz feladat, vagyis csak igen kis méretű gráf esetén van reményünk egzakt eljárással optimális megoldásra jutni, éppen ezért egy heurisztikus eljárást javasolunk. Célunk gyakorlati megfontolásokon alapszik: egyrészt szeretnénk egy valóságos problémákon alkalmazható eljárást adni, így az eljárás futási idejét szeretnénk alacsonyan tartani, másrészt olyan eljárást keresünk, mely a felhasználók számára kielégítő megoldással szolgál. Ennek köszönhető a rendhagyónak számító hasznosságfüggvényünk is.

Először két olyan eljárást ismertetünk, melyet több ponton használunk majd az algoritmus során:

Lexikografikus rendezés: ennek során mindig egy csúcsokból álló halmazt rendezünk egy másik csúcsokból álló halmaz és az erőforrás keretek (pénz és idő) szűkössége alapján (mely meghatározásának menetét később pontosan ismertetjük). Szükségünk van továbbá arra az *s** küszöbértékre, melynél alacsonyabb értékelésű csúcsokat törölni fogunk a releváns pontok halmazából (lásd, az algoritmus első lépése). Formálisan tehát $L(C_1, C_2, sc, s^*)$, ahol C_1 a csúcsok azon halmaza, melyet rendezni szeretnénk, C_2 amely halmazhoz rendezzük, *sc* az erőforrások szűkösségének mértékét állítja sorrendbe (idő vagy pénz), és *s** az értékelések küszöbértéke. Képezzük az alábbi értékeket:

$$\frac{\frac{u(s_i,a)}{u(s^*+1,a)}}{d^*(c_i,C_2)+v_i} \qquad \frac{\frac{u(s_i,a)}{u(s^*+1,a)}}{b_i}$$

ahol $d^*(c_i,C_2)$ a c_i csúcs és a C_2 halmaz közötti átlagos távolságot jelöli. A számláló tehát azt fejezi ki, hogy az adott s_i értékelésű pont hány s^*+1 értékelésű pont hasznosságával egyenértékű (a releváns pontok között ugyanis s^*+1 értékelésű a minimális, hiszen az s^* értékelésűeket és az annál kisebbeket töröljük a gráfból). Ezt osztjuk a nevezőben a pont felvételének várható költségével, ami az odaút menetideje plusz a látogatási idő. A pénzben kifejezett költségek rendezésekor ugyanezen az elven a nevezőben a belépődíj szerepel. A következő lépésben rendezzük a C_1 halmaz csúcsait, először aszerint, amelyik korlát szűkösebb. Ha ez például az idő, akkor a fenti hasznosság/ időköltség mutató alapján rendezzük csökkenő sorrendbe, majd az így kapott listát nagyjából 6 egyenlő részre osztjuk kvantilisek segítségével (csak az utolsó csoport elemszáma különbözhet a többitől). A második korláthoz tartozó mutató alapján is sorba rendezzük a csúcsokat a 6 csoporton belül. A csoportosításra azért van szükség, mert az első mutatószám értékei alapján már egyértelműen sorba rendezhetjük általában a csúcsokat, így azok kis eltérése esetén sem lenne lehetőségünk a lexikografikus rendezésnél a második mutató alapján felülvizsgálni a sorrendet.

Outlier számítás: Az outlier kereső eljárásunk $O(H, cr)$ egy adott H Hamilton-kör outlier értékeit adja meg egy cr kritikus időérték mellett. Meghatározzuk H minden csúcsára a ki- és bemenő élek összidejét, majd azok átlagát és szórását. Azon i elemeket tartjuk outliernek, melyek esetén

$$t_{i,be} + t_{i,ki} > t_H^* + cr\sigma_H$$

ahol t_H^* az átlagos be- és kimenő élköltség és σ_H a szórás. A cr értéke az algoritmus egyes lépéseinél változhat. Ezt külön jelezzük majd.

Heurisztikus algoritmusunk az alábbi lépésekből áll:

1. A probléma egyszerűsítése: töröljük a gráf minden csúcsát (az azokba bemenő és azokból kimenő élekkel együtt), melynek értékelése kisebb vagy egyenlő s^* küszöbértéknél, melyet a gyakorlatban választhatunk úgy, hogy a megmaradó csúcsok összes látogatási ideje ne haladja meg a PT_{max} rendelkezésre álló összidőkeret kétszeresét. Gyakorlati tapasztalatunk alapján az ennél több pont szerepeltetése nem javít az optimumon, ellenben az algoritmus számítási igényét fölöslegesen növeli. Ennél általában konzervatívabb megoldás, ha s^* értékét úgy választjuk, hogy megegyezzen az értékelések átlagával, hiszen ha $s_i < s^*$, akkor $u(s_i,a) < 0$, tehát nem javíthat a célfüggvény

értéken. (Az általunk vizsgált esetekben mindig volt annyi átlagon felüli értékelésű csúcs, hogy azok összes látogatási ideje meghaladja a PT_{max} rendelkezésre álló összidőkeret kétszeresét.) Az így kapott halmazt a továbbiakban a releváns csúcsok halmazának nevezzük.

2. Fix csúcsok: Rögzítsük a kötelezően meglátogatandó csúcsokat. Ezek az eljárás során soha nem kerülhetnek az útvonalból törlésre. A korábbi megegyezés alapján azokat tekintjük kötelezőnek, melyekre vonatkozóan a turista értékelése maximális volt.

3. Csoportosítás: A releváns csúcsok alapján megbecsüljük, melyik erőforrás korlátunk szűkösebb. Ennek érdekében összevetjük az alábbi két hányadost:

- a releváns csúcsok látogatási idejéhez hozzáadjuk a releváns csúcsok közötti átlagos távolságot, és ezt az összeget szorozzuk a releváns csúcsok számával, majd elosztjuk a rendelkezésre álló PT_{max} időkerettel,
- a releváns csúcsok látogatási költségének összegét osztjuk a BP költségvetési kerettel.

Amelyik érték nagyobb, azt tekintjük szűkösebb korlátnak, és a lexikografikus rendezések alkalmával a pontszámok után rögtön azt a korlátot vesszük előre. Ha tehát a szűkös korlát az idő, akkor a rendezésnél az alábbi sorrendet vesszük figyelembe a változók körében: értékelés, idő, pénz. A maximális pontszámú (tehát kötelező) csúcsok mellé a maradék releváns csúcsra lexikografikus rendezés után választjuk az első $5P$ darab csúcsot. Ez az egyetlen olyan lépés, ahol a fent ismertetett lexikografikus elrendezéstől eltértünk annyiban, hogy első rendező elvként a pontszámot használtuk. A releváns pontok halmazának outlier csúcsait rendhagyó módon egy a teljes halmazra meghatározott legrövidebb Hamilton-kör meghatározásával kezdjük (melynek kezdő- és végpontja a szálloda). Ebből $cr = 1$ értékválasztás mellett használjuk az $O(H, cr)$ függvényt az outlierek kiszűrésére (természetesen csak a nem maximális értékelésű csúcsok lehetnek outlierek). Azért ezt az eljárást választottuk, mert bár eshet távol néhány csúcs a "központtól", ám ha oda egy viszonylag rövid élköltségekből felépíthető út vezet, akkor tapasztalataink alapján nem érdemes rögtön eldobni. Jó példa erre a Városliget, míg tipikus outliernek nevezhető a Nagytétényi kastély.

4. Napi utak építése: A megmaradó csúcsokat P darab (napok száma) klaszterre bontjuk Hartigan-Wong klaszterező eljárással [265], melyet relatív hatékonysága miatt választottunk. Minden klaszterre kiszámoljuk a szállodával alkotott legrövidebb Hamilton-kört a TSP megoldására adott algoritmussal. Ennek hátterében az a megfontolás áll, hogy amennyiben csak egy fix csúcsokból

álló halmazon szeretnénk a célfüggvényünket maximalizálni, az ekvivalens a csúcsokat összekötő élek élköltségeinek β-adik hatványaival vett gráfon történő legrövidebb Hamilton-kör meghatározásával, hiszen a csúcsok változatlansága miatt mind a hasznosságok, mind a látogatási idők értéke állandó az adott halmazra. Ezt kihasználva az R szoftverben beépített TSP optimalizáló (Repetitive Nearest Neighbor Algorithm) eljárást alkalmaztuk [266]. Ezt a klaszterező eljárást 10-szer ismételjük meg, hiszen a klaszerezés gyakran vezet különböző eredményre. Az 10 eredmény közül azt választjuk, ahol P darab Hamilton-körre számolt célfüggvény értékünk maximális.

5. Feltöltés: Az előző lépésben P darab utat kaptunk, mely a szállodánál kezdődik és ott ér véget. Amennyiben még nem értük el a napi idő- és pénzkeret 1,2-szeresét, akkor az $L(C_r, C_i, sc, s^*)$ eljárással rendezzük az i-edik napra a releváns csúcsok halmazát, melyeket még egy útba sem illesztettünk be (jelölje ezt C_r). Itt fontos megemlíteni két elvet:

- Azokat a csúcsokat, melyeket egy adott napra beillesztünk, automatikusan kivesszük a még megmaradt releváns csúcsok C_r halmazából.
- Ha egy csúcsot kiveszünk egy napból, azt automatikusan visszarakjuk a C_r halmazba.

Így minden napra rendeztük C_r elemeit, és a lista elejéről kezdve elkezdjük feltölteni a csúcsokkal a napokat, amíg el nem érjük az idő- és pénzkorlát 1,2-szeresét. Amennyiben egy csúcs két nap szerinti rendezésben is bekerülne az útba, oda helyezzük, ahol magasabb marginális célfüggvény javulást eredményez. Az i-edik napra való felvétel kritériuma, hogy az így keletkező Hamilton-körben az adott pont ne legyen outlier, ahol az $O(H, cr)$ függvényt $cr = 1{,}5$ mellett értékeljük ki.

6. Csere: Minden nap csúcsaira meghatározzuk a többi nap pontjaitól vett 3 legkisebb érték átlagát, ezt a csúcs saját napjára is kiszámítjuk (ahol a másodiktól a negyedik legkisebb értékig vesszük az átlagot, hiszen a legkisebb érték, az önmagával vett távolság, ami 0). Ezután minden csúcsot arra a napra helyezünk át, hol ez az érték minimális. Ezt az iterációt 10-szer ismételjük meg egymás után.

7. Levágás: Ha van olyan nap, ahol meghaladtuk a napi időkeretet több mint 5%-kal (ennyit engedélyezünk legfeljebb), akkor az $L(C_i, C_i, sc, s^*)$ alapján (vagyis saját magával) rendezve a naphoz tartozó csúcsok halmazát az utolsó elemeket addig távolítjuk el a napi útból, míg az időkihasználása a keret 105%-ánál nem lesz kevesebb. A pénzkorlát túllépése esetén az(oka)t a ponto(ka)t távolítjuk el, ahol az egységnyi pénzköltségre eső marginális célfüggvény növekedés minimális

8. Feltöltés: Amennyiben van olyan nap, ahol még van szabad időkapacitás, a C_r elemeit $L(C_r, C_i, sc, s^*)$ eljárással rendezzük az i-edik napra, és az első elemtől kezdve elkezdjük a napot feltölteni,

míg az időkorlátot és a *P* napra szánt költségvetési korlátot át nem lépjük. Itt ismét a felvétel kritériumaként az *O(H, 1,5)* függvényt alkalmazzuk, mint korábban.

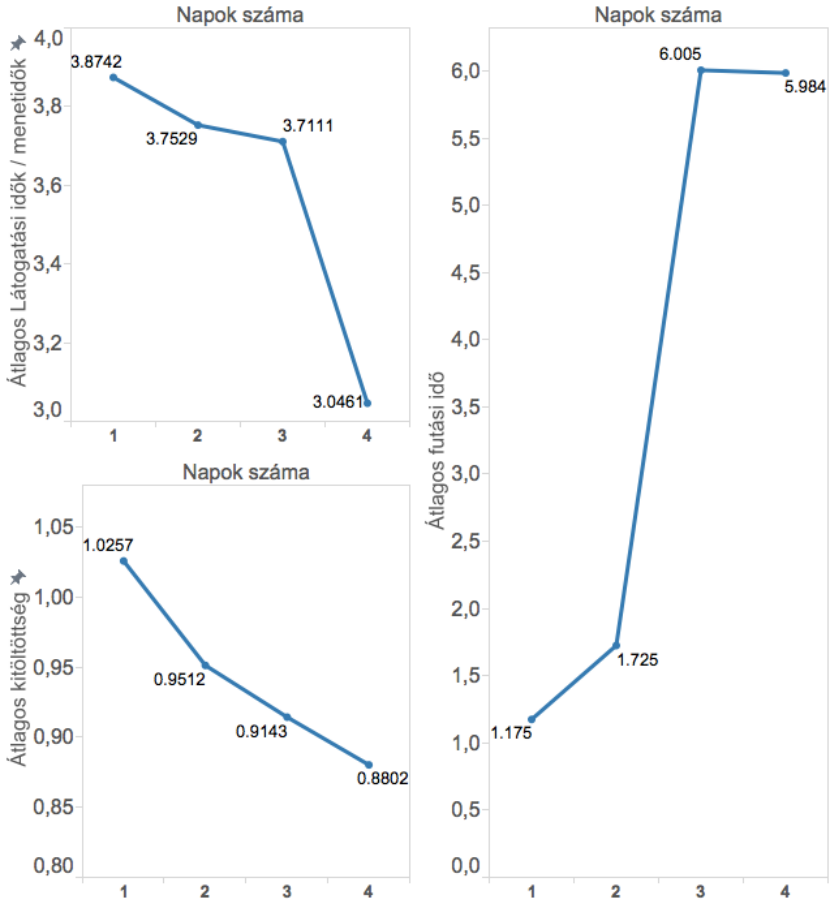
4.5. Az eredmények kiértékelése

A fenti algoritmust a 150 budapesti turisztikai látványosságot tartalmazó adathalmazon tesztelve az alábbi megállapításokat tehetjük:

- Pozitív *a* értékek (konkáv hasznossági görbe) jellemzően nagy kitérőket eredményeznek
- Amennyiben ezek alacsony α és *β* értékekkel párosulnak, úgy az utak "szétesőek", tehát viszonylag kevés pontot, és nagy élköltségeket tartalmaznak.
- Viszonylag kis α értékek (0,5 alatt) csak magas (>1,5) *β* és alacsony *a* (< - 0,5) értékek mellett ad "jó" megoldást.
- Általában elmondható, hogy *a* < - 0,5; α > 0,5 és *β* >1,5 esetén kaptunk jó megoldásokat.
- Az α = 0,75; *a* = -1 és *β* = 2 választása mellett adja összességében 1-2-3 és 4 napos túrákra a legjobb megoldást (hosszabb túrákat nem vizsgáltunk), ahol a napok feltöltöttsége is igen jó, és a teljes túrákra kalkulált látogatási idő - menetidő hányados is kiemelkedően magas. Ezzel a paraméter kombinációval tervezett 4 napos túrát mutatunk be az *E.1 mellékletben*, ahol összehasonlításként szerepeltetjük α = *a* = 0 esetet, mely a szakirodalomban széles körben elterjedt profitösszeg-maximalizálást jelenti. Ennek tanulsága alapján a profitösszeg maximalizálás jóval alacsonyabb látogatási idő - menetidő arányt eredményez (átlagosan 1,43 szemben az általunk kiemelt eset átlagosan 3,8-es eredményével), a napok kitöltöttsége mindkét esetben átlagosan 95% körüli, és a futási idő érthető módon átlagosan nagyjából 1 másodperccel hosszabb az általunk választott paraméterek esetében.
- Az algoritmus 3 napos útvonal megtervezését 100-szor futtatva átlagosan 3,73 másodperc alatt végezte el. Sajnos ezt nem tudjuk összevetni Vansteenwegen et al. [142] vagy Gavalas et al. [140] eredményeivel, hiszen merőben más feladatot oldottunk meg, sőt azok optimális megoldása számolható LP feladatként, míg célfüggvényünk miatt a miénk egy NLP feladat. Az összevetés kedvéért egy másik mobil applikációhoz tervezett heurisztikus eljárást említve Sylejmani és Dika 2011-es cikkében [24] 40 látványosságra tervezett 3 napos túrájuk számítási ideje átlagosan 81,7 másodperc volt Tabu search algoritmussal.
- A futási időről általában elmondhatjuk, hogy egy 4 napos túra megtervezése is átlagosan 6 másodpercen belül tartható. Mivel már az első lépésben csökkentjük a gráf méretét, kijelölve a

releváns csúcsok részhalmazát, ezért a napok számától (P) és a napi időkeretektől (T_{max}) függ a probléma mérete. Amennyiben egy olyan furcsa városban végeznénk a kísérletet, ahol minden csúcs meglátogatása pénzköltséggel jár, ez esetben a pénzügyi korlátunk is ugyanúgy lehet effektív, mint esetünkben az időkorlát. Az túrák megtervezésének átlagos számítási idejét a *17. ábrán* foglaljuk össze. A teszteket az alábbi paraméterekkel rendelkező laptopon futtattuk: 3,8 GB RAM, Intel Core i3-3217U CPU, 1.80GHz × 4 processzor. Minden paraméter kombinációra 20 alkalommal végeztünk el a tesztet, és az eredményeket (outlier értékektől való tisztítás nélkül) átlagoltuk. A futási idők érdekessége a 2- és 3 napos útvonalak tervezése közötti nagy eltérés az átlagos futási időben, míg ez nem növekszik 4 nap megtervezése esetén. Az *E.2. mellékletben* összefoglaltuk az a, α és β paraméterek függvényében is a futási időket a napok száma szerinti bontásban. Az a értékének változása látszólag nincs hatással a futási időre és a β paraméter is csak 3 és 4 napos túrák esetén növeli kis mértékben azt. Az α növekedése azonban 3 és 4 napos túrák esetén drasztikusan növeli a futási időt, mintegy 3-szorosára. Itt vélhetően az áll a háttérben, hogy ilyenkor egyre nehezebb olyan pontokat találnia az algoritmusnak, mely növelni tudná a célfüggvény értékét.

17. ábra: Az útvonaltervező algoritmus eredményei



- A napok kitöltöttsége csökkenő tendenciát mutat a napok számának növelésével (*17. ábra*). Az összes általunk vizsgált esetre átlagosan 94,3%-os értéket mértünk. A paraméterek közül csak az α paraméter növekedése van negatív hatással a napok kitöltöttségére (*E.2 melléklet*), hiszen itt a célfüggvény hasznosság tényezője egyre kevésbé számít, így nehezebb olyan csúcsokat találni, melynek hasznossága ellensúlyozni tudja a látogatási idő - menetidő hányadosban bekövetkező romlást.
- A látogatási idő - menetidő hányados (mely a *P* napra együtt értendő) a napok számának növekedésével csökkenő tendenciát mutat, hiszen egyre nagyobb távolságra találjuk a következő csúcsokat, amelyeket még felvehetünk az utakba. Az *a* és *β* paraméter változása csekély hatással van a hányadosra, az α paraméter növelésével azonban drasztikusan növelhető a háyados értéke.

4.6. Empirikus vizsgálat

Az általunk tervezett algoritmus jóságának mércéje hagyományosan az lenne, hogy mennyire tud az optimálishoz közeli eredményekkel szolgálni. Ennek megadása azonban nehezen értelmezhető, tekintve, hogy a szakirodalomban használatostól merőben eltérő feladatot definiáltunk a célfüggvénynek köszönhetően. Mivel megoldásunk legfőbb célja személyreszabott túrautak tervezése, és ezen keresztül a felhasználók elégedettségének maximalizálása, így a heurisztikus eljárásunk végső fokmérőjének is a felhasználók által kinyilvánított értékelést tekintjük.

A kutatás jelen szakaszának egyik legfontosabb erdménye - mely egyben meg is különbözteti az összes eddig javasolt eljárástól - az a célfüggvény, mely reményeink szerint alkalmas arra, hogy jól, vagy az eddigieknél jobban írja le a felhasználó céljait. Ezeket szem előtt tartva összehasonlítást végeztünk az általunk 4.4.4-es alfejezetben leírt eljárás és az utazóügynök probléma óta a területen széleskörben alkalmazott pontösszeg maximalizáló eljárás között. Tehát a két útvonaltervező algoritmus megegyezik a fent ismertetett algoritmussal, csupán a célfüggvényt változtatjuk. Az első esetben a saját paraméterbeállításaink szerint α = 0,75 és *a* = -1 értékeket választjuk, míg a második esetben α = 0 és *a* = 0 értékválasztással az alábbi célfüggvényhez jutunk:

$$C(0,\beta,0,R) = \sum_{p=1}^{P}\sum_{i=1}^{N}\theta_{ip}s_i$$

ami nem más, mint az utazóügynök problémából jól ismert célfüggvény, ahol minden meglátogatott csúcsban felvesszük az ott gyűjthető profitot, és célunk a túra végén a legnagyobb profitösszeg elérése. A vizsgálathoz létrehoztunk egy a 3.8-as alfejezetben ismertetett vizsgálathoz hasonló adatbázist és weboldalt. A felhasználók négy városra (Budapest, London, Párizs és Róma) tölthették ki a kérdőívet a *travelscheduletest.hopto.org* oldalon. A vizsgálat alapját képező nagyjából 950 látványosságot tartalmazó adatbázis struktúráját tekintve megegyezik a 3.8.2-ben ismertetett adatbázissal (lásd *7. ábra*), kiegészítve a helyszíneken töltendő időkkel (vizit idő), a helyszíneken fizetendő belépődíjakkal (amennyiben nem ingyenes), valamint a szélességi és hosszúsági értékekkel. A várost leképező gráfot a 4.4.1-es alfejezetben ismertetett módon állítottuk elő mind a 4 város esetében.

A felhasználók feladata, hogy a megadott preferenciáik és egyéb paraméterek alapján kalkulált 2 útvonaltervet (a tervbe felvett látványosságok és az őket összekötő útvonalak alapján) értékeljék 1-10-ig, ahol az 1-es a legkevésbé felel meg az ízlésüknek, míg a 10-es a számukra leginkább tetsző megoldást jelenti. A kitöltés az alábbi lépések szerint zajlott:

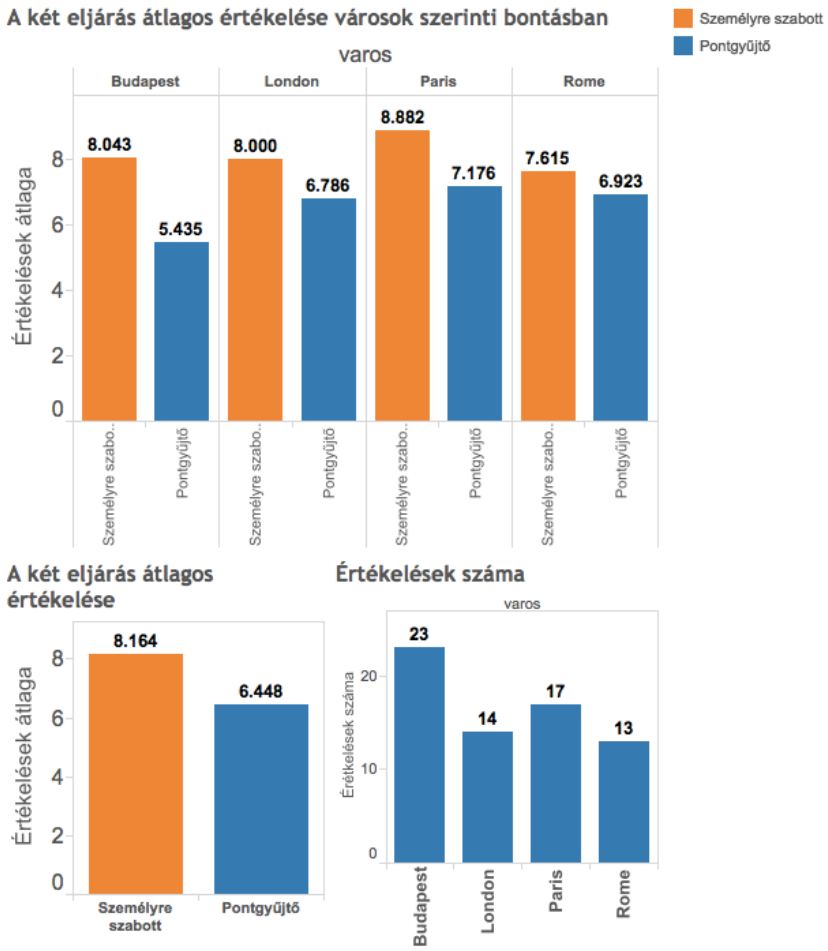
1. Regisztráció (csak egyedi felhasználónév szükséges, hogy meg tudjuk különböztetni a kísérlet résztvetőit).
2. Döntés arról, hogy mely város látványosságairól szeretne ajánlást kapni (Budapest, London, Párizs vagy Róma).
3. A 3.8.1-es alfejezben meghatározott 17 faktor (lásd *6. ábra*) értékelésének leadása (1-4 közötti egész szám).
4. Döntés arról, hogy hány napot szeretne a városban tölteni (1, 2, 3 vagy 4).
5. Beállítja, hogy naponta hány órát töltene városnézéssel (6, 7, 8, 9 vagy 10).
6. Döntés arról, maximum hány Eurót szánna a napi belépődíjakra (50 és 500 Euró között).
7. Beállítja, mennyire szívesen sétál a városban (1-es lehető legkevesebbet, 2-es, ha szívesen sétál, de nem szereti a nagyon hosszú túrákat, és 3-as, ha kedveli azokat). Ez a β lustasági paraméter beállításához szükséges, melyet a $\beta = 3{,}5 - érték/3$ képlet alapján kalkuláltunk, tehát például 3-as értékadás esetén $\beta = 2{,}5$ lesz a paraméterbeállítás.
8. Választhat szállodát (ez városonként 6 opcióból lehetséges, melyeket térképen jelöltünk).
9. A továbbiakban a rendszer az adott városhoz tartozó összes látványosságra kiszámítja a felhasználó 17 faktorra adott értékelése alapján a látványosságokhoz tartozó pontszámokat (a 3.9-es alfejezetben megadott mechanizmus alapján), majd ezeket felhasználva kalkulálja a beállított paraméterek alapján az útvonalakat mindkét célfüggvény esetén.

10. A felhasználó kiértékeli 2 különböző módon kalkulált ajánlást (1-10 közötti egész számmal). Amennyiben a válaszadó nem ismeri az egyik ajánlott helyszínt, a nevére kattintva elnavigálja a felhasználót a látványosságot ismertető wikipedia oldalra, ezzel segítve őt a döntésben. Szöveges értékelést is adhattak a kitöltők, hasznos tanácsaikkal és észrevételeeikkel segítve a kutatás további fázisait.
11. Opcionálisan más város látványosságairól is kérhet további ajánlásokat, melyeket értékelhet.

A vizsgálat alá vont két eljárás közül azt értékeljük jobbnak, melyre a felhasználók a másiknál szignifikánsan magasabb érétkelést adtak.

A kérdőívet 67 felhasználó töltötte ki, és az általuk adott értékelések alapán kijelenthetjük, hogy az általunk tervezett célfüggvénnyel kalkulált útvonalak átlagosan szignifikánsan jobb eredményt értek el. A két megoldásra adott értékeléseket természetesen t-tesztnek is alávetettük, mely alátámasztja a fentieket (*E.4-es melléklet*). A *18. ábrán* összefoglaljuk az értékelések végéredményét város szerinti bontásban és a teljes populációra nézve is.

18. ábra: A két eljárás által tervezett útvonalakra adott értékelések



Vizsgálatunk eredménye alapján kijelenthetjük, hogy az általunk tervezett célfüggvényt használó algoritmus szignifikánsan jobb útvonalakat generál a felhasználók számára, mint a korábban széles körben alkalmazott pontösszeg maximalizáló eljárás a teljes populációra nézve és minden városban külön-külön is. Ezzel elértük kezdeti célkitűzésünket, hiszen olyan személyre szabott útvonalakat tervező algoritmus megalkotása volt feladatunk, mely a felhasználóktól kapott minimális információ alapján képes a preferenciáiknak leginkább megfelelő ajánlásokat tenni. A fejezet zárásaként az alábbiakban a kutatás jelen szakaszának konklúzióját vonjuk le, valamint kijelöljük az előttünk álló fontosabb kutatási irányokat.

4.7. Következtetések és lehetséges kutatási irányok kijelölése

Az előző alfejezetben bemutatásra került a TOP megoldására adott heurisztikus algoritmusunk, mely bár az alkalmazott módszerekben is sok ponton eltér a szakirodalomban található megoldásoktól, mégis legnagyobb vívmánya az a hasznossági- és célfüggvény, mely gyakorlatias megközelítésben a felhasználó preferenciáit tartja szem előtt a “pontgyűjtéssel” szemben. Az eredmények értékelése igen nehéz, hiszen célunk olyan útvonaltervező algoritmus megalkotása volt, mely a felhasználók személyreszabott igényei (helyszínek értékelése) alapján képes másodpercek alatt, számukra megfelelő útvonalat tervezni. Az algoritmus eredményeként előállított attraktív, személyre szabott útvonaltervek, valamint a néhány másodperces futásiidők lehetővé teszik a megoldás gyakorlati alkalmazását. Bár esetünkben is értelmezhető az optimális megoldástól való eltérés mértéke, ennek kiszámítása mégis akadályokba ütközik (NP-nehéz problémáról lévén szó), másrészt nehezen vethető össze más kutatások eredményeivel, hiszen eltérő célfüggvénnyel dolgoztunk. Az útvonaltervezés értékelésének egy lehetséges módjának tekintjük, ha az algoritmusunk által készített útvonalakat és egy a szakirodalomban szereplő másik eljrás eredményeit értékeltetjük tesztalanyokkal, hiszen az ő értékeléseikre tekintünk az útvonalterv jóságának végső fokmérőjeként. Vizsgálatunk során saját megoldásunkat vetettük össze a mások által használt montösszeg maximalizáló célfüggvényre épülő eljárással. A felhasználók értékelései alapján az általunk javasolt megoldás szignifikánsan jobb útvonalterveket készít, mint a másik tesztelt módszer.

Az alábbiakban megfogalmazunk néhány további tervet a kutatás folytatását illetően:

- A kezdeti útvonalak klaszterezésen alapuló kialakítása helyett jó megoldás lehet P darab egymástól kellő távolságra található csúcs kijelölése: ezek egyrészt a kötelező pontok lehetnek, másrészt egyéb, magas értékelésű csúcsok. Az egyes napokra ezekből kiindulva építhetünk fákat,

melyeket úttá rendezhetünk át Lin−Kernighan-algoritmussal. A klaszterező eljárás ugyanis nem mindig vezet ugyanarra az eredményre, ezért is van szükségünk az algoritmus kezdeti lépésében 10 ilyen iterációra. Az iterációk számát növelve ugyan biztosíthatjuk a jó klaszterezést, ám jelentősen növeljük vele a futási időt (20 klaszterezéssel már átlagosan további 2,5 másodperccel).

- A jelenlegi kutatási szakaszban egyetlen α, β, a kombinációt emeltünk ki, mint optimálisnak tűnő megoldást. Ezzel magasan tudjuk tartani a napok kitöltöttségét, jellemzően a legmagasabb értékelésű csúcsokon rendre áthalad az útvonal, és alacsony élköltségeivel a célfüggvény értékét magasan tartja. Amennyiben a napok kitöltöttségét és az egységnyi megtett útra eső látogatási időt (vagy hasznosságot) elfogadjuk az útvonaltervezés jóságának fokmérőjeként, akkor lehetőség nyílna rögzített a érték mellett az optimális α és β kombinációk meghatározására, sőt akár megadható lenne β az α függvényében, és ezzel csökkenthetjük a paraméterek számát.
- Az előbbi gondolatmeneten tovább haladva, legalább két ilyen függvény definiálása is szükséges lenne, hiszen β eredendően "lustasági" paraméter, így annak különböző értékei mellett más-más felhasználói igényeket tudunk kiszolgálni (továbbra is szem előtt tartva az útvonalak optimalitására tett törekvéseket).
- Fontos megemlíteni, hogy a célfüggvény konstrukciójából adódóan az utolsó, feltöltő lépésben könnyen lehet, hogy már nem tudunk olyan pontot illeszteni bármelyik nap útvonalába, mely ugyan még a korlátok szerint elférne, de rontana a célfüggvény értékén. Ez akkor következhet be, ha a marginális hasznosságnövelésével nem tudja kompenzálni az addigi átlagos látogatási idő - menetidő arányban okozott romlást. Mivel célfüggvényünk maximalizálása mellett az időkorlát kitöltését is fontos szempontnak tartottuk, így egy valós gyakorlati megoldás esetén akár a célfüggvény rovására is feltölthetjük a napokat további pontokkal. Mi most ettől eltekintettünk.
- A jelenlegi algoritmus egy kézenfekvő és a gyakorlatban szükséges kiterjesztése lenne az időablakok kezelése, vagyis alkalmassá tenni az eljárást a TOPTW megoldására. A jövőbeli kutatás egyik fő mérföldkövének tekinthető ennek megvalósítása.
- További gyakorlati fontossága lenne az algoritmus kiegészítésének, hogy adott árkategóriában választani tudjunk a rendelkezésre álló szállodák közül, és az algoritmus által köréjük szervezett P napos túrák közül a számára legkedvezőbbet választhatja ki, vagy megteheti ezt helyette az algoritmus is.

5. A kutatási eredmények összegzése

A dolgozatban az alábbi új tudományos eredmények kerültek bemutatásra:

1. **Földfelszín modellek kidolgozása**

   Hogyan lehet a GPS alapú túranaplók adatait pontosabbá és elemzésre alkalmassá tenni? Létezik olyan digitális emelkedési modell (DEM), mely a valós magasság értékeket jól közelíti?

   A GPS alapú túranaplók szélességi és hosszúsági adatainak korrekcióját Kálmán-filter alkalmazásával végeztük. Mivel a GPS magasság adatai nem kellő pontosságúak, így azokat egy digitális magasság modellből kell származtatnunk. A földfelszín közelítésére a NASA által közzétett adatok alapján **két DEM modellt is alkottunk: az egyik bilineáris interpoláción alapszik, és négyzetekkel közelíti a földfelszínt, míg a másik háromszögekkel teszi azt. A teszteredményeink alapján mindkét modell igen közeli eredményt ad a jelenlegi piaci sztenderdnek tekintett Google DEM modelljéhez.**

2. **A túraszakaszok meredeksége és a túrázó sebessége közötti összefüggés meghatározása**

   Létezik a sebesség és az adott szakasz meredeksége közötti összefüggést leíró ismert megoldásoknál pontosabb?

   A Waldo Tobler által 1993-ban közzétett tisztán meredekség alapú sebességbecslő eljárás óta nem született olyan becslési megoldás, mely ezt az összefüggést felülvizsgálná, pontosítaná. A rendelkezésünkre álló **2400 túranapló alapján illesztett meredekség-sebesség összefüggést leíró görbe eredményeink alapján jobb menetidőbecslést tesz lehetővé, mint a Tobler-görbe.** Kiemelendő, hogy az általunk tanulmányozott kutatások során használt adathalmazok mennyisége és minősége sem közelíti meg a vizsgálataink során felhasznált adathalmazét.

3. **Menetidőbecslő eljárások megalkotása**

   Milyen eljárással adható a túrázók menetidejének egy a jelenlegieknél pontosabb becslése?

Menetidőbecslő eljárások évezredek óta léteznek, ám az elmúlt mintegy 120 évben a Naismith-féle ökölszabályt alkalmazták a gyakorlatban a túrázók menetidejének közelítésére. Ennek megannyi pontosítása született az idők folyamán, ennek legutóbbi mérföldköve Pittman et al. 2012-es eredménye, mely átlagosan 18%-os pontossággal becsül menetidőt túraútvonalakra. Az általnk javasolt két eljárás dinamikusan gyűjt adatot a túra közben és adaptálja azt a becslés során. **Az átlagsebességen alapuló eljárás átlagosan 12,94%-os pontosságú, míg a meredekség alapú eljárás 11,58%-os átlagos hibával működik** a vizsgált 2400 túranapló alapján. Ez az általunk ismert eddigi legpontosabb eljárás.

4. Turisztikai célú hibrid ajánlórendszer

Hogyan lehet kevés kezdeti információ alapján turisztikai célú ajánlórendszert építeni, mely személyre szabott ajánlásokat tesz a felhasználónak?

Mivel nem áll rendelkezésünkre a kutatás jelen fázisában egy olyan kiterjedt adatbázis, mely tartalmazná a felhasználók látványosságokra vonatkozó értékeléseit, így 3 európai fővárosra építettünk saját adatbázist, valamint egy 17 faktorból álló listát, melyek kombinációjával szeretnénk leírni az egyes helyszíneket (**tartalom alapú modul**). A **tudás alapú modul** a látványosságokat leíró faktorokra vonatkozó értékeléseket gyűjti be a felhasználótól, így a **tulajdonság kiterjesztő technikával integrált modulokból előálló hibrid rendszer** képes ezen igen csekély kezdeti információból is személyre szabott ajánlásokat tenni.

5. A turisták klasszifikálása

Lehetséges a turistákat klasszifikálni a kezdetben magukkal kapcsolatban megadott információk alapján? Hogyan építhető fel ennek érdekében egy személyre szabott ajánlásokat adó rendszer kérdőíve?

Empirikus vizsgálatunkhoz egy weboldalt hoztunk létre, ahol az ajánlásokhoz csak az **általunk megalkotott 17 faktorra** kell értékelést adjanak a felhasználók. Ezek alapján **6 különböző turista típust sikerült azonosítanunk**, melyek közül az 3-3 hasonlóan rendezte sorba a felkínált 4 ajánlási eljárás eredményét. A csoportokról elmondható, hogy az őket leíró faktorok (melyekre jellemzően magas értékelést adtak) szoros összefüggésben vannak

egymással, erre utal magas korrelációs együttható értékük. A turisták klasszifikálása segítséget nyújt az egyes felhasználók számára tett ajánlások további pontosításában.

6. Célfüggvény a TOP feladathoz

Milyen hasznosságfüggvénnyel írható le általános módon a felhasználók látványosságokra vonatkozó értékelése? Milyen célfüggvény vezet a gyakorlatban a felhasználók számára leginkább elfogadható útvonaltervezéshez?

Az útvonaltervező algoritmusok a TSP óta hagyományosan a csúcsokban gyűjthető profitok összegét tekintik az optimalizálandó célfüggvénynek, ez azonban figyelmen kívül hagy jópár gyakorlati megfontolást, éppen ezért ritkán vezet jó eredményre. Ilyen gyakorlati megfontolás például, hogy a felhasználó által alacsony értékelést kapott pontokat akkor se vegyük be az útvonaltervbe, ha azok igen kis költséggel megtehetőek, vagy éppen az, hogy igyekezzünk fajlagosan a lehető legtöbb időt a helyszínek meglátogatásával tölteni (a gráf élein történő séták helyett). **Javasolt hasznossági függvényünk és célfüggvényünk a korábbi pontösszeg-maximalizálás egy kiterjesztéseként értelmezhető**, hiszen a paraméterek bizonyos értékei mellett (α=a=0) visszakapjuk azt:

$$u(s_i, a) = \begin{cases} \frac{1-e^{-a(s_i - s^*)}}{a} \mid a \neq 0 \\ s_i - s^* \mid a = 0 \end{cases}$$

$$C(\alpha, \beta, a, R) = \left(\frac{\sum_{p=1}^{P} \sum_{i=1}^{N} \theta_{ip} v_i}{\sum_{i=1}^{N-1} \sum_{j=2}^{N} \tau_{ijp} t_{ij}^{\beta}} \right)^{\alpha} \times \left(\sum_{p=1}^{P} \sum_{i=1}^{N} \theta_{ip} u(s_i, a) \right)^{1-\alpha}$$

7. Heurisztikus algoritmus a TOP megoldására

Milyen algoritmus adható a kitűzött TOP feladatra, mely alacsony futási idejével alkalmazhatóvá teszi azt egy valós applikációban?

A TOP feladatra adott heurisztikus algoritmus célja az volt, hogy egyszerűségével, és ebből adódóan rövid futási idejével lehetőséget adjon annak későbbi gyakorlati alkalmazhatóságára. A **4 másodperc alatti eredmény**, valamint a célfüggvénynek köszönhető attraktív útvonaltervek megfelelő alapját képezik egy személyre szabott

túrautakat tervező alkalmazás megalkotásának. Mivel a felhasználói elégedettség optimalizálását tartjuk legfőbb célunknak, így nagy sikerként könyvelhetjük el, hogy a felhasználók körében végzett tesztek alapján az általunk javasolt célfüggvény segítségével **szignifikánsan jobb útvonalterveket tudunk előállítani, mint a mások által használatos pontösszeg maximalizáló eljárással**.

A melléklet

1. Orienteering Problem formalizálása

Legyen adott egy G(V, E) gráf, amelynek minden v_i csúcsához egy π_i nemnegatív profitérték van rendelve, melyet az ügynök megkap, ha meglátogatja a v_i csúcsot, valamint v_i és v_j csúcsok közötti e_{ij} élhez t_{ij} élköltséget rendelünk, ami a távolság megtételéhez szükséges idő. A feladat T_{max} idő alatt maximális pontot összegyűjteni úgy, hogy minden csúcs legfeljebb egyszer látogatható meg. A kezdő- és a végpont fix, és gyakran meg is egyeznek egymással. Jelölje továbbá h_i, hogy az i-edik csúcs hanyadik lépésben kerül sorra az úton, valamint τ_{ij} értéke legyen 1, ha az i-edik csúcs után a j-edik következik az úton, és 0 különben. Ugyan fontos szerepet játszik az egyes csúcsok kiválasztásában az ott töltendő idő is, ám ezt gyakran nem szerepeltetik a modellben, inkább szétosztják a csúcs előtti és utáni élekre (jellemzően fele-fele arányban). Ekkor az OP formalizálása a következőképpen alakul:

$$max \sum_{i=2}^{N-1} \sum_{j=2}^{N} \pi_i \tau_{ij}$$

$$\sum_{j=2}^{N} \tau_{1j} = \sum_{i=1}^{N-1} \tau_{iN} = 1$$

$$\sum_{j=2}^{N} \tau_{kj} = \sum_{i=1}^{N-1} \tau_{ik} \leqslant 1; \forall k = 2, ..., N-1$$

$$\sum_{i=1}^{N-1} \sum_{j=2}^{N} \tau_{ij} t_{ij} \leqslant T_{max}$$

$$h_i - h_j + 1 \leqslant (N-1)(1-\tau_{ij}); \forall i, j = 2, ..., N$$

$$2 \leqslant h_i \leqslant N; \forall i = 2, ..., N$$

$$\tau_{ij} \in \{0, 1\} \forall i, j = 1, ..., N$$

Az egyes kifejezések jelentése a következő:

1. A célfüggvény: a csúcsoknál begyűjtött profitok összege legyen maximális.
2. Az út az *1*-es csúcsnál kezdődik, és az *N*-ediknél ér véget.
3. Az út összefüggő, és minden csúcsot csak legfeljebb egyszer látogatunk meg.
4. Betartjuk az időkorlátot.
5. és 6. együtt garantálja, hogy ne legyenek körök az útban, Miller–Tucker–Zemlin javaslata alapján [207].
7. A τ_{ij} értékkészlete 0 vagy 1.

2. Orienteering Problem with Time Windows formalizálása

A OP-nél leírtaktól annyiban tér el az OPTW, hogy minden csúcsot csak az $[O_i,C_i]$ nyitvatartási ideje alatt lehet meglátogatni, és jelöljük s_i-vel az i-edik csúcshoz való megérkezés időpontját. Ekkor az OPTW leírható az alábbi módon:

$$max \sum_{i=2}^{N-1} \sum_{j=2}^{N} \pi_i \tau_{ij}$$

$$\sum_{j=2}^{N} \tau_{1j} = \sum_{i=1}^{N-1} \tau_{iN} = 1$$

$$\sum_{j=2}^{N} \tau_{kj} = \sum_{i=1}^{N-1} \tau_{ik} \leqslant 1; \forall k = 2, ..., N-1$$

$$\sum_{i=1}^{N-1} \sum_{j=2}^{N} \tau_{ij} t_{ij} \leqslant T_{max}$$

$$s_i + t_{ij} - s_j + 1 \leqslant M(1 - \tau_{ij}); \forall i, j = 1, ..., N$$

$$O_i \leqslant s_i \leqslant C_i; \forall i = 1, ..., N$$

$$\tau_{ij} \in \{0, 1\} \forall i, j = 1, ..., N$$

Látható, hogy az OP-hez képest csupán a körmentesség feltétele változott (itt M egy nagy konstans értéket jelöl), valamint kibővült a nyitvatartási idő korlátjával a feltételrendszer.

3. Team Orienteering Problem formalizálása

Legyen adott egy G(V,E) gráf, amelynek minden v_i csúcsához egy π_i nemnegatív profitérték van rendelve, melyet az ügynök megkap, ha meglátogatja a v_i csúcsot, valamint v_i és v_j csúcsok közötti e_{ij} élhez t_{ij} élköltséget rendelünk, ami a távolság megtételéhez szükséges idő. A feladat T_{max} idő alatt P darab ügynök számára maximális pontot összegyűjteni úgy, hogy minden csúcs legfeljebb egyszer látogatható meg. A kezdő- és a végpont fix, és gyakran meg is egyeznek egymással. Jelölje továbbá h_{ip}, hogy a p-edik útnál az i-edik csúcs hanyadik lépésben kerül sorra az úton, valamint τ_{ijp} értéke legyen 1, ha a p-edik útnál az i-edik csúcs után a j-edik következik az úton, és 0 különben. Legyen θ_{ip} értéke 1, ha a p-edik úton az i-edik csúcsot meglátogatják, és 0 különben. Ekkor a TOP megfogalmazható a következőképpen:

$$max \sum_{p=1}^{P} \sum_{i=2}^{N-1} \pi_i \theta_{ip}$$

$$\sum_{p=1}^{P} \sum_{j=2}^{N} \tau_{1jp} = \sum_{p=1}^{P} \sum_{i=1}^{N-1} \tau_{iNp} = P$$

$$\sum_{p=1}^{P} \theta_{kp} \leqslant 1; \forall k = 2, ..., N-1$$

$$\sum_{j=2}^{N} \tau_{kjp} = \sum_{i=1}^{N-1} \tau_{ikp} = \theta_{kp}; \forall k = 2, ..., N-1; \forall p = 1, ..., P$$

$$\sum_{i=1}^{N-1} \sum_{j=2}^{N} \tau_{ijp} t_{ij} \leqslant T_{max}; \forall p = 1, ..., P$$

$$h_{ip} - h_{jp} + 1 \leqslant (N-1)(1 - \tau_{ijp}); \forall i, j = 2, ..., N; \forall p = 1, ..., P$$

$$2 \leqslant h_{ip} \leqslant N; \forall i = 2, ..., N; \forall p = 1, ..., P$$

$$\tau_{ijp}, \theta_{ip} \in \{0, 1\} \forall i, j = 1, ..., N; \forall p = 1, ..., P$$

Az egyes kifejezések jelentése a következő:

1. A célfüggvény: a csúcsoknál begyűjtött profitok összege legyen maximális az összes utat figyelembe véve.
2. Minden út az *1*-es csúcsnál kezdődik, és az *N*-ediknél ér véget.
3. Minden csúcsot csak legfeljebb egyszer látogatunk meg.
4. Minden út egyenként összefüggő.
5. Betartjuk az időkorlátot.
6. és 7. együtt garantálja, hogy ne legyenek körök az útban, Miller–Tucker–Zemlin javaslata alapján [207].
8. A τ_{ijp} és θ_{ip} értékkészlete 0 vagy 1.

4. Team Orienteering Problem with Time Windows formalizálása

A TOP-nél leírtaktól annyiban tér el a TOPTW, hogy minden csúcsot csak az $[O_i, C_i]$ nyitvatartási ideje alatt lehet meglátogatni, és jelöljük s_{ip}-vel a *p*-edik út során az *i*-edik csúcshoz történő megérkezés időpontját. Ekkor a TOPTW leírható az alábbi módon:

$$max \sum_{p=1}^{P} \sum_{i=2}^{N-1} \pi_i \theta_{ip}$$

$$\sum_{p=1}^{P} \sum_{j=2}^{N} \tau_{1jp} = \sum_{p=1}^{P} \sum_{i=1}^{N-1} \tau_{iNp} = P$$

$$\sum_{p=1}^{P} \theta_{kp} \leqslant 1; \forall k = 2, ..., N-1$$

$$\sum_{j=2}^{N} \tau_{kjp} = \sum_{i=1}^{N-1} \tau_{ikp} = \theta_{kp}; \forall k = 2, ..., N-1; \forall p = 1, ..., P$$

$$\sum_{i=1}^{N-1} \sum_{j=2}^{N} \tau_{ijp} t_{ij} \leqslant T_{max}; \forall p = 1, ..., P$$

$$s_{ip} + t_{ij} - s_{jp} \leqslant M(1 - \tau_{ijp}); \forall i, j = 1, ..., N; \forall p = 1, ..., P$$

$$O_i \leqslant s_{ip} \leqslant C_i; \forall i = 1, ..., N; \forall p = 1, ..., P$$

$$\tau_{ijp}, \theta_{ip} \in \{0, 1\} \forall i, j = 1, ..., N; \forall p = 1, ..., P$$

Látható, hogy az OP-hez képest csupán a körmentesség feltétele változott (itt *M* egy nagy konstans értéket jelöl), valamint kibővült a nyitvatartási idő korlátjával a feltételrendszer.

B melléklet

Legyen ($N_{1,c}$,$N_{2,c}$,$N_{3,c}$,$N_{4,c}$) síkra eső merőleges vetülete rendre ($Q_{1,1}$,$Q_{1,2}$,$Q_{2,1}$,$Q_{2,2}$), és a földfelszín modell által egy adott *x* ponthoz rendelt magasság értéket jelölje *f(x)*. Ekkor a *6. ábra* jelöléseit használva:

$$f(x, y_1) \approx \frac{x_2 - x}{x_2 - x_1} f(Q_{11}) + \frac{x - x_1}{x_2 - x_1} f(Q_{21})$$
$$f(x, y_2) \approx \frac{x_2 - x}{x_2 - x_1} f(Q_{12}) + \frac{x - x_1}{x_2 - x_1} f(Q_{22})$$

$$\begin{aligned} f(x, y) &\approx \frac{y_2 - y}{y_2 - y_1} f(x, y_1) + \frac{y - y_1}{y_2 - y_1} f(x, y_2) \\ &\approx \frac{y_2 - y}{y_2 - y_1} \left(\frac{x_2 - x}{x_2 - x_1} f(Q_{11}) + \frac{x - x_1}{x_2 - x_1} f(Q_{21}) \right) + \frac{y - y_1}{y_2 - y_1} \left(\frac{x_2 - x}{x_2 - x_1} f(Q_{12}) + \frac{x - x_1}{x_2 - x_1} f(Q_{22}) \right) \\ &= \frac{1}{(x_2 - x_1)(y_2 - y_1)} \left(f(Q_{11})(x_2 - x)(y_2 - y) + f(Q_{21})(x - x_1)(y_2 - y) + f(Q_{12})(x_2 - x)(y - y_1) + f(Q_{22})(x - x_1)(y - y_1) \right) \end{aligned}$$

Tehát a *P* pont magasság koordinátáját úgy számoljuk, hogy a négyzetek felszínre eső merőleges vetületeivel vett téglalapok területének arányában súlyozzuk a megfelelő középpontokhoz tartozó magasság értékeket (lásd az 5. és *6. ábrán*). (forrás: https://en.wikipedia.org/wiki/Bilinear_interpolation)

C melléklet

C.1. A többváltozós menetidőbecslő modell tesztstatisztikái

A többváltozós modell tesztstatisztikái									
variable	**coeff**	**std. err.**	**t-value**	**p-value**	**variable**	**coeff**	**std. err.**	**t-value**	**p-value**
intercept	2,435E+09	1,753E+09	1,389	1,65	**d30_1**	1,013E+13	1,295E+13	0,782	0,434
β_1	-3,533e	3,901E+01	-9,057	< 2e-16	**d30_2**	7,437E+12	9,782E+12	0,760	0,447
β_2	2,987E-01	5,376E-02	5,557	2,74E-08	**d30_3**	-9,595E+13	1,214E+14	-0,791	0,429
β_3	5,979E-02	6,094E-03	9,811	< 2e-16	**d30_4**	NA	NA	NA	NA
β_4	4,291E-03	3,796E-04	11,306	< 2e-16	**d30_5**	NA	NA	NA	NA
β_5	1,388E-04	1,173E-05	11,830	< 2e-16	**d30_6**	-2,339E+15	2,967E+15	-0,788	0,430
β_6	1,990E-06	1,659E-07	11,998	< 2e-16	**d30_7**	NA	NA	NA	NA
β_7	1,027E-08	8,522E-10	12,045	< 2e-16	**S_1**	-2,945E+12	3,731E+12	-0,789	0,430
a30_1	1,032E+12	1,303E+11	7,915	2,48E-15	**S_2**	1,219E+12	1.568e+12	0,778	0,437
a30_2	2,151E+13	2,509E+12	8,575	< 2e-16	**S_3**	1,864E+12	2,509E+12	0,743	0,458
a30_3	1,154E+14	1,341E+13	8,606	< 2e-16	**S_4**	NA	NA	NA	NA
a30_4	NA	NA	NA	NA	**S_5**	-3.934e+13	6,880E+12	-0,588	0,556
a30_5	NA	NA	NA	NA	**S_6**	NA	NA	NA	NA
a30_6	-1,362E+16	1,531E+15	-8,894	< 2e-16	**S_7**	7.634e+14	4.374e+15	0,175	0,861
a30_7	-1,508E+17	1,598E+16	-9,438	< 2e-16	**p_1**	1,995E+12	4,473E+12	0,446	0,655
p_1	-6,589E+10	7,043E+09	-9,356	< 2e-16	**p_2**	4,479E+13	1,023E+14	0,438	0,661
p_2	NA	NA	NA	NA	**p_3**	3,787E+14	8,776E+14	0,432	0,666
p_3	-5.390e+11	5,445E+10	-9,900	< 2e-16	**p_4**	NA	NA	NA	NA
p_4	8.325e+12	3,346E+11	9,974	< 2e-16	**p_5**	-1,070E+16	2,535E+16	-0,422	0,672
p_5	-7.720e+13	7,818E+12	-9,874	< 2e-16	**p_6**	NA	NA	NA	NA
p_6	3.240e+14	3,329E+13	9,734	< 2e-16	**p_7**	NA	NA	NA	NA
p_7	NA	NA	NA	NA					

RSE	**0,468**
adj. Rsq	**0,0892**
Degr. Freedom	**794653**
F stat	**4578**
p-value	**< 2.2e-16**

C.2. Az illesztett v(m) sebesség-meredekség görbe tesztstatisztikái

A *p(m)* polinom együtthatói és az illesztés tesztstatisztikái

	coeff	std. err.	t value		p(m)	exp1	exp2
0	1,422E+00	4,011E-03	354,590	RSE	0.01648	0.04387	0.03422
1	-1,708E-01	1,466E-01	-1,165	adj. R^2	0.9774	0.9359	0.9649
2	-6,566E+01	5,107E+00	-1,2858	Degr. Freedom	124	112	113
3	-8,371E+01	7,463E+01	-1,122	F stat	581.4	1652	3135
4	1,129E+04	1,689E+03	6,686	p-value	< 2.2e-16	< 2.2e-16	< 2.2e-16
5	1,156E+04	1,201E+04	0,962				
6	-1,075E+06	2,132E+05	-5,041				
7	-6,130E+05	7,470E+05	-0,821				
8	4,643E+07	1,131E+07	4,104				
9	1,112E+07	1,570E+07	0,708				
10	-7,398E+08	2,126E+08	-3,480				
exp1_coeff	2.3203	0.05709	40.65				
exp1_interc	0.4462	0.01617	27.60				
exp2_coeff	-2.4672	0.04407	-55.99				
exp2_interc	0.3769	0.01253	30.08				

C.3. A három sebességbecslő eljárás össszehasonlításának tesztstatisztikái

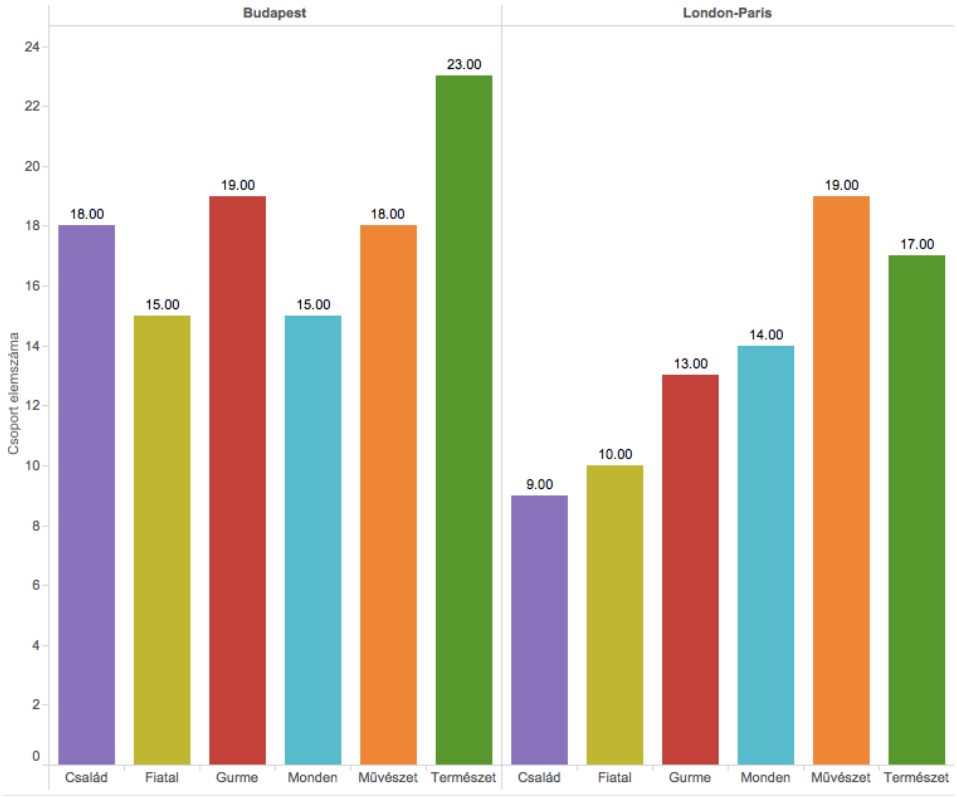
A kétmintás Welch-tesztek eredményei				
	mer.ill - atl.ill	**mer.ill - Tobler**	**atl.ill - Tobler**	**mer - Tobler.ill**
t-value	1,797	17,429	14,086	6,439
p-value	0,0740	< 2.2e-16	< 2.2e-16	1.02e-09
mer_mean	0,1158	0,1158	-	0,1158
atl_mean	0,1294	-	0,1294	-
Tobler_mean	-	0,1668	0,1668	0,1348
DF	194,88	135,78	130,79	183,72
95% conf int.	-0,0008	0,0493	0,0411	0,0164
95% conf int.	0,0161	0,0619	0,0546	0,0309
Ho	**rejected**	**rejected**	**rejected**	**rejected**

D melléklet

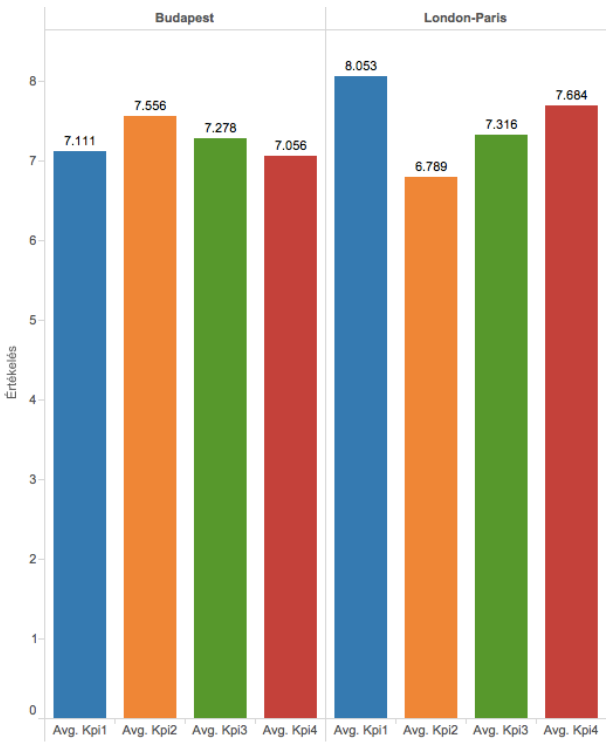
D.1. A 17 faktor korrelációs mátrixa

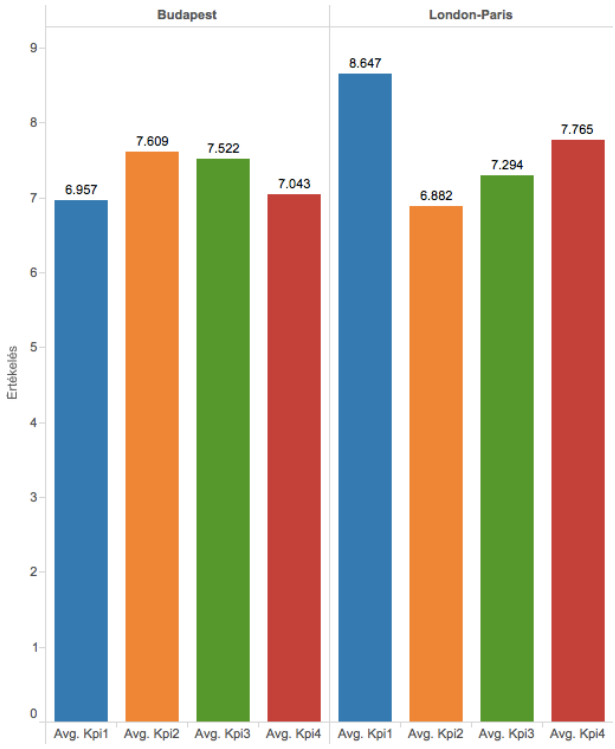
	museum/art	park/nature	architecture/facade	history/culture	shopping/fashion	vista point	top sight	music/bars/nightlife	market/local food	street/square	monument	bath/sport/recreation	theatre/entertainment/cinema	cafe/restaurant	church/religious place	university/science/technology	amusement/family/children
museum/art	1,00	0,14	0,35	0,59	-0,11	-0,08	-0,03	0,07	-0,14	0,16	0,37	-0,22	-0,13	-0,19	0,44	0,19	0,03
park/nature		1,00	0,10	0,05	0,12	0,34	0,15	0,02	-0,05	0,01	-0,11	-0,05	0,14	-0,11	-0,08	0,04	0,12
architecture/facade			1,00	0,56	0,01	0,15	0,20	0,16	0,03	0,28	0,54	-0,09	-0,09	-0,04	0,46	0,09	0,05
history/culture				1,00	0,08	0,00	0,16	0,04	-0,07	0,26	0,55	-0,11	0,00	0,05	0,40	0,21	0,07
shopping/fashion					1,00	0,10	0,40	0,42	0,16	0,31	0,14	0,23	0,54	0,52	0,05	0,16	0,25
vista point						1,00	0,22	0,13	-0,22	0,18	0,14	-0,03	0,06	0,16	0,12	0,13	-0,02
top sight							1,00	0,26	0,03	0,27	0,42	0,44	0,34	0,28	0,07	0,23	0,13
music/bars/nightlife								1,00	0,08	0,37	0,15	0,38	0,39	0,28	0,00	0,25	0,09
market/local food									1,00	0,39	-0,13	0,23	0,12	0,25	0,02	0,05	0,14
street/square										1,00	0,35	0,24	0,33	0,33	0,23	0,37	-0,02
monument											1,00	-0,02	0,10	0,05	0,50	0,30	0,13
bath/sport/recreation												1,00	0,51	0,38	-0,19	0,30	0,22
theatre/entertainment/cinema													1,00	0,55	-0,06	0,48	0,40
cafe/restaurant														1,00	0,08	0,23	0,16
church/religious place															1,00	0,23	0,21
university/science/technology																1,00	0,42
amusement/family/children																	1,00

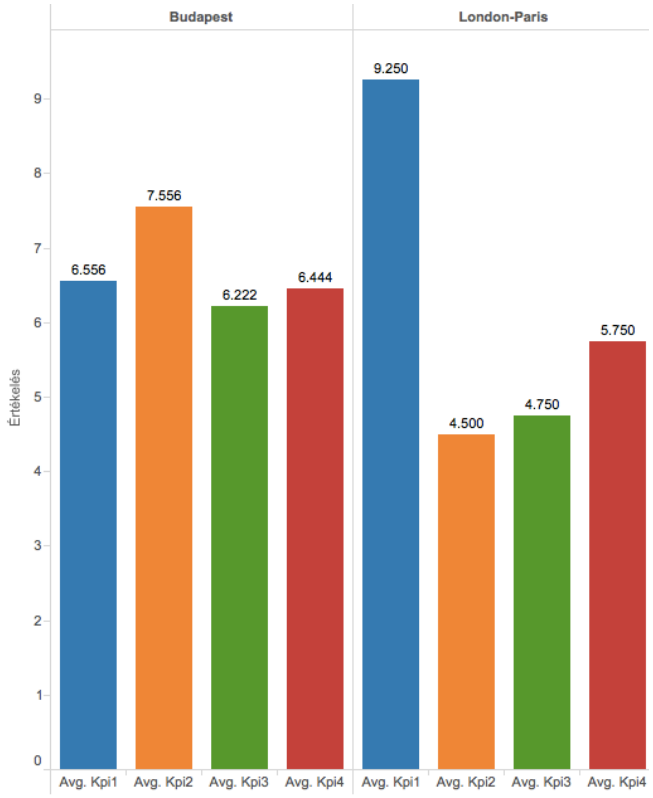
D.2. Az egyes turista típusok összsokaságon belüli megoszlása

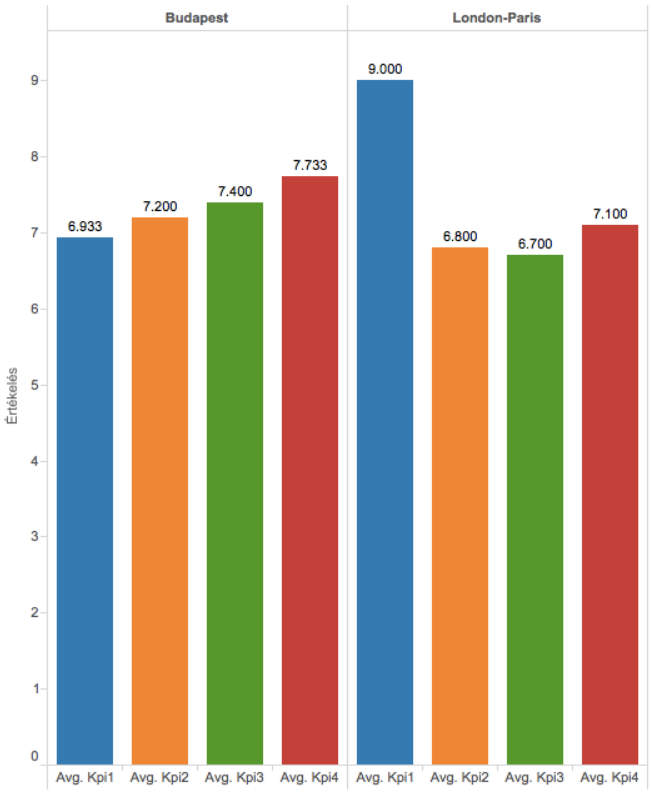


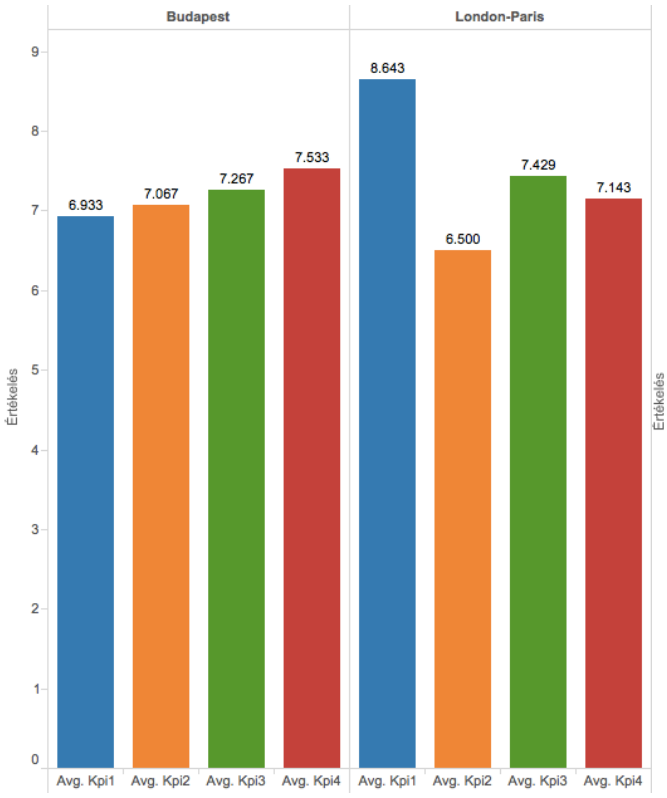
D.3. Az egyes turista típusok által adott értékelések (sorrendben: Művészet kedvelő, természet kedvelő, családos, fiatal, mondén, gurmé)

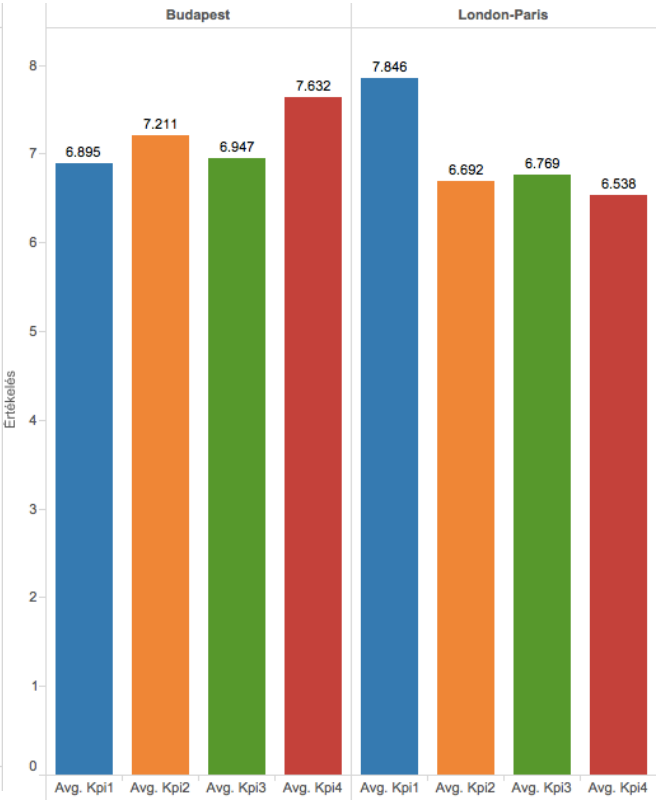




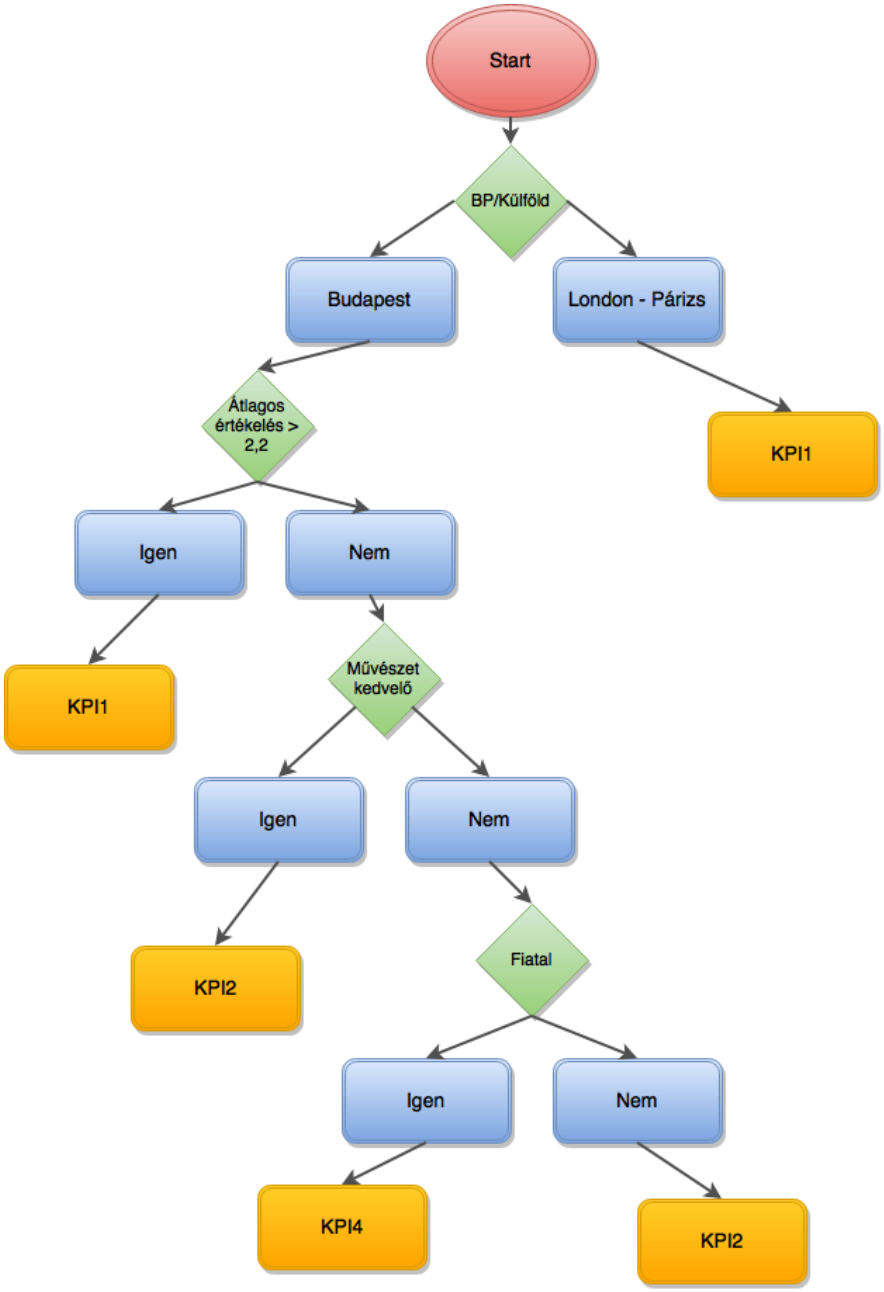

Budapest
London-Paris
Értékelés
0
1
2
3
4
5
6
7
8
9
6.556
7.556
6.222
6.444
9.250
4.500
4.750
5.750
Avg. Kpi1
Avg. Kpi2
Avg. Kpi3
Avg. Kpi4


Budapest
London-Paris
Értékelés
0
1
2
3
4
5
6
7
8
9
6.933
7.200
7.400
7.733
9.000
6.800
6.700
7.100
Avg. Kpi1
Avg. Kpi2
Avg. Kpi3
Avg. Kpi4


Budapest
London-Paris
Értékelés
0
1
2
3
4
5
6
7
8
9
6.933
7.067
7.267
7.533
8.643
6.500
7.429
7.143
Avg. Kpi1
Avg. Kpi2
Avg. Kpi3
Avg. Kpi4

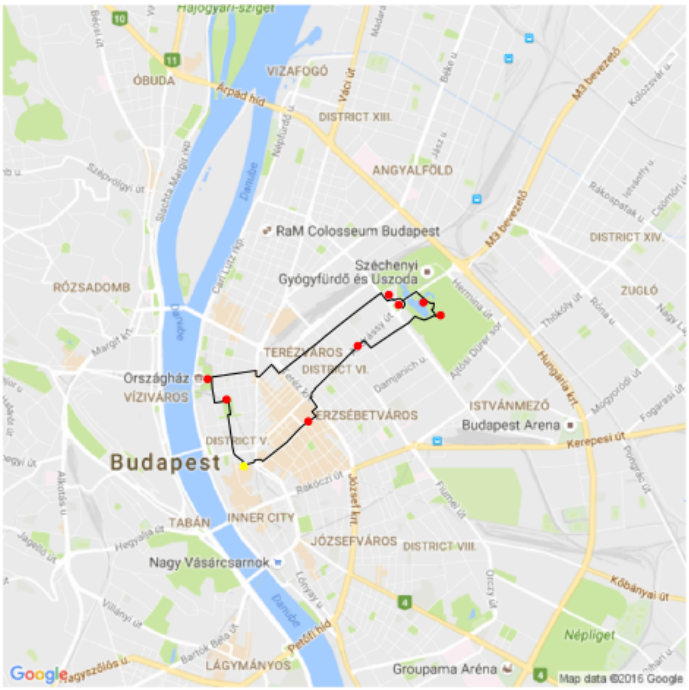

Budapest
London-Paris
Értékelés
0
1
2
3
4
5
6
7
8
6.895
7.211
6.947
7.632
7.846
6.692
6.769
6.538
Avg. Kpi1
Avg. Kpi2
Avg. Kpi3
Avg. Kpi4

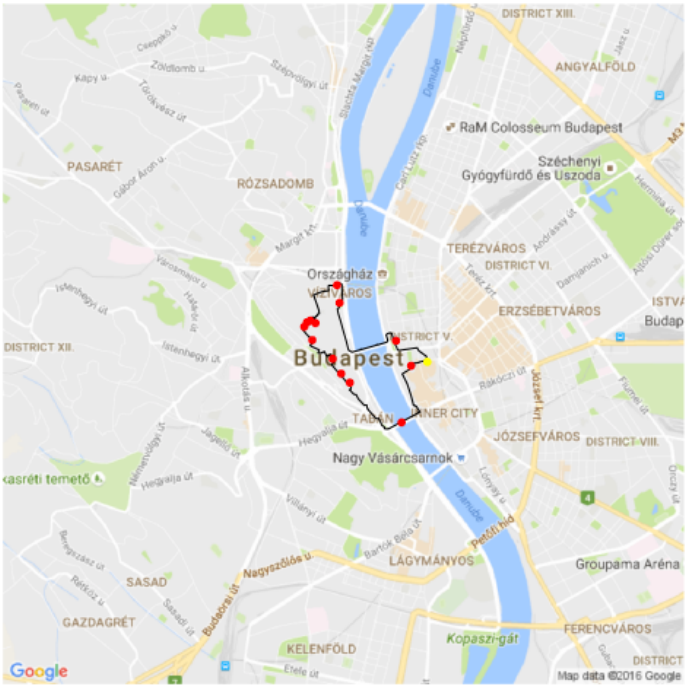
D.4. Az ajánlásra a turista típusok és értékelések alapján adott döntési mechanizmus ábrája

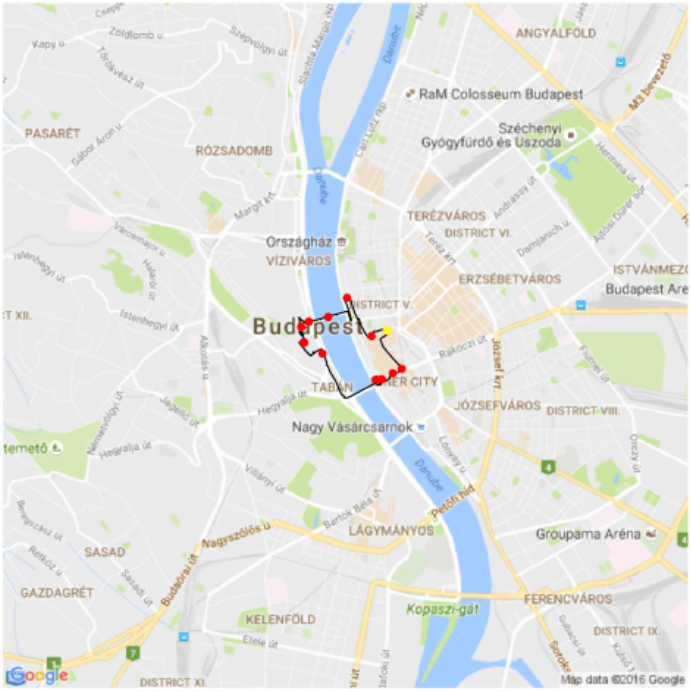


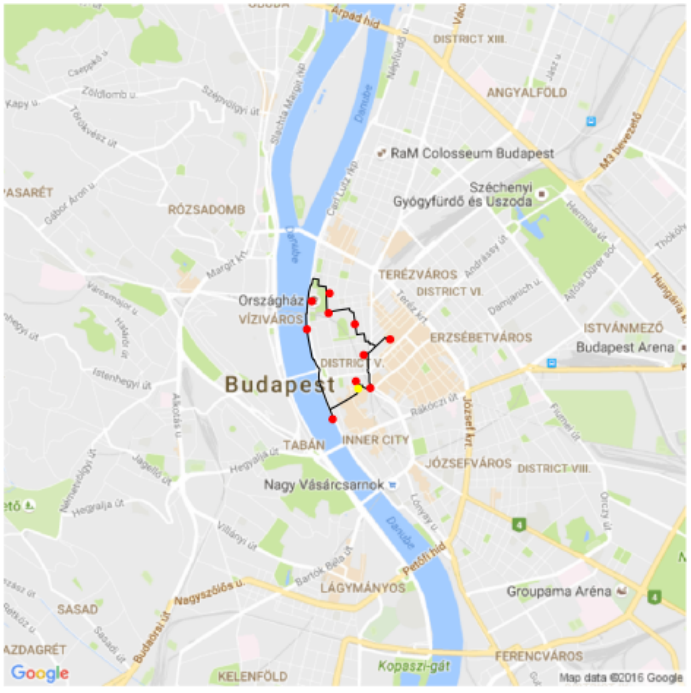
E melléklet

E.1. Az α = 0,75; *a* = -1 és *β* = 2 eset útvonalterve

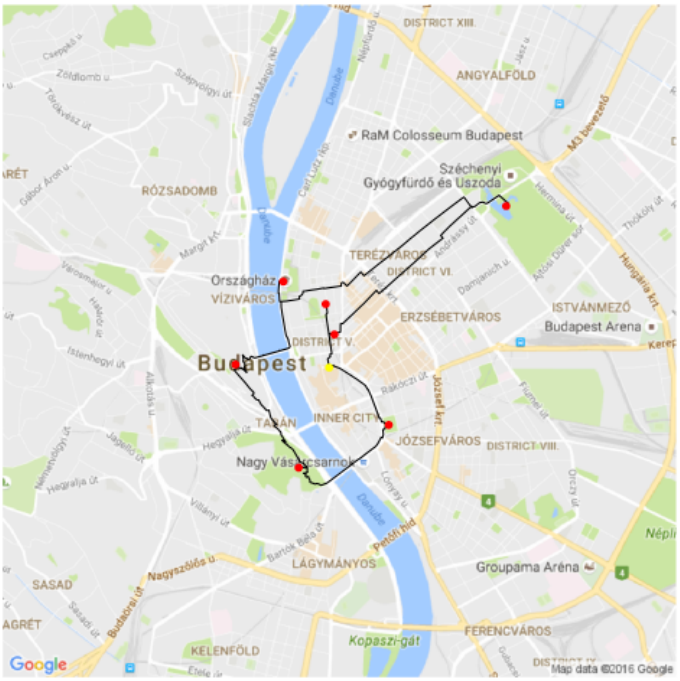


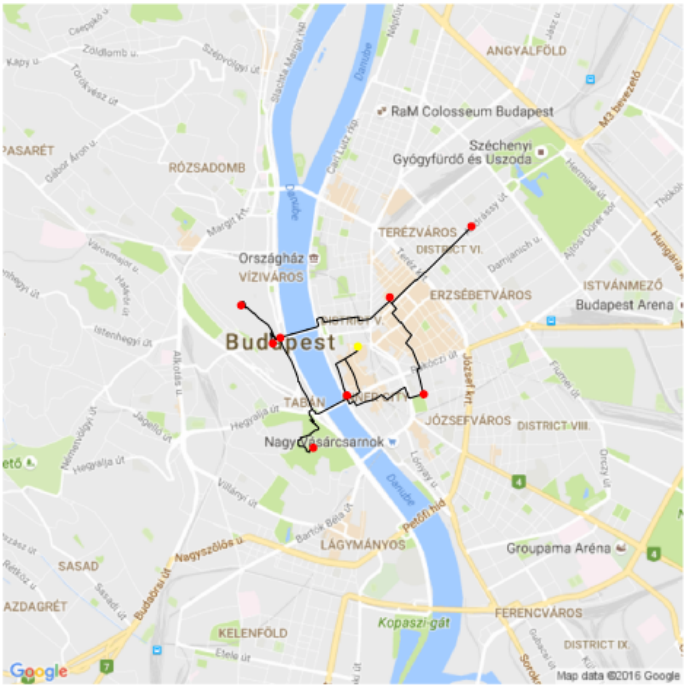


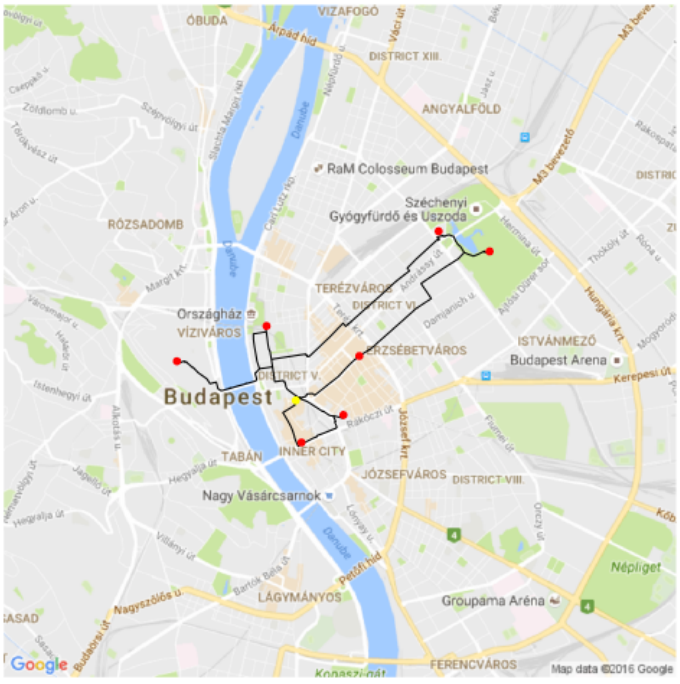




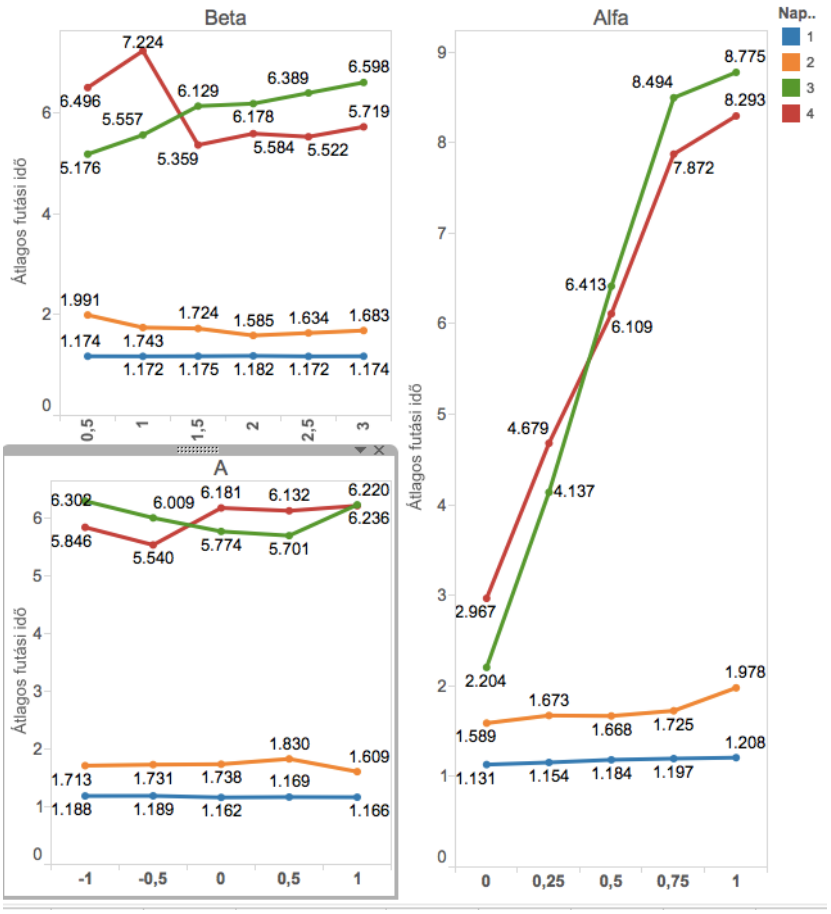
Az $\alpha = 0$ és $a = 0$ eset útvonalterve

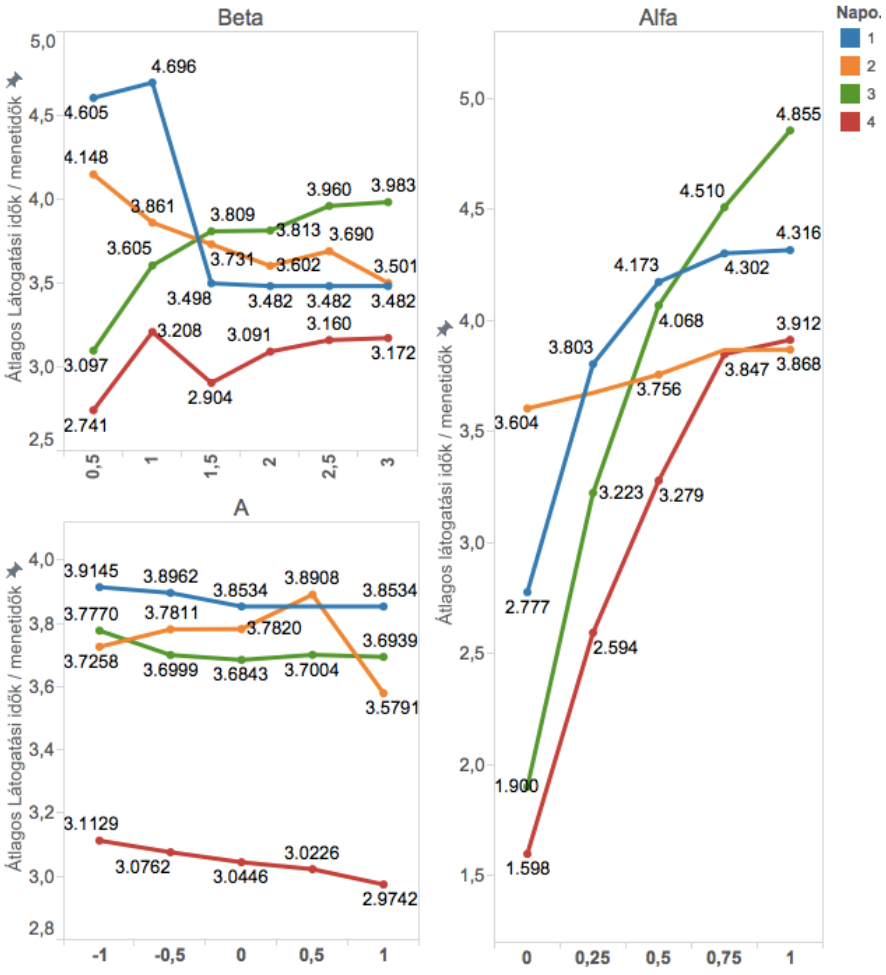




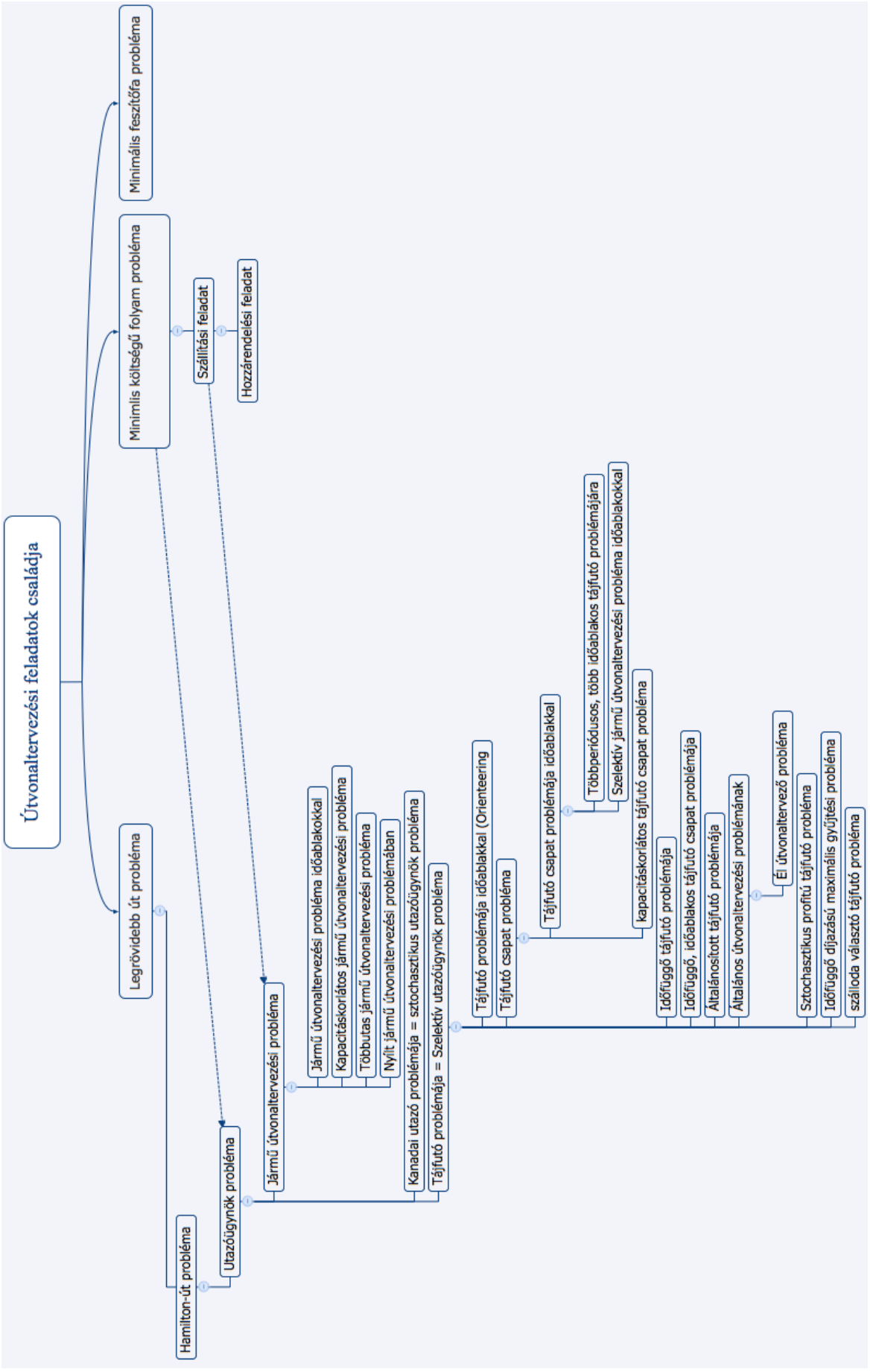


E.2. Az útvonaltervező algoritmus eredményei a paraméterek függvényében




Beta
Alfa
A
Napo.
1
2
3
4
Átlagos Látogatási idők / menetidők
Átlagos látogatási idők / menetidők
Átlagos Látogatási idők / menetidők
5,0
4,5
4,0
3,5
3,0
2,5
2,0
1,5
3,8
3,6
3,4
3,2
2,8
0,5
1
1,5
2
2,5
3
-1
-0,5
0
0,5
1
0
0,25
0,5
0,75
1
4.696
4.605
4.148
3.861
3.809
3.960
3.983
3.813
3.690
3.605
3.731
3.602
3.501
3.498
3.482
3.482
3.482
3.097
3.208
3.091
3.160
3.172
2.904
2.741
3.9145
3.8962
3.8534
3.8908
3.8534
3.7770
3.7811
3.7820
3.6939
3.7258
3.6999
3.6843
3.7004
3.5791
3.1129
3.0762
3.0446
3.0226
2.9742
4.855
4.510
4.316
4.173
4.302
4.068
3.912
3.803
3.847
3.868
3.756
3.604
3.223
3.279
2.777
2.594
1.900
1.598

E.3. Az útvonaltervező feladatok családja



E.4. Az útvonaltervező eljárásokra adott értékelések t-tesztje

	teszt értékek
t-érték	7,1172
p-érték	1.745e-11
saját eljárás átlag	8,1810
pontösszeg max átlag	6,3879
DF	208,44
95% konf int.	1,2964
95% konf int.	2,2898
Ho	**rejected**

Köszönetnyilvánítás

Szeretném hálám kifejezeni mindazon Kollégáknak, Barátoknak és nem utolsó sorban Családtagoknak, akik tanácsaikkal, munkájukkal és türelmükkel segítettek a dolgozat megírása során.

Ábrák jegyzéke

Táblázatok jegyzéke

Irodalomjegyzék

[1] K. Menger (1928): *Ein Theorem über die Bogenlange*, Anzeiger — Akademie der Wissenschaften in Wien — Mathematisch-naturwissenschaftliche, Klasse 65, pp. 264–266.

[2] G. Birkhoff (1946): *Tres observaciones sobre el algebra lineal*, Revista Facultad de Ciencias Exactas, Puras y Aplicadas Universidad Nacional de Tucuman, Serie A (Matematicas y Fisica Teorica), Vol. 5, pp. 147–151.

[3] G. Dantzig, R. Fulkerson, S. Johnson (1954): *Solution of a Large Scale Traveling Salesman Problem*, Journal of the Operations Research Society of America, Vol. 2, pp. 393–410. doi: 10.1287/opre.2.4.393

[4] D. Kőnig (1931): *Graphok és mátrixok*, Matematikai és Fizikai Lapok Vol. 38, pp. 116–119.

[5] T. Gallai (1958): *Maximum-minimum Satze uber Graphen*, Acta Mathematica Academiae Scientiarum Hungaricae, Vol. 9, pp. 395–434.

[6] J. Egerváry (1931): *Mátrixok kombinatorius tulajdonságairól*, Matematikai és Fizikai Lapok, Vol. 38, pp. 16–28.

[7] E. Egerváry (1958): *Bemerkungen zum Transportproblem*, MTW Mitteilungen, Vol. 5, pp. 278–284.

[8] H.W. Kuhn (1955): *The Hungarian method for the assignment problem*, Naval Research Logistics Quarterly, Vol. 2, pp. 83–97. doi: 10.1007/978-3-540-68279-0_2

[9] A. Schrijver (2005): *On the history of combinatorial optimization (till 1960),* Handbook of Discrete Optimization (K. Aardal, G.L. Nemhauser, R. Weismantel, eds.), Elsevier, Amsterdam, pp. 1–68. doi:10.1016/S0927-0507(05)12001-5

[10] G.B. Dantzig - J.H. Ramser (1959): *The Truck Dispatching Problem*, Management Science, Vol. 6, pp. 80–91. doi: 10.1287/mnsc.6.1.80

[11] Y. Chang - L. Chen (2007): *Solving the Vehicle Routing Problem with Time Windows via a Genetic Algorithm*, Discrete and Continuous Dynamical Systems Supplement, pp. 240-249. doi: 10.1016/j.eswa.2008.09.001

[12] B. Bullnheimer - R. F. Hartl - C. Strauss (1999): *Applying the Ant System to the Vehicle Routing Problem*, S. Voss - I.H. Osman - C. Roucairol (eds.): Meta-Heuristics: Advances and Trends in Local Search Paradigms for Optimization, Kluwer Academic Publishers Norwell, pp. 285-296. doi: 10.1007/978-1-4615-5775-3_20

[13] Z.J. Czech - P. Czarnas (2003): *Parallel simulated annealing for the vehicle routing problem with time windows*, in Proceesings of the 5th International Conference, pp. 233-240. doi: 10.1109/EMPDP.2002.994313

[14] O. Bräysy - M. Gendreau (2003): *Tabu Search Heuristics for the Vehicle Routing Problem with Time Windows*, Sociedad de Estadística e Investigación Operativa TOP, Vol. 10, No. 2, pp. 211-237. doi: 10.1007/BF02579017

[15] T. Bektaş - G. Erdoğan - S. Røpke (2011): *Formulations and Branch-and-Cut Algorithms for the Generalized Vehicle Routing Problem*, Journal of Transportation Science, Vol. 45, No. 3, pp. 299 - 316. doi: 10.1287/trsc.1100.0352

[16] A. Pessoa - M. Poggi de Aragao - E. Uchoa (2008): *Robust Branch-Cut-and-Price Algorithms for Vehicle Routing Problems*, The Vehicle Routing Problem: Latest Advances and New Challenges, series of the Operations Research/Computer Science Interfaces, Vol. 43, pp. 297-325. doi: 10.1007/978-0-387-77778-8_14

[17] L.M. Rousseau - M. Gendreau - G. Pesant - F. Focacci (2004): *Solving VRPTWs with Constraint Programming Based Column Generation*, Kluwer Academic Publishers, Annals of Operations Research, Vol. 130, pp. 199–216. doi: 10.1023/B:ANOR.0000032576.73681.29

[18] D. Feillet - P. Dejax - M. Gandreau (2004): *Traveling Salesman Problems with Profits*, Journal of Transportation Science, Vol. 39, No. 2, pp. 188-205. DOI: 10.1287/trsc.1030.0079

[19] B. Golden - L. Levy - R. Vohra (1984): *The Team Orienteering Problem*, Naval Research Logistics Quarterly, Vol. 34, pp. 307–318. DOI: 10.1002/1520-6750(198706)34:3<307::AID-NAV3220340302>3.0.CO;2-D

[20] S.E. Butt - T.M. Cavalier (1994): *A heuristic for the multiple tour maximum collection problem*, Computers and Operations research, Vol. 21, pp. 101-111. *A heuristic for the multiple tour maximum collection problem*

[21] W. Souffriau - P. Vansteenwegen - G. Vanden Berghe - D.D. Van Oudheusden (2013): *The Multiconstraint Team Orienteering Problem with Multiple Time Windows*, Jounal of Transportation Science, Vol. 47, No. 1, pp. 53-63. doi: 10.1287/trsc.1110.0377

[22] P. Vansteenwegen - W. Souffriau, - D. Van Oudheusden (2011): *The orienteering problem : a survey*, European Journal of Operational Research, Vol. 209, No. 1, pp. 1–10. doi:10.1016/j.ejor.2010.03.045

[23] A. Garcia - O. Arbelaitz - M. T. Linaza - P. Vansteenwegen - W. Souffriau (2010): *Personalized tourist route generation*, in Proceedings of the 10th International Conference on Current Trends in Web Engineering, pp. 486–497. doi: 10.1007/978-3-642-16985-4_47

[24] K. Sylejmani - A. Dika (2011): *Solving touristic trip planning problem by using taboo search approach*, International Journal of Computer Science Issues, Vol. 8, Issue 5, No 3, pp. 139-149. doi: 10.1109/HIS.2012.6421351

[25] W. Naismith: *Notes and queries*, [1892] Scottish Mountaineering Club Journal, Vol. 2, p. 133.

[26] E. Langmuir (1995): *Mountaincraft and Leadership*, 3rd ed. Sportscotland.

[27] R. Aitken (1977): *Wilderness Areas in Scotland*, unpublished Ph.D. Thesis. University of Aberdeen. Aberdeen.

[28] W. Tobler (1993): *Three presentations on geographical analysis and modeling: Non-isotropic geographic modeling speculations on the geometry of geography global spatial analysis*, National Center for Geographic Information and Analysis Technical Report, Vol. 93, No. 1, pp. 1–24.

[29] A. Kay (2012): *Route choice in hilly terrain*, Geographical Analysis, Vol. 44, No. 2, pp. 87–108. DOI: 10.1111/j.1538-4632.2012.00838.x

[30] A. Pitman - M. Zanker - J. Gamper - P. Andritsos (2012): *Individualized hiking time estimation*, in Proceedings of the 23rd International Workshop on Database and Expert Systems Applications, pp. 101-105. doi: 10.1109/DEXA.2012.51

[31] A. Pitman - J. Bernhart - C. Posch - M. Zambaldi - M. Zanker (2013): *Time-of-arrival estimation in mobile tour guides*, in Proceedings of the 20th Conference on Information and Communication Technologies in Tourism (ENTER), pp. 7-81. doi: 10.1007/978-3-642-36309-2_7

[32] F. Ricci (2011): *Mobile Recommender Systems,* Journal of Information Technology & Tourism, Vol. 12, No. 3, pp. 205-231. doi: 10.3727/109830511X12978702284390

[33] G. Tumas - F. Ricci (2009): *Personalized mobile city transport advisory system*, W. Höpken - U. Gretzel - R. Law (eds.): Information and Communication Technologies in Tourism, Springer Vienna, pp. 173–183. doi: 10.1007/978-3-211-93971-0_15

[34] A. Kay (2012): *Pace and Critical Gradient for Hill Runners: An Analysis of Race Records*, Journal of Quantitative Analysis in Sports, Vol. 8, No. 4. doi: 10.1515/1559-0410.1456

[35] D. Gavalas - C. Konstantopoulos - K. Mastakas - G. Pantziou (2014): *A survey on algorithmic approaches for solving tourist trip design problems*, Journal of Heuristics, Vol. 20, No. 3, pp. 291-328. doi: 10.1007/s10732-014-9242-5

[36] J. Letchner - J. Krumm - E. Horvitz (2006): *Trip router with individualized preferences (trip): incorporating personalization into route planning*, in Proceedings of the 18th Conference on Innovative Applications of Artificial Intelligence, Vol. 2, pp. 1795–1800. doi: 10.1.1.67.194

[37] F. V. Renatus (1767): *De Re Militari Book I: The Selection and Training of New Leviess*, english translation by John Clarke, p. 390.

[38] P. Scarf (1998): *An empirical basis for Naismith's rule*, Mathematics Today, Vol. 34, pp. 149-151. doi: 10.1080/02640410400023282

[39] J. H. Mathews - K. K. Fink (2004): *Numerical Methods Using Matlab*, 4th Edition, Prentice-Hall Inc., Upper Saddle River, New Jersey, USA, pp. 280-290. doi: 10.1002/0471705195

[40] Zs. Magyari-Sáska - S. Dombay (2012): *Determining Minimum Hiking Time using DEM*, Geographia Napocensis Anul, Vol. 6, Nr. 2, pp. 124-129.

[41] E. Imhof (1950): *Gelaende und Karte*, Rentsch, Zurich, pp. 217-220.

[42a] http://www.merriam-webster.com/dictionary/information

[42b] U. Hanani - B. Shapira - P. Shoval (2001): *Information filtering: Overview of issues, research and systems*, User Modeling and User-Adapted Interaction, Vol. 11, pp. 203-259. doi: 10.1023/A: 1011196000674

[43] P. Melville - V. Sindhwani (2011): *Recommender Systems*, in C. Sammut - G.I. Webb (Eds.): Encyclopedia of Machine Learning, Springer. pp. 829-838. doi: 10.1007/978-0-387-30164-8_705

[44] http://en.citizendium.org/wiki/Recommendation_system

[45] E. Rich (1979): *User modeling via stereotypes*, Cognitive Science, Vol. 3, No. 4, pp. 329–354. doi: DOI: 10.1207/s15516709cog0304_3

[46] R. E. Nisbett - T. D. Wilson (1977): *Telling more than we can know: Verbal reports on mental processes,* Psychological Review, Vol. 84, No. 3. pp. 231-259. doi: 10.1037/0033-295X.84.3.231

[47] http://www.walkbase.com/blog/nordics-largest-shopping-centre-wi-fi-analytics-will-drive-our-marketing-decisions

[48] D. Goldberg - B. Oki - D. Nichols - D. B. Terry (1992): *Using Collaborative Filtering to Weave an Information Tapestry,* Communications of the ACM, December, Vol. 35, No. 12, pp. 61-70. doi: 10.1145/138859.138867

[49] M. Sanderson - W. B. Croft (2012): *The History of Information Retrieval Research*, Proceedings of the IEEE, Vol. 100, pp. 1444–1451. doi: 10.1109/JPROC.2012.2189916

[50] G. Salton - A. Wong - C. S. Yang (1975): *A vector space model for automatic indexing*, Communications of the ACM, Vol.18, No.11, pp. 613-620. doi: 10.1145/361219.361220

[51] M.H. Ferrara - M. P. LaMeau (2012): *Pandora Radio/Music Genome Project. Innovation Masters: History's Best Examples of Business Transformation*. Detroit. pp. 267-270. Gale Virtual Reference Library. doi: 10.5860/CHOICE.50-2756

[52] M. Balabanovic´ - Y Shoham (1997): *Fab: Content-based, Collaborative Recommendation*, Communications of the ACM, Vol.40, No.3, pp.66-72. doi: 10.1145/245108.245124

[53] http://www.gravityrd.com

[54] J. B. Schafer - J. A. Konstan - J. Riedl (2001): *E-Commerce recommendation applications,* Data Mining and Knowledge Discovery, Vol. 5, No. 1, pp. 115–153. doi: 10.1007/978-1-4615-1627-9_6

[55] http://www.nytimes.com/2012/02/19/magazine/shopping-habits.html?_r=0

[56] Y. Rong - X. Wen - H. Cheng (2014): *A Monte Carlo Algorithm for Cold Start Recommendation*, WWW'14 Proceedings of the 23rd international conference on World wide web, pp. 327-336. doi: 10.1145/2566486.2567978

[57] E.B. Santos Jr. - M.G. Manzato - R. Goularte (2014): *Evaluating the impact of demographic data on a hybrid recommender model*, IADIS International Journal on WWW/Internet, Vol. 2, No.2, pp. 149-167.

[58] M. Kendall (1938): *A New Measure of Rank Correlation*, Biometrika, Vol. 30, pp. 81-89. doi: 10.1093/biomet/30.1-2.81

[59] A. M. Rashid - S. K. Lam - A. LaPitz - G. Karypis - J. Riedl (2008): *Towards a Scalable kNN CF Algorithm: Exploring Effective Applications of Clustering*, Advances in Web Mining and Web Usage Analysis, Lecture Notes in Computer Science, Vol. 4811, pp. 147-166. doi: 10.1.1.144.5863

[60] J. Breese - D. Heckerman - C. Kadie (1998): *Empirical analysis of predictive algorithms for collaborative filtering,* Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence (UAI '98), pp. 43-52. doi: 10.1.1.201.9694

[61] M. S. Charikar (2002): S*imilarity Estimation Techniques from Rounding Algorithms*, Proceedings of the 34th Annual ACM Symposium on Theory of Computing, pp. 380–388. doi: 10.1145/509907.509965

[62] P. Jaccard (1912): *The distribution of the flora in the alpine zone*, New Phytologist, Vol. 11, pp. 37-50. doi: 10.1111/j.1469-8137.1912.tb05611.x

[63] J. Herlocker - J. A. Konstan - J. Riedl (2002): *An empirical analysis of design choices in neighborhood-based collaborative filtering algorithms*, Information Retrieval, Vol. 5, No. 4, pp. 287–310. doi: 10.1023/A:1020443909834

[64] U. Shardanand - P. Maes (1995): *Social information filtering: Algorithms for automating "word of mouth"*, in ACM CHI '95, pp. 210–217. Press/Addison-Wesley Publishing Co. doi: 10.1145/223904.223931

[65] B.M. Sarwar - G.Karypis - J.A.Konstan - J.Riedl (2000): *Analysis of recommendation algorithms for E-commerce*, in Proceedings of the ACM E-Commerce, pp. 158–167. doi=10.1.1.38.5552

[66] E. Deza - M. M. Deza (2006): *Dictionary of Distances*, Elsevier, p. 69. doi: 10.1016/j.ejc. 2009.03. 020 .

[67] T. Zhang - R. Ramakrishnan - M. Livny (1996): *BIRCH: an efficient data clustering method for very large databases*, In SIGMOD'96, Montreal, Canada, pp. 103-114. doi:10.1.1.17.2504

[68] J. B. MacQueen (1967): *Some methods for classification and analysis of multivariate observations*, in Proceedings of the 5th Symposium on Math, Statistics, and Probability, pp. 281–297. doi:10.1.1.308.8619

[69] S.H.S. Chee - J. Han - K. Wang (2001): *RecTree: An Efficient Collaborative Filtering Method*, in Proceedings of the Third International Conference on Data Warehousing and Knowledge Discovery, pp. 141-151. doi: 10.1007/3-540-44801-2_15

[70] M. Ester - H.P. Kriegel - J. Sander - X. Xu (1996): *A density-based algorithm for discovering clusters in large spatial databases with noise*, in Proceedings of the International Conference on Knowledge Discovery and Data Mining (KDD '96), pp. pp. 226-231.

[71] L. Si - R. Jin (2003): *Flexible mixture model for collaborative filtering*, in Proceedings of the 20th International Conference on Machine Learning, Vol. 2, pp. 704–711. doi: 10.1145/1031171.1031201

[72] J. Canny (2002): *Collaborative filtering with privacy via factor analysis*, in Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 238–245. doi: 10.1145/564376.564419

[73] S. Vucetic - Z. Obradovic (2005): *Collaborative filtering using a regression-based approach*, Knowledge and Information Systems, Vol. 7, No. 1, pp. 1–22. doi: 10.1007/s10115-003-0123-8

[74] D. Lemire - A. Maclachlan (2005): *Slope one predictors for online rating-based collaborative filtering*, in Proceedings of the SIAM Data Mining Conference (SDM '05), pp. 471-475. DOI: http://dx.doi.org/10.1137/1.9781611972757.43

[75] J. Pearl (1988): *Probabilistic Reasoning in Intelligent Systems: Net- works of Plausible Inference*, Morgan Kaufmann, San Francisco, Calif, USA

[76] K. Miyahara - M. J. Pazzani (2002): *Improvement of collaborative filtering with the simple Bayesian classifier*, Information Processing Society of Japan, Vol. 43, No. 11, pp. 3429-3437.

[77] R. Greinemr - X. Su - B. Shen - W. Zhou (2005): *Structural extension to logistic regression: discriminative parameter learning of belief net classifiers*, Machine Learning, Vol. 59, No. 3, pp. 297–322. doi: 10.1007/s10994-005-0469-0

[78] T. Hofmann - J. Puziha (1999): *Latent class models for collaborative filtering*, in Proceedings of the 16th International Joint Conference on Artificial Intelligence, pp. 688–693.

[79] P. Symeonidis - P. Symeonidis - R. Nanopoulos - A. Papadopoulos - Y. Manolopoulos (2006): *Scalable Collaborative Filtering based on Latent Semantic Indexing*, in Proceedings of the 21st AAAI Workshop on Intelligent Techniques for Web Personalization (ITWP), Boston, MA, pp. 1–9.

[80] A.N. Burnetas - M. N. Katehakis (1995): *Optimal Adaptive Policies for Markov Decision Processes,* Mathematics of Operations Research, Vol. 22, No. 1, pp. 222-255. doi: 10.1287/moor. 22.1.222

[81] G. Shani - D. Heckerman - R. I. Brafman (2005): *An MDP-based recommender system*, Journal of Machine Learning Research, Vol. 6, pp. 1265–1295. doi:10.1.1.8.567

[82] D. Billsus and M. J. Pazzani (1998): *Learning collaborative information filters*, in Proceedings of the Fifteenth International Conference on Machine Learning, pp. 46-54. doi:10.1.1.40.4781

[83] G.H. Golub - W. Kahan (1965): *Calculating the singular values and pseudo-inverse of a matrix*, Journal of the Society for Industrial and Applied Mathematics: Series B, Numerical Analysis, Vol. 2, No. 2, pp. 205-224. DOI:10.1137/0702016

[84] B. M. Sarwar - G. Karypis - J. A. Konstan - J. T. Riedl (2000): *Application of dimensionality reduction in recommender system — a case study*, in WebKDD Workshop at the ACM SIGKKD

[85] M. Kurucz - A. A. Benczúr - K. Csalogány (2007): *Methods for large scale SVD with missing values*, in Proceedings of the KDD Cup and Workshop at the 13th ACM SIGKDD Conference, pp. 31-38.

[86] B. M. Sarwar - G. Karypis, J. Konstan - J. T. Riedl (2002): *Incremental SVD-based algorithms for highly scaleable recommender systems*, in Proceeding of the Conference on Computer and Information Technology 2002, pp. 125-130. doi:10.1.1.3.7894

[87] M. D. Ekstrand - J. T. Riedl - J. A. Konstan (2010): *Collaborative Filtering Recommender Systems*, Foundations and Trends in Human–Computer Interaction, Vol. 4, No. 2, pp. 81–173. doi: 10.1561/1100000009

[88] X. Su - T. M. Khoshgoftaar (2009): *A Survey of Collaborative Filtering Techniques,* Advances in Artificial Intelligence, Vol. 2009, Article ID 421425, 19 pages doi:10.1155/2009/421425

[89] B.M. Sarwar - G. Karypis - J.A. Konstan - J.Reidl (2001): *Item-based collaborative filtering recommendation algorithms*, Proceedings of the 10th international conference on World Wide Web, pp. 285–295. doi: 10.1145/371920.372071

[90] G. Linden - B. Smith - J. York (2003): *Amazon.com Recommendations: Item-to-Item Collaborative Filtering*, IEEE Internet Computing, Vol. 7, No. 1, pp. 76-80. doi: 10.1109/MIC. 2003.1167344

[91] G. Karypis (2001): *Evaluation of item-based top-N recommendation algorithms*, in Proceedings of the International Conference on Information and Knowledge Management (CIKM '01), pp. 247–254. doi: 10.1145/502585.502627

[92] F. Cacheda - V. Carneiro - D. Fernandez - V. Formoso (2011): *Comparison of collaborative filtering algorithms: Limitations of current techniques and proposals for scalable, high-*

performance recommender systems, ACM Transactions on the Web (TWEB), Vol. 5, No. 1, February 2011 Article No. 2. doi: 10.1145/1921591.1921593

[93] Takács G. - Pilászy I. - Németh B. - Tikk D. (2009): *Scalable Collaborative Filtering Approaches for Large Recommender Systems*, Journal of Machine Learning Research Vol. 10, pp. 623–656. doi:10.1007/978-3-642-38577-3_39

[94] S. A. Goldman - M. K. Warmuth (1995): *Learning binary relations using weighted majority voting*, Machine Learning, Vol. 20, No. 3, pp. 245–271. doi: 10.1145/168304.168396

[95] X. Su - T. M. Khoshgoftaar - X. Zhu - R. Greiner (2008): *Imputation-boosted collaborative filtering using machine learning classifiers*, in Proceedings of the 23rd Annual ACM Symposium on Applied Computing, pp. 949–950. doi: 10.1145/1363686.1363903

[96] H.C. Wu - R.W.P. Luk - K.F. Wong - K.L. Kwok (2008): *Interpreting TF-IDF term weights as making relevance decisions*, ACM Transactions on Information Systems, Vol. 26, No. 3, pp. 1-37. doi: 10.1145/1361684.1361686

[97] E-H. Han - G. Karypis (2000): *Centroid-Based Document Classification: Analysis & Experimental Results*, in Proceedings of the 4th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD), pp. 424 - 431. doi: 10.1007/3-540-45372-5_46

[98] M. Pazzani - D. Billsus (1997): *Learning and Revising User Profiles: The Identification of Interesting Web Sites*, Machine Learning, Vol. 27, pp. 313-331. doi: 10.1023/A:1007369909943

[99] N. Littlestone (1988): *Learning Quickly When Irrelevant Attributes Abound: A New Linear-threshold Algorithm,* Machine Learning, Vol. 2, No. 4, pp. 285-318. doi: 10.1023/A:1022869011914

[100] R.O. Duda - P.E. Hart - D.G. Stork (2001): *Pattern Classification*, John Wiley & Sons. doi: 10.1.1.320.4607

[101] A. van den Oord - S. Dieleman - B. Schrauwen (2013): *Deep content-based music recommendation*, in Proceedings of Neural Information Processing Systems Conference (NIPS 2013), Vol. 26, pp. 1-9.

[102] A. Gershman - A. Meisels - K.H. Lke - L. Rokach - A. Schclar - A. Sturm (2010): *A Decision Tree Based Recommender System*, in Proceedings of the 10th International Conference of Innovative Internet Community Services, pp. 170-179.

[103] P. Li - S. Yamada (2004): *A Movie Recommender System Based on Inductive Learning*, in Proceedings of the IEEE Conference on Cybernetics and Intelligent Systems, 2004. Vol. 1, pp. 318-323. doi: 10.1109/ICCIS.2004.1460433

[104] G. Adomavicius - A. Tuzhilin (2005): *Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions*, IEEE Transactions on Knowledge and Data Engineering, Vol. 17, No. 6, pp. 734-749. doi: 10.1109/TKDE.2005.99

[105] P. Resnick - N. Iacovou - M. Sushak - P. Bergstrom - J. Riedl (1994): *GroupLens: An open architechure for collaborative filtering of netnews*, In Proceedings of the ACM Conf. Computer Support Cooperative Work (CSC),pp. 175-186. doi: 10.1145/192844.192905

[106] L. Chen - M. deGemmis - A. Felfernig - P. Lops - F. Ricci - G. Semeraro (2013): *Human Decision Making and Recommender Systems*, ACM Transactions on Interactive Intelligent Systems, Vol. 3, No. 3, pp. Article 17. doi: 10.1145/2365952.2366040

[107] A. Felfernig - S. Haas - G. Ninaus - M. Schwarz - T. Ulz - M. Stettinger - K. Isak - M. Jeran - S. Reiterer (2014): *RecTurk: Constraint-based Recommendation based on Human Computation*, RecSys'2014 CrowdRec Workshop, Foster City, CA, USA, pp. 1-6.

[108] A. Felfernig - R. Burke (2008): *Constraint-based Recommender Systems: Technologies and Research Issues*, ACM International Conference on Electronic Commerce, pp. 17-26. doi: 10.1145/1409540.1409544

[109] L. Chen - P. Pu (2012): *Critiquing-based recommenders: survey and emerging trends*, User Modeling and User-Adapted Interaction Journal (UMUAI), Vol. 22, No. 1-2, pp. 125-150. doi: 10.1007/s11257-011-9108-6

[110] R. Burke (2000): *Knowledge-based Recommender Systems*, Encyclopedia of Library and Information Science, Vol. 69, No. 32, pp. 180-200. doi:10.1.1.21.6029&rank=1

[111] Zs. Sándor - Cs. Csiszár (2015): *Role of Integrated Parking Information System in Traffic Management*, Periodica Polytechnica - Civil Engineering, Vol. 59, No. 3, pp. 327-336. doi: 10.3311/PPci.7361

[112] R. Burke (2002): *Hybrid recommender systems: Survey and experiments*, User Modeling and User-Adapted Interaction, Vol. 12, No. 4, pp. 331–370. doi:10.1.1.88.8200&rank=1

[113] M. Claypool - A. Gokhale - T. Miranda - P. Murnikov - D. Netes - M. Sartin (1999): *Combining Content-Based and Collaborative Filters in an Online Newspaper*, SIGIR '99 Workshop on Recommender Systems: Algorithms and Evaluation. Berkeley, CA. Accessed at http://www.cs.umbc.edu/~ian/sigir99-rec/papers/claypool_m.ps.gz doi:10.1.1.145.9794

[114] B. Mobasher - X. Jin - Y. Zhou (2004): *Semantically Enhanced Collaborative Filtering on the Web*, In B. Berendt et al. (eds.): Web Mining: From Web to Semantic Web. LNAI Vol. 3209, Springer, pp. 57-76. doi: 10.1007/978-3-540-30123-3_4

[115] B. Smyth - P. Cotter (2000): *A Personalized TV Listings Service for the Digital TV Age*, Knowledge-Based Systems, Vol. 13, pp. 53-59. DOI: 10.1016/S0950-7051(00)00046-0

[116] D. Billsus - M. Pazzani (2000): *User Modeling for Adaptive News Access*, UMUAI Vol.10, No. 2-3, pp. 147-180. doi: 10.1023/A:1026501525781

[117] C. Basu - H. Hirsh - W. Cohen (1998): *Recommendation as Classification: Using Social and Content-Based Information in Recommendation*, in Processing of the 15th National Conference on Artificial Intelligence, pp. 714-720.

[118] P. Melville - R.J. Mooney - R. Nagarajan (2002): *Content-Boosted Collaborative Filtering for Improved Recommendations*, in Processing of the 18th National Conference on Artificial Intelligence, pp. 187-192. doi: 10.1109/CSNT.2012.218

[119] M.J. Pazzani (1999): *A Framework for Collaborative, Content-Based and Demographic Filtering*, Artificial Intelligence Review, Vol. 13, No. 5-6, pp. 393-408. doi: 10.1023/A:1006544522159

[120] R.J. Hyndman - A.B. Koehler (2006): *Another look at measures of forecast Accuracy*, International Journal of Forecasting, Vol. 22, No. 4, pp. 679-688. doi:10.1016/j.ijforecast.2006.03.001

[121] J.S. Armstrong - F. Collopy (1992): *Error Measures For Generalizing About Forecasting Methods: Empirical Comparisons*, International Journal of Forecasting, Vol. 8, No. 1, pp. 69-80. doi:10.1016/0169-2070(92)90008-W

[122] K. Steif (2004): *Creating a Model for Geodemographic representations of Housing Market Activity: A Research Note with possible Public Policy implications*, Middle States Geographer, Vol 37, pp. 116-121.

[123] F. Ricci - L. Rokach - B. Shapira - P.B. Kantor (2011): *Recommender Systems Handbook,* Springer. doi: 10.1007/978-0-387-85820-3

[124] D.M. Blei - A.Y. Ng - M.I. Jordan - J. Lafferty (2003): *Latent Dirichlet allocation*, Journal of Machine Learning Research, Vol. 3, No. 4–5, pp. 993–1022. doi:10.1162/jmlr.2003.3.4-5.993.

[125] H. Hotelling (1929): *Stability in Competition*, Economic Journal, Vol. 39, pp. 41-57. doi: 10.1007/978-1-4613-8905-7_4

[126] P.A. Chirita - W. Nejdl - C. Zamfir (2005): *Preventing shilling attacks in online recommender systems*, in Proceedings of the 7th annual ACM international workshop on Web Information and Data Management pp. 67-74. doi: 10.1145/1097047.1097061

[127] P. Adamopoulos - A. Tuzhilin (2013): *On Unexpectedness in Recommender Systems: Or How to Better Expect the Unexpected*, ACM Transactions on Intelligent Systems and Technology, Vol. 1, No. 1, Article 1. doi:10.1145/2559952

[128] M. Ghazanfar - A. Prugel-Bennett (2011): *Fulfilling the Needs of Gray-Sheep Users in Recommender Systems*, A Clustering Solution, in Proceedings of the International Conference on Information Systems and Computational Intelligence, Harbin, China, pp. 1-6.

[129] K. B. Duan - S. S. Keerthi (2005): *Which Is the Best Multiclass SVM Method? An Empirical Study*, Multiple Classifier Systems. Lecture Notes on Computer Science, Vol. *3541,* pp. 278–285. doi: 10.1007/11494683_28

[130] N. Friedman - D. Geiger - M. Goldszmidt (1997): *Bayesian network classifiers*, Machine Learning, Vol. 29, No. 2-3, pp. 131–163. DOI: 10.1002/9780470400531.eorms0099

[131] D. B. Rubin (1987): *Multiple Imputation for Nonresponse in Surveys*, John Wiley & Sons, New York, NY, USA

[132] F. Rodrigues - F. C. Pereira, - A. Alves - S. Jiang - J. Ferreira (2012): *Automatic Classification of Points-of-Interest for Land-use Analysis*, GEOProcessing 2012 : The Fourth International Conference on Advanced Geographic Information Systems, Applications, and Services, pp. 41-49.

[133] W. Cohen - P. Ravikumar - S. Fienberg (2003): *A comparison of string distance metrics for name-matching tasks,* Proceedings of the IJCAI-2003 Workshop on Information Integration on the Web (IIWeb-03), Acapulco, Mexico, pp. 73-78. doi=10.1.1.112.8784

[134] J. Payne - J. Bettman - E. Johnson (1993): *The adaptive decision maker*, Cambridge University Press. doi:10.1111/j.1467-9280.1993. tb00265.x.

[135] K. Cheverst - N. Davies - K. Mitchell - A. Friday - C. Efstratiou (2000): *Developing a context-aware electronic tourist guide: some issues and experiences*, in Proceedings of the SIGCHI conference on human factors in computing systems, pp. 17–24. doi: 10.1145/332040.332047

[136] W. Höpken - M. Fuchs - M. Zanker - T. Beer (2010): *Context-based adaptation of mobile applications in tourism*, Information Technology and Tourism, Vol. 12, No. 2, pp. 175–195. doi: 10.3727/109830510X12887971002783

[137] T. Horozov - N. Narasimhan - V. Vasudevan (2006): *Using location for personalized POI recommendations in mobile environments*, in Proceedings of the 2006 international symposium on applications and the internet (SAINT'06), pp. 124–129. doi:10.1109/SAINT.2006.55

[138] N.S. Savage - M. Baranski - N.E. Chavez - T. Höllerer (2011): *I'm feeling LoCo: a location based context aware recommendation system*, in Proceedings of the 8th international symposium on location-based services (LBS'11), pp. 37-54. doi:10.1007/978-3-642-24198-7_3

[139] A. García-Crespo - J. Chamizo - I. Rivera - M. Mencke - R. Colomo-Palacios - J.M. Gómez-Berbís (2009): *SPETA: social pervasive e-tourism advisor*, Telematics and Informatics, Vol. 26, No. 3, pp. 306–315. doi:10.1016/j.tele.2008.11.008

[140] D. Gavalas - M. Kenteris (2011): *A pervasive web-based recommendation system for mobile tourist guides*, Personal and Ubiquitous Computing, Vol. 15, No. 7, pp. 759–770. doi: 10.1007/ s00779-011-0389-x

[141] Y. Zheng - X. Xie (2011): *Learning travel recommendations from user-generated GPS traces*, ACM Transactions on Intelligent Systems and Technology, Vol. 2, No. 1, pp. 2–29. doi: 10.1145/1889681.1889683

[142] P. Vansteenwegen - W. Souffriau - G. Vanden Berghe - D.D. Van Oudheusden (2011): *The city trip planner: an expert system for tourists*, Expert Systems with Applications, Vol. 38. No. 6, pp. 6540–6546. doi:10.1016/j.eswa.2010.11.085

[143] B. Brown - M. Chalmers - M. Bell - I. MacColl - M. Hall - P. Rudman (2005): *Sharing the square: collaborative leisure in the city streets*, in Proceedings of the 9th European conference on computer-supported cooperative work (ECSCW'05), pp. 427–447. doi: 10.1007/1-4020-4023-7_22

[144] C.C. Yu - H.P. Chang (2009): *Personalized location-based recommendation services for tour planning in mobile tourism applications*, in Proceedings of the 10th International conference on e-commerce and web Technologies (EC-Web'09), pp. 38–49. doi:10.1007/978-3-642-03964-5_5

[145] T. Shiraishi - M. Nagata - N. Shibata - Y. Murata - K. Yasumoto - M. Ito (2005): *A Personal navigation system with a schedule planning facility based on multi-objective criteria*, in Proceedings of 2nd international conference on mobile computing and ubiquitous networking (ICMU'05); pp. 104–109. doi:10.1.1.59.5613

[146] A. Garcia - P. Vansteenwegen - O. Arbelaitz - W. Souffriau - M.T. Linaza (2013): *Integrating public transportation in personalized electronic tourist guides*, Computers and Operations Research, Vol. 40, No. 3, pp. 758–74. DOI: 10.1016/j.cor.2011.03.020

[147] D. Gavalas - C. Konstantopoulos - K. Mastakas - G. Pantziou (2014): *Mobile recommender systems in tourism*, Journal of Network and Computer Applications, Vol. 39, pp. 319–333. doi: 10.1016/j.jnca.2013.04.006

[148] *Interview with Stephen Kaufer*, BBC Radio 4: The Bottom Line, October 16, 2014.

[149] TripSay. ⟨http://www.tripsay.com/⟩

[150] DieToRecs. ⟨http://www.modul.ac.at/dietorecs⟩

[151] Heracles. ⟨http://www.isi.edu/integration/Heracles/⟩ letöltve: 2015.12.08.

[152] P. Scarf (2007): *Route choice in mountain navigation, Naismith's rule, and the equivalence of distance and climb*, Journal of Sports Science, Vol. 25, No. 6, pp. 719-726. doi: 10.1080/02640410600874906

[153] S. Fritz - S. Carver (2000): *Modelling remoteness in roadless areas using GIS,* In: B.O. Parks - K.M. Clarke - M.P. Crane, (editors): Problems, Prospects and Research Needs, in Proceedings of the 4th International Conference on Integrating GIS and Environmental Modelling (GIS/EM4), No. 157.

[154] M. Yang - F. van Coillie - M. Liu - R. de Wulf - L. Hens - X. Ou (2014): *A GIS Approach to Estimating Tourists' Off-road Use in a Mountainous Protected Area of Northwest Yunnan, China*, Mountain Research and Development, Vol. 34, No. 2, pp. 107-117. doi: 10.1659/MRD-JOURNAL-D-13-00041.1

[155] W.J. Li - X.D. Ge - C.Y. Liu (2005): *Hiking trails and tourism impact assessment in protected area: Jiuzhaigou Biosphere Reserve, China*, Environment Monitoring Assessment, Vol. 108, pp. 279–293. doi:10.1007/s10661-005-4327-0

[156] N.A. Lynn - R.D. Brown (2003): *Effects of recreational use impacts on hiking experiences in natural areas*, Landscape Urban Planning, Vol. 64, pp. 77–87. doi:10.1016/S0169-2046(02)00202-5

[157] I. Herzog (2013): *The potential and limits of optimal path analysis*, in A. Bevan, M. Lake (eds.): Computational Approaches to Archaeological Spaces, Walnut Creek, Left Coast Press, pp. 179-211.

[158] T.L. Kienlin - K. Cappenberg - M.M. Korczyńska (2013): *Überlegungen zu den spätbronze- und früheisenzeitlichen Landnutzungsstrategien im mittleren Dunajectal*, Kleinpolen. In: G. Kalaitzoglou- G. Lüdorf (Hrsg.), Petasos, Festschrift für Hans Lohmann. Mittelmeerstudien Vol. 2, pp. 319-332.

[159] I.I. Ullah - S.M. Bergin (2012): *Modeling the consequences of village site location*, in D.White - S. Surface-Evans (eds.): Least Cost Analysis of Social Landscapes, Archaeological Case Studies, Salt Lake City, University of Utah Press, pp. 155-173.

[160] P. Verhagen - K. Jeneson (2012): *A Roman puzzle. Trying to find the Via Belgica with GIS*, in A. Chrysanthi - P. Murrieta-Flores - C. Papadopoulos (eds.): *Thinking Beyond the Tool*, BAR International Series 2344, Oxford, Archaeopress, pp. 123-130.

[161] K. Rademaker - D.A. Reid - G.R.M. Bromley (2012): *Connecting the dots*, in D. White - S. Surface-Evans (eds.): Least Cost Analysis of Social Landscapes, Archaeological Case Studies, Salt Lake City, University of Utah Press, pp. 32-45.

[162] I. Herzog (2014): *A review of case studies in archaeological least-cost analysis*, Archeologia e Calcolatori, Vol. 25, pp. 223-239. doi:10.11141/ia.34.7

[163] N.J. Wood - M.C. Schmidtlein (2013): *Community variations in population exposure to near-field tsunami hazards as a function of pedestrian travel time to safety*, Natural Hazards, Vol. *65, No.* 3, pp. 1603-1628. doi: 10.1007/s11069-012-0434-8

[164] P.W. Gething - F.A. Johnson - F. Frempong-Ainguah - P. Nyarko - A. Baschieri - P. Aboagye - J. Falkingham - Z. Matthews - P.M. Atkinson (2012): *Geographical access to care at birth in Ghana: a barrier to safe motherhood*, BMC Public Health, Vol. 12, pp. 991-998. DOI: 10.1186/1471-2458-12-991

[165] A.M. Noor - A.A. Amin - P.W. Gething - P.M. Atkinson - S.I. Hay - R.W. Snow (2006): *Modelling distances travelled to government health services in Kenya*, Tropical Medicine & International Health, Vol. 11, No. 2, pp. 188-196. DOI: 10.1111/j.1365-3156.2005.01555.x

[166] G.C. Carr - H. Erzberger - F. Neuman (2000): *Fast-time study of airline-influenced arrival sequencing and scheduling*, Journal of Guidance, Control and Dynamics, Vol. 23, No. 3, pp. 526–531. doi: 10.2514/2.4559

[167] A. Rong - H. Hakonen - R. Lahdelma (2003): *Estimated Time of Arrival (ETA) Based Elevator Group Control Algorithm with More Accurate Estimation*, Turku Centre for Computer Science TUCS Technical Report No 584, ISBN 952-12-1289-6

[168] K.A.S. Al-Khateeb - J.A.Y. Johari - W.F. Al-Khateeb (2008): *Dynamic Traffic Light Sequence*, Journal of Computer Science, Vol. 4, No. 7, pp. 517–524. doi: 10.3844/jcssp.2008.517.524

[169] *P. Zhou - Y. Zheng - M. Li (2012): How Long to Wait?: Predicting Bus Arrival Time with Mobile Phone based Participatory Sensing,* in Proceedings of the 10th International Conference on Mobile Systems, Applications and Services, pp. 379-392. doi: 10.1109/TMC.2013.136

[170] C. Yu - J. Lee - M.J. Munro-Stasiuk (2003): *Extensions to least-cost path algorithms for roadway planning*, International Journal of Geographical Information Science, Vol. 17, No 4, pp. 361–376. DOI:10.1080/1365881031000072645

[171] S. Bagli - D. Geneletti - F. Orsi (2011): *Routing of power lines through least-cost path analysis and multi-criteria evaluation to minimise environmental impacts*, Environmental Impact Assessment Review, Vol. 31, pp. 234–239. DOI: 10.1016/j.eiar.2010.10.003

[172] Zs. Magyari-Sáska (2013): *Efficient Spatial Time-cost Analysis for Search of Lost Tourists*, Geographia Technica, No. 1, pp. 47-55.

[173] J. Hrncir - Q. Song - P. Zilecky - M. Nemet - M. Jakob (2014): *Bicycle route planning with route choice preferences*, in Prestigious Applications of Artificial Intelligence, pp. 1149-1154. DOI: 10.3233/978-1-61499-419-0-1149

[174] R.F. Pribul - J. Price (2005): *An Investigation into the Race Strategies of Elite and Non-Elite Orienteers*, Scientific Journal of Orienteering, Vol. 16, pp. 34-40.

[175] J.M. Norman (2004): *Running uphill: energy needs and Naismith's rule*, Journal of the Operational Research Society, Vol. 55, pp. 308–311. doi:10.1057/palgrave.jors.2601671

[176] A.E. Minetti (1995): *Optimum gradient of mountain paths*, Journal of Applied Physiology, Vol. 79, pp. 1698–1703.

[177] A.E. Minetti - C. Moia - G.S. Roi - D. Susta - G. Ferretti (2002): *Energy cost of walking and running at extreme uphill and downhill slopes*, Journal of Applied Physiology, Vol. 93, pp. 1039–1046. doi:10.1152/japplphysiol.01177.2001

[178] W.G. Rees (2004): *Least-cost paths in mountainous terrain*, Computers and Geosciences, Vol. 30, pp. 203–209. doi:10.1016/j.cageo.2003.11.001

[179] E.I. Verriest (2008): *A variant to Naismith's problem with application to path planning*, in Proceedings of the 17th World Congress, International Federation of Automatic Control (eds. M.J. Chung - P. Misra), pp. 7136–7141. doi:10.3182/20080706-5-KR-1001.01210

[180] S. Mills (1982): *Naismith's rule*, Climber and Rambler, Vol. 21, pp. 47.

[181] C. Hirt - M.S. Filmer - W.E. Featherstone (2010): *Comparison and validation of recent freely-available ASTER-GDEM ver1, SRTM ver4.1 and GEODATA DEM-9S ver3 digital elevation models over Australia*, Australian Journal of Earth Sciences, Vol. 57, No. 3, pp. 337-347. doi: 10.1080/08120091003677553

[182] http://e4ftl01.cr.usgs.gov/SRTM/SRTMGL1.003/2000.02.11/

[183] A.K. Skidmore (1989): *A Comparison of Techniques for Calculating Gradient and Aspect from a Gridded Digital Elevation Model*, International Journal of Geographical Information Science, Vol. 3, No 4, pp. 323–334. doi: 10.1080/02693798908941519

[184] P. Getreuer (2011): *Linear Methods for Image Interpolation*, Image Processing On Line, Vol. 1, doi:10.5201/ipol.2011.g_lmii

[185] C. Wiener (1873): *Ueber eine Aufgabe aus der Geometria situs*, Mathematische Annalen Vol. 6, pp. 29–30.

[186] L.R. Ford, Jr (1956): *Network Flow Theory*, The RAND Corporation, Santa Monica, California, paper P-923.

[187] A. Shimbel (1955): *Structure in communication nets*, in Proceedings of the Symposium on Information Networks (New York, 1954), Polytechnic Press of the Polytechnic Institute of Brooklyn, pp. 199–203.

[188] R. Bellman (1958): *On a routing problem*, Quarterly of Applied Mathematics, Vol. 16, pp. 87–90.

[189] E.F. Moore (1959): *The shortest path through a maze*, in Proceedings of an International Symposium on the Theory of Switching, 2–5 April 1957, The Annals of the Computation Laboratory of Harvard University Vol. 30, Harvard University Press, Cambridge, pp. 285–292.

[190] M. Leyzorek - R.S. Gray - A.A. Johnson - W.C. Ladew - S.R. Meaker, Jr - R.M. Petry - R.N. Seitz (1957): *Investigation of Model Techniques - A Study of Model Techniques for Communication Systems*, Case Institute of Technology, Cleveland, Ohio.

[191] E.W. Dijkstra (1959): *A note on two problems in connexion with graphs*, Numerische Mathematik Vol. 1, pp. 269–271. doi:10.1007/BF01386390

[192] G.B. Dantzig (1958): *On the Shortest Route through a Network*, The RAND Corporation, Santa Monica, California, paper P-1345. Published in Management Science, Vol. 6, 1960, pp. 187–190. *On the Shortest Route through a Network*,

[193] A. Schrijver (2012): *On the History of the Shortest Path Problem*, Documenta Mathematica, Extra Volume ISMP, pp. 155-168. doi:10.1016/S0927-0507(05)12001-5

[194] V. Jarník (1930): *O jistém problému minimálním* (Egy bizonyos minimális problémáról), Práce Moravské Přírodovědecké Společnosti, Vol. 6, pp. 57–63. (cseh nyelven)

[195] R. C. Prim: *Shortest connection networks and some generalizations*, Bell System Technical Journal, 36 (1957), pp. 1389–1401. doi:10.1002/j.1538-7305.1957.tb01515.x

[196] J. B. Kruskal, Jr. (1956): *On the Shortest Spanning Subtree of a Graph and the Travelling Salesman Problem*, in Proceedings of Amererican Mathematics Society, Vol. 7, pp. 48-50. doi: 10.1090/S0002-9939-1956-0078686-7

[197] M.L. Fredman - R.E. Tarjan (1987): *Fibonacci Heaps and Their Uses in Improved Network Optimization Algorithms*, Journal of the Association for Computing Machinery, Vol. 34, No. 3, pp. 596-615. doi:10.1109/SFCS.1984.715934

[198] J. Pearl (1984): *Heuristics: Intelligent Search Strategies for Computer Problem Solving*, Addison-Wesley, p. 48. doi:10.1016/S0736-5853(86)80081-8

[199] P.E. Hart - N.J. Nilsson - B. Raphael (1968): *A Formal Basis for the Heuristic Determination of Minimum Cost Paths*, Transactions on Systems Science and Cybernetics, Vol. 4, No. 2, pp. 100–107. doi:10.1109/TSSC.1968.300136

[200] H. Berliner (1979): *The B* Tree Search Algorithm. A Best-First Proof Procedure*, Artificial Intelligence, Vol. 12, No. 1, pp. 3-40. doi:10.1016/0004-3702(79)90003-1

[201] D. de Champeaux - L. Sint (1977): *An improved bidirectional heuristic search algorithm*, Journal of the ACM, Vol. 24, No. 2, pp. 177-191. doi:10.1145/322003.322004

[202] http://www.dis.uniroma1.it/challenge9/format.shtml

[203] R. Geisberger - P. Sanders - D. Schultes - D. Delling (2008): *Contraction Hierarchies: Faster and Simpler Hierarchical Routing in Road Networks*, in Proceedings of the 7th international conference on Experimental Algorithms, WEA'08, pp. 319-333 doi: 10.1007/978-3-540-68552-4_24

[204] D. Delling - T. Pajor - R. F. Werneck (2012): *Round-Based Public Transit Routing*, in Proceedings of the Sixth International Symposium on Combinatorial Search, pp. 130–140. doi: 10.1287/trsc.2014.0534

[205] J. Dibbelt - T. Pajor - B. Strasser - D. Wagner (2013): *Intriguingly Simple and Fast Transit Routing*, in Proceedings of the 12th International Symposium on Experimental Algorithms, pp. 43–54. doi: 10.1007/978-3-642-38527-8_6

[206] B.V. Cherkassky - A.V. Goldberg - T. Radzik (1996): *Shortest paths algorithms: theory and experimental evaluation*, Mathematical Programming, Vol. 73, No. 2, pp. 129-174. doi: 10.1007/BF02592101

[207] C. Miller - A. Tucker - R. Zemlin (1960): *Integer programming formulations and travelling salesman problems*, Journal of the ACM, Vol. 7, pp. 326–329. doi:10.1145/321043.321046

[208] I. Chao - B. Golden - E. Wasil (1996): *Theory and methodology – a fast and effective heuristic for the orienteering problem*, European Journal of Operational Research, Vol. 88, pp. 475-489. doi:10.1016/0377-2217(95)00035-6

[209] T. Tsiligirides (1984): *Heuristic methods applied to orienteering*, Journal of the Operational Research Society, Vol. 35, No. 9, pp. 797-809. doi:10.1057/jors.1984.162

[210] S. Kataoka - S. Morito (1988): *An algorithm for the single constraint maximum collection problem*, Journal of the Operations Research Society of Japan, Vol. 31, No. 4, pp. 515-530.

[211] X. Wang - B. Golden - E. Wasil (2008): *Using a genetic algorithm to solve the generalized orienteering problem*, In: B. Golden - S. Raghavan - E. Wasil (Eds.): The Vehicle Routing Problem: Latest Advances and New Challenges, pp. 263-274. doi:10.1007/978-0-387-77778-8_12

[212] W. Souffriau - P. Vansteenwegen - J. Vertommen - G. Vanden Berghe - D. Van Oudheusden (2008): *A personalised tourist trip design algorithm for mobile tourist guides*, Applied Artificial Intelligence, Vol. 22, No. 10, pp. 964-985. doi: 10.1080/08839510802379626

[213] G. A. Croes (1958): *A method for solving traveling salesman problems*, Operations Research, Vol. 6, pp. 791-812. doi: 10.1287/opre.6.6.791

[214] S. Lin (1965): *Computer solutions of the traveling salesman problem*, Bell Systems Technology Journal, Vol. 44, pp. 2245-2269. doi: 10.1002/j.1538-7305.1965.tb04146.x

[215] S. Lin - B. W. Kernighan (1973): *An Effective Heuristic Algorithm for the Traveling-Salesman Problem*, Operations Research, Volume 21, pp. 498-516. doi:10.1287/opre.21.2.498

[216] R. Ramesh - Y. Yoon - M. Karwan (1992): *An optimal algorithm for the orienteering tour problem*, ORSA Journal on Computing, Vol. 4, pp. 155-165. doi:10.1016/0305-0548(91)90086-7

[217] M. Fischetti - J. Salazar - P. Toth (1998): *Solving the orienteering problem through branch-and-cut*, INFORMS Journal on Computing, Vol.10, pp.133-148. doi: 10.1007/978-3-319-18161-5_17

[218] R. Ramesh - K. Brown (1991): *An efficient four-phase heuristic for the generalized orienteering problem*, Computers and Operations Research, Vol. 18, pp. 151-165. doi: 10.1016/0305-0548(91)90086-7

[219] M. Gendreau - G. Laporte - F. Semet (1998): *A tabu search heuristic for the undirected selective travelling salesman problem*, European Journal of Operational Research, Vol. 106, pp. 539-545. doi:10.1016/S0377-2217(97)00289-0

[220] M. Kantor - M. Rosenwein (1992): *The orienteering problem with time windows*, The Journal of the Operational Research Society, Vol. 43, No. 6, pp. 629-635. DOI: 10.2307/2583018

[221] G. Righini - M. Salani (2006): *Dynamic programming for the orienteering problem with time windows*, Technical Report No. 91, Dipartimento di Tecnologie dell'Informazione, Universita degli Studi Milano, Crema, Italy. doi:10.1016/j.cor.2008.01.003

[222] C. Barnhart - E.L. Johnson - G.L. Nemhauser - M.W.P. Savelsbergh - P.H. Vance (1998): *Branch-and-Price: Column Generation for Solving Huge Integer Programs*, Operations Research, Vol. 46, No. 3, pp. 316-329.

[223] S. Butt - D. Ryan (1999): *An optimal solution procedure for the multiple tour maximum collection problem using column generation*, Computers and Operations Research, Vol. 26, pp. 427-441. doi:10.1016/S0305-0548(98)00071-9

[224] S. Boussier - D. Feillet - M. Gendreau (2007): *An exact algorithm for the team orienteering problem*, 4OR, Vol. 5, pp. 211-230. doi:10.1007/s10288-006-0009-1

[225] I. Chao - B. Golden - E. Wasil (1996): *Theory and methodology – the team orienteering problem*, European Journal of Operational Research, Vol. 88, pp. 464-474. doi: 10.1016/0377-2217(94)00289-4

[226] H. Tang - E. Miller-Hooks (2005): *A tabu search heuristic for the team orienteering problem*, Computer and Operations Research, Vol. 32, pp.1379-1407. doi:10.1016/j.cor.2003.11.008

[227] C. Archetti - A. Hertz - M. Speranza (2007): *Metaheuristics for the team orienteering problem*, Journal of Heuristics, Vol. 13, pp. 49-76. doi: 10.1007/s10732-006-9004-0

[228] L. Ke - C. Archetti - Z. Feng (2008): *Ants can solve the team orienteering problem*, Computers and Industrial Engineering, Vol. 54, pp. 648-665. doi:10.1016/j.cie.2007.10.001

[229] P. Vansteenwegen - W. Souffriau - G. Vanden Berghe - D. Van Oudheusden (2009): *A guided local search metaheuristic for the team orienteering problem*, European Journal of Operational Research, Vol. 196, No. 1, pp.118-127. doi:10.1016/j.ejor.2008.02.037

[230] P. Vansteenwegen - W. Souffriau - G. Vanden Berghe - D. Van Oudheusden (2009): *Metaheuristics for tourist trip planning*, In: M. Geiger - W. Habenicht - M. Sevaux - K. Sörensen (eds.): Metaheuristics in the Service Industry, Lecture Notes in Economics and Mathematical Systems, Vol. 624, pp. 15–31. doi:10.1016/j.cor.2015.03.016

[231] F. Tricoire - M. Romauch - K. Doerner - R. Hartl (2010): *Heuristics for the multi-period orienteering problem with multiple time windows*, Computers and Operations Research, Vol. 37, No. 2, pp. 351–367. doi:10.1016/j.cor.2009.05.012

[232] P. Hansen - N. Mladenovic - J.A.M. Perez (2010): *Variable neighbourhood search: methods and applications*, Annals of Operations Research, Vol. 175, pp. 367-407. doi: 10.1007/s10288-008-0089-1

[233] P. Vansteenwegen - W. Souffriau - G. Vanden Berghe - D. Van Oudheusden (2009): *Iterated local search for the team orienteering problem with time windows*, Computers and Operations Research, Vol. 36, No. 12, pp. 3281-3290. doi:10.1016/j.cor.2009.03.008

[234] M. Schilde - K. Doerner - R. Hartl - G. Kiechle (2009): *Metaheuristics for the bi- objective orienteering problem*, Swarm Intelligence, Vol. 3, pp. 179-201. doi: 10.1007/s11721-009-0029-5

[235] M. Gendreau - G. Laporte - F. Semet (1998): *A branch-and-cut algorithm for the undirected Selective Travelling Salesman Problem*, Networks, Vol. 32, pp. 263-273. DOI: 10.1002/(SICI)1097-0037(199812)32:4<263::AID-NET3>3.0.CO;2-Q

[236] T. Ilhan - S. Iravani - M. Daskin (2008): *The orienteering problem with stochastic profits*, IIE Transactions, Vol 40, pp. 406-421. DOI:10.1080/07408170701592481

[237] E. Triantaphyllou (2000): *Multi-Criteria Decision Making: A Comparative Study*, Dordrecht, The Netherlands: Kluwer Academic Publishers (now Springer), p. 320. DOI: 10.1007/978-1-4757-3157-6

[238] C. Archetti - D. Feillet - A. Hertz - M. Speranza (2009): *The capacitated team orienteering and profitable tour problems*, Journal of the Operational Research Society, Vol. 60, pp. 831-842. doi:10.1057/palgrave.jors.2602603

[239] L. Muyldermans - P. Beullens - D. Cattrysse - D. Van Oudheusden (2005): *Exploring variants of 2- and 3-opt for the general routing problem*, Operations Research, Vol. 53, No. 6, pp. 982-995. doi:10.1287/opre.1040.0205

[240] P. Vansteenwegen - D. Van Oudheusden (2007): *The mobile tourist guide: An or opportunity*, OR Insights, Vol. 20, No. 3, pp. 21-27. doi:10.1057/ori.2007.17

[241] E. Erkut - J. Zhang (1996): *The maximum collection problem with time-dependent rewards*, Naval Research Logistics, Vol. 43, No. 5, pp. 749-763, DOI: 10.1002/(SICI)1520-6750

[242] H. Tang - E. Miller-Hooks - R. Tomastik (2007): *Scheduling technicians for planned maintenance of geographically distributed equipment*, Transportation Research, Part E: Logistics and Transportation Review, Vol. 43, No. 5, pp. 591-609. doi:10.1016/j.tre.2006.03.004

[243] G. Erdogan - G. Laporte (2013): *The orienteering problem with variable profits*, Networks, Vol. 61, No. 2, pp. 104-116. DOI: 10.1002/net.21496

[244] R. A. Abbaspour - F. Samadzadegan (2011): *Time-dependent personal tour planning and scheduling in metropolises*, Expert Systems and Applications, Vol. 38, pp. 12439-12452. doi: 10.1016/j.eswa.2011.04.025

[245] A. Garcia - P. Vansteenwegen - O. Arbelaitz - W. Souffriau - M. T. Linaza (2013): *Integrating public transportation in personalised electronic tourist guides*, Computers & Operations Research, Vol. 40, No. 3, pp. 758-774. doi:10.1016/j.cor.2011.03.020

[246] E.H.C. Lu - C.Y. Lin - V.S. Tseng (2011): *Trip-Mine: An Efficient Trip Planning Approach with Travel Time Constraints*, in Proceedings of the IEEE 12th International Conference on Mobile Data Management, Vol. 1, pp. 152-161. doi:10.1109/MDM.2011.13

[247] D. Gavalas - C. Konstantopoulos - K. Mastakas - G. Pantziou - N. Vathis (2015): *Heuristics for the Time Dependent Team Orienteering Problem: Application to Tourist Route Planning*, Computers & Operations Research, Vol. 62, pp. 36-50. doi:10.1016/j.cor.2015.03.016

[248] A. Divsalar - P. Vansteenwegen - D. Cattrysse (2013): *A variable neighborhood search method for the orienteering problem with hotel selection*, International Journal of Production Economics, Vol. 145, No. 1, pp.150-160. doi:10.1016/j.ijpe.2013.01.010

[249] W. Souffriau - P. Vansteenwegen - G. Vanden Berghe - D. Van Oudheusden (2011): The planning of cycle trips in the province of East Flanders, Omega, Vol. 39, No. 2, pp.209-213. doi: 10.1016/j.omega.2010.05.001

[250] A. Yahi - A. Chassang - L. Raynaud - H. Duthil - D. H. Chau: *Aurigo - An Interactive Tour Planner for Personalized Itineraries*, in Proceedings of the 20th International Conference on Intelligent User Interfaces, pp. 275-285. doi: 10.1145/2678025.2701366

[251] M. De Choudhury - M. Feldman - S. Amer-Yahia - N. Golbandi - R. Lempel - C. Yu (2010): *Automatic construction of travel itineraries using social breadcrumbs*, in Proceedings of the 21st ACM conference on Hypertext and Hypermedia, pp. 35-44. doi: 10.1145/1810617.1810626

[252] A. Popescu - G. Grefenstette (2009): *Deducing trip related information from flickr*, in Proceedings of the 18th international conference on World wide web (WWW'2009), pp. 1183-1184. doi: 10.1145/1526709.1526919

[253] C. Lucchese - R. Perego - F. Silvestri - H. Vahabi - R. Venturini (2012): *How Random Walks Can Help Tourism*, in Proceedings of the 34th European Conference on IR Research, pp. 195-206. doi:10.1007/978-3-642-28997-2_17

[254] K.H. Lim (2015): *Recommending Tours and Places-of-Interest based on User Interests from Geo-tagged Photos*, in Proceedings of the 2015 ACM SIGMOD on PhD Symposium, pp. 33-38. doi: 10.1145/2744680.2744693

[255] M. Castro - K. Sörensen - P. Vansteenwegen - P. Goos (2015): *A fast metaheuristic for the travelling salesperson problem with hotel selection*, 4OR quarterly journal of the Belgian, French and Italian Operations Research Societies, Vol. 13, No. 1, pp. 15-34. DOI: 10.1007/ s10288-014-0264-5

[256] C. Verbeeck - K. Sörsensen - E.H. Aghezzaf - P. Vansteenwegen (2014): *A fast solution method for the time-dependent orienteering problem*, European Journal of Operational Research, Vol. 236, pp. 419-432. doi:10.1016/j.ejor.2013.11.038

[257] G. Birkhoff (1946): *Tres observaciones sobre el algebra lineal*, Revista Facultad de Ciencias Exactas, Puras y Aplicadas Universidad Nacional de Tucuman, Serie A (Matematicas y Fisica Teorica), Vol. 5, pp- 147-151.

[258] O. Boruvka (1926): *O jistém problému minimálním [On a minimal problem]*, Práce Moravské Príirodovedecké Spolecnosti, Brno [Acta Societatis Scientiarum Naturalium Moravicae], Vol. 3, pp. 37-58. (cseh nyelven, német előszóval)

[259] E.R. Swanson (1979): *Geometric Dilution of Precision*, Journal of the Institution of Navigation, Vol. 25, No. 4, pp. 425-429. DOI: 10.1002/j.2161-4296.1978.tb01345.x

[260] M.S. Goh - D.F. Shen - S.H. Hong (2007): *Processing of GPS Data with Difference HDOP in Guide Robot for the Visually Impaired*, International Journal of Computer Science and Network Security, Vol. 7, No.10, pp. 90-97. DOI: 10.1007/BF02521054

[261] D. Ciresan - U. Meier - J. Masci - L.M. Gambardella - J. Schmidhuber (2011): *Flexible, High Performance Convolutional Neural Networks for Image Classification*, Proceedings of the Twenty-Second international joint conference on Artificial Intelligence, Vol. 2, pp. 1237–1242. doi: 10.5591/978-1-57735-516-8/IJCAI11-210

[262] D. Gavalas - C. Konstantopoulos - K. Mastakas - G. Pantziou - Y. Tasoulas (2013): *Cluster-based heuristics for the team orienteering problem with time windows*, in Proceedings of 12th International Symposium on Experimental Algorithms, pp. 390-401. doi: 10.1007/978-3-642-38527-8_34

[263] L.M. Gambardella - R. Montemanni - D. Weyland (2012): *Coupling ant colony systems with strong local searches*, European Journal of Operational Research, Vol. 220, No. 3, pp. 831-843. DOI: 10.1016/j.ejor.2012.02.038

[264] M. Gulliksson - P. Wedin (1992): Modifying the QR-Decomposition to Constrained and Weighted Linear Least Squares, SIAM Journal on Matrix Analysis and Applications, Vol. 13, No. 4, pp. 1298-1313. doi:10.1016/S0024-3795(02)00262-8

[265] J.A. Hartigan - M.A. Wong (1979): *Algorithm AS 136: A K-Means Clustering Algorithm*, Journal of the Royal Statistical Society, Series C, Vol. 28, No. 1, pp. 100-108. DOI: 10.2307/2346830

[266] G. Gutin - A. Yeo - A. Zverovich (2002): *Traveling salesman should not be greedy: domination analysis of greedy-type heuristics for the TSP*, Discrete Applied Mathematics, Vol. 117, pp. 81-86. doi:10.1016/S0166-218X(01)00195-0

[267] K. Asdemir - J.S. Varghese - K. Ramayya (2009): *Dynamic pricing of multiple home delivery options*, European Journal of Operational Research, Vol. 196, No. 1, pp. 246–257.10.1016/j.ejor. 2008.03.005

[268] X. Yang - A.K. Strauss - C.S.M. Currie - R. Eglese (2013): *Choice-Based Demand Management and Vehicle Routing in E-Fulfillment*, Transportation Science, Vol. 50, No. 2, pp. 473 - 488, DOI: 10.1287/trsc.2014.0549

[269] M. De Choudhury - M. Feldman - S. Amer-Yahia - N. Golbandi - R. Lempel - C. Yu (2010): *Automatic construction of travel itineraries using social breadcrumbs*, in Proceedings of the 21st ACM conference on Hypertext and Hypermedia, pp. 35-44. doi: 10.1145/1810617.1810626

[270] A. Bolat (1999): *Assigning arriving flights at anairport to the available gates*, Journal of the Operational Research Society, Vol. 50, No. 1, pp. 23–34.

[271] B. Maharjan - T. I. Matis (2012): *Multi-commodity flow network model of the flight gate assignment problem*, Computers and Industrial Engineering, Vol. 63, No. 4, pp. 1135–1144.

[272] A. Bouras - M.A. Ghaleb - U.S. Suryahatmaja - A.M. Salem (2014): *The Airport Gate Assignment Problem: A Survey*,The Scientific World Journal, Vol. 2014, Article ID 923859, pp. 1-27, dx.doi.org/10.1155/2014/923859

[273] F.W. Cathey - D.J. Dailey (2003): *A prescription for transit arrival/departure prediction using automatic vehicle location data*, Transportation Research Part C: Emerging Technologies, Vol. 11, No. 3-4, pp. 241–264, DOI: 10.1016/S0968-090X(03)00023-8

[274] T.S. Rappaport - J.H. Reed - B.D. Woerner (2002): *Position location using wireless communications on highways of the future*, IEEE Communications Magazine, Vol. 34, No. 10, pp. 33-41, DOI: 10.1109/35.544321

[275] M.J. Lighthill - G.B. Whitham (1955): *On kinematic waves. II. A theory of traffic flow on long crowded roads*, Proceedings of Royal Society A, Vol. 229, pp. 281–345.

[276] P.I. Richards (1956): *Shockwaves on the highway*, Operations Research, Vol. 4, pp. 42–51.

[277] S. Logghe - L.H. Immers (2008): *Multi-class kinematic wave theory of traffic flow*, Transportation Research Part B, Vol. 42, pp. 523–541, DOI:10.1016/j.trb.2007.11.001

[278] K. E. Watkins - B. Ferris - A. Borning - S. G. Rutherford - D. Layton (2011): Where Is My Bus? Impact of mobile real-time information on the perceived and actual wait time of transit riders. Transportation Research Part A, Vol. 45, pp 839-848.

[279] V.W. Stover - E.D. McCormack (2012): *The Impact of Weather on Bus Ridership in Pierce County, Washington*, The Journal of Public Transportation, Vol. 15, No. 1, pp. 95-110.

[280] N.H. Vu - A.M. Khan (2010): *Bus running time prediction using a statistical pattern technique*, Transportation Planning and Technology, Vol. 33, No. 7, pp. 625-642

[281] Zs. Sándor - Cs. Csiszár (2016): *Method for analysis and prediction of dwell times at stops in local bus transportation*, Transport, May 2016, DOI: 10.3846/16484142.2016.1190402

Saját publikációk a témakörben

Referált szakmai folyóiratok magyar nyelven

Apáthy M. Sándor [2016]: *Egy heurisztikus útvonaltervező algoritmus többnapos túrák tervezésére*, Szigma Matematikai-közgazdasági folyóirat, Vol. 3-4, elfogadva

Apáthy M. Sándor [2016]: Az útvonaltervező algoritmusok történeti áttekintése, különös tekintettel azok turisztikai célú alkalmazásaira, Alkalmazott Matematikai Lapok, Vol. 33, elfogadva

Apáthy M. Sándor [2016]: A menetidőbecslés alkalmazásai, Közlekedéstudományi Szemle, elfogadva

Apáthy M. Sándor [2016]: Turistatípusok azonosítása - Egy lehetséges turisztikai ajánlórendszer, Vezetéstudomány, bírálat alatt

Egyéb publikációk magyar nyelven

Apáthy M. Sándor [2016]: *Földfelszíni gyalogos közlekedés modellezése*, Innováció és fenntartható felszíni közlekedés Konferencia 2016. Budapest, Magyarország, 2016.08.29 - 2016.08.31, paper 22, ISBN: 978-963-88875-2-8

Apáthy M. Sándor [2011]: Fejezetek a modern közgazdaságtudományból – recenzió [Móczár József: Fejezetek a modern közgazdaságtudományból. Akadémiai Kiadó, Budapest. 2008. 608 oldal ISBN 9789630585378], Gazdaság és társadalom, Vol. 3, No. 3-4, pp. 189-194. ISSN 0865-7823

Referált szakmai folyóiratok idegen nyelven

Apáthy M. Sándor [2016]: *Personalised hiking time estimation*, Pure Mathematics and Applications, 10497-PU.M.A.-AOP_2016-09-20

Apáthy M. Sándor [2016]: *Practical Route Planning Algorithm,* Periodica Polytechnica Transportation Engineeering, Vol 45, No. 2, elfogadva


25 km
IMPASSABLE