Közgazdasági Szemle, LIV. évf., 2007. április (376–383. o.)

BENEDEK GÁBOR

„Dupla vagy semmi”

Duplikációbecslés szimulációs módszerekkel

A tanulmányban bemutatjuk az adatbázisokban fellelhető duplikációk számának mintából való becslését szimulációs módszerek segítségével. A kutatáshoz egy valós vállalati adatbázist használtunk. A szimuláció során három eltérő elméleti modellt vizsgáltunk meg és hasonlítottunk össze.*

Journal of Economics Literature (JEL) kód: C15, C80, M31

A 21. században az ügyfélkapcsolatok irányításának *(Customer Relationship Management, CRM)* világában élünk. Fogyasztási, viselkedési szokásaink, társadalmi kapcsolataink, valamint demográfiai adataink óriási adatbázisokban tárolva és elemezve megtalálhatók a nagyobb multinacionális vállalatokban, amelyek ezek segítségével alakítják termékfejlesztéseiket, marketing- és értékesítési stratégiájukat. Valamennyien érezzük ennek pozitív és negatív hatásait, amikor személyre szabott levelekkel *(direct mail, DM),* sms/mms-kampányokkal, vagy személyes telefonhívással kapunk információt vagy kedvező vásárlási ajánlatot egy-egy termékről, szolgáltatásról.

Az óriási méretű ügyféladatbázisok karbantartása nem egyszerű feladat, különösen akkor nem, ha a vállalat kezdetben nem helyezett megfelelő hangsúlyt az adatok tisztaságára, vagy amikor több rendszer különböző módon szervezett adatait kell közös nevezőre hozni (például egy felvásárlás esetén). Az egyik legnagyobb feladatot az ügyfélduplikációk elkerülése jelenti, azaz annak megoldása, hogy ugyanaz az ügyfél ne szerepeljen több különböző azonosító kód alatt, ne tűnjön úgy egy vállalat számára, hogy az több különböző ügyfél. Bármilyen furcsa, sokszor maguk a vállalatok képtelenek pontos választ adni arra a kérdésre, hogy összesen hány ügyfelük van.

A duplikációk megtalálása nagy adatbázisban nem könnyű feladat. Számos automatikus szövegfeldolgozó program segítheti a keresést, de a legtöbb esetben nem kerülhető el a manuális ellenőrzés. (Gondoljunk például arra, hogy két gyakori vezetéknevű ügyfél még akkor is viszonylag nagy valószínűséggel lehet különböző, ha ugyanaz a keresztnevük, címük, esetleg foglalkozásuk! Ezzel szemben egy nagyon ritka vezetéknév esetén még akkor is elképzelhető duplikáció, ha maga a vezetéknév eltér, például egyikben *i*, másikban *y* szerepel.) Éppen ezért a duplikációk megtalálása és kiszűrése hosszadalmas és költséges feladat. Ha egy vállalat ilyen manuális tisztítást is tartalmazó munkába kezd, fontos, hogy tisztában legyen azzal, mekkora feladatot vesz a nyakába, mekkora lehet a teljes ügyfélkör duplikációja, illetve elég nagy-e a duplikáció ahhoz, hogy megérje megtisztítani az adatbázist.

A továbbiakban azt szeretnénk bemutatni, hogy a duplikációk számának mintából való becslése bonyolult feladat, amelyet érdemes szimulációs módszerek segítségével kezelni.

* Szeretnénk köszönetet mondani az SPSS Hungary Kft.-nek, hogy a kutatáshoz szükséges szoftvereket rendelkezésemre bocsátotta.

Benedek Gábor, Budapesti Corvinus Egyetem.

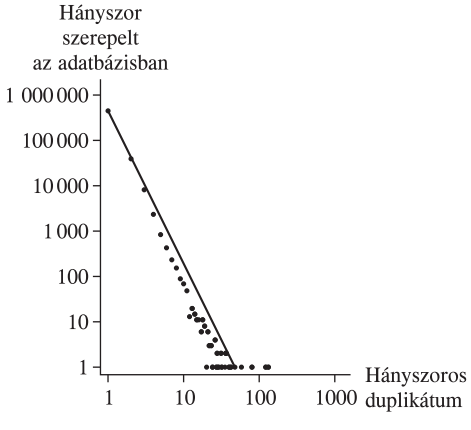
Adatbázis

Kutatásunkat egy valós vállalat valós ügyféladatbázisának segítségével mutatjuk be. Az elemzés értelmezéséhez az iparág és a vállalat megnevezése lényegtelen, fontos azonban, hogy kellően nagy adatbázis állt rendelkezésre. Ez esetünkben egy több mint félmillió ügyfelet tartalmazó adatbázis volt, ahol természetesen ez a szám nem feltétlenül a valóban különböző, hanem a vállalat által különbözőnek tekintett (különböző azonosítóval tárolt) ügyfeleket jelenti. Első lépésünk az volt, hogy automatikus eljárás segítségével megkíséreltük azonosítani a duplikációkat. Az automatikus eljárás során a név, a cím, a munkahely, a szabadon megadott egyéb demográfiai jellemzők (telefonszám, kor) és a belépés dátuma segítségével olyan algoritmust alkottunk, amely a nagyon nagy valószínűséggel azonos ügyfeleket tudta összekapcsolni, azaz feltételeztük, hogy a manuális ellenőrzés során ezekhez további duplikátumok kerülnek. Vizsgáljuk meg, milyen duplikációeloszlást kaptunk az automatikus kereső algoritmus alkalmazása során!

Az *1. ábrán* log-log skálán ábrázoltuk a duplikációs eloszlást. Jól látható, hogy a legtöbb ügyfél egyszer szerepel az adatbázisban. Ugyanakkor 40 ezer ügyfél kétszer, kilencezer ügyfél háromszor stb. Sőt, az sem kizárt, hogy egy ügyfél ötvennél több alkalommal szerepeljen, de a továbbiakban őket extrém esetnek tekintjük, és nem foglalkozunk velük. Feltételezzük, hogy a valós duplikációeloszlás hasonló, azaz lineáris csökkenést mutat egy log-log skálán. Az automatikus algoritmussal becsült duplikáció 13 százalék volt, ahol:

$$duplikáció = 1 - \frac{különböző\ ügyfelek}{összes\ azonosító}.$$

1. ábra
Duplikációeloszlás automatikus algoritmussal

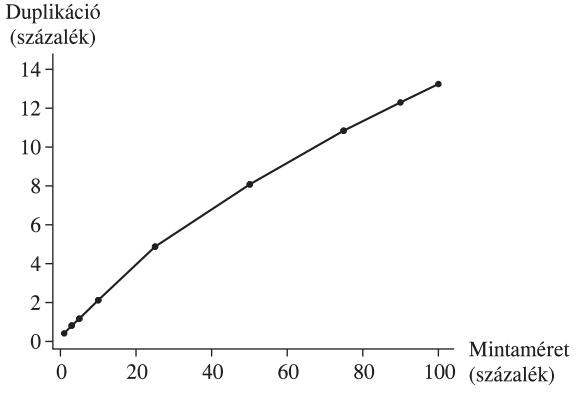


Módszerünk a valós duplikáció becslésére a következő. Veszünk egy kis mintát a teljes adatbázisból. Meghatározzuk a duplikáció eloszlását manuális módszerrel, majd ugyanezt az eloszlást felvetítjük a teljes sokaságra, és megbecsüljük a valódi duplikációt.

Mintavétel

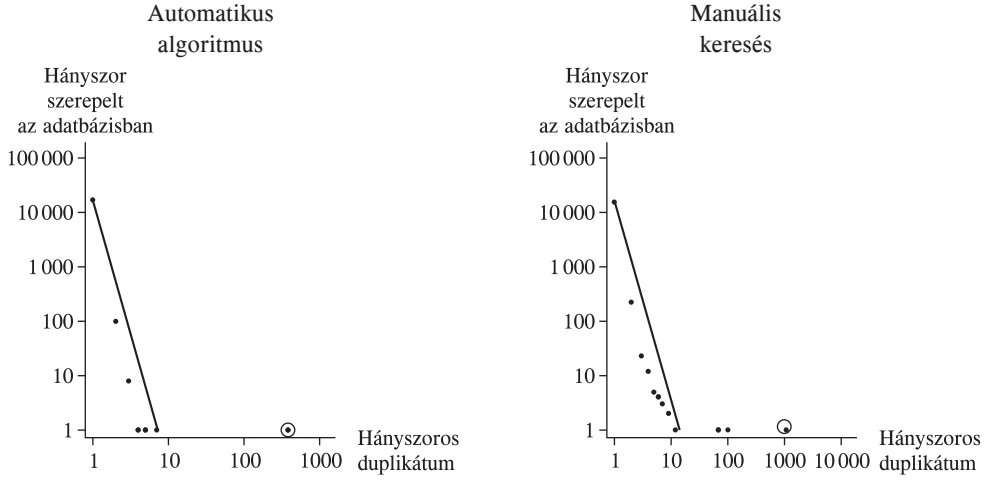
A kis mintának az az előnye, hogy a költséges manuális eljárást kisméretű halmazon kell megvalósítani. Hátránya azonban az, hogy minél kisebb a minta, annál kisebb valószínűséggel találunk duplikációt. Az automatikus duplikációkereső algoritmus segítségével megnézhetjük, hány duplikációt találunk különböző méretű mintákban *(2. ábra)*.

2. ábra
Található duplikációk száma a mintaméret függvényében



Természetesen minél kisebb mintát veszünk, annál nagyobb a talált duplikáció szórása. Ezért végül viszonylag nagy, 3 százalékos mintát vettünk, és ezen a mintán az automatikus algoritmus mellett manuálisan is vizsgáltuk a duplikációt. A *3. ábrán* jól látható a duplikáció eloszlása automatikus és manuális eljárások mellett. Míg az előbbi esetben a 3 százalékos mintán talált duplikáció 0,8 százalék, addig az utóbbi, manuális eljárással kiegészített esetben 1,3 százalék, azaz több mint másfélszerese. (Az ábrákon jól látszik egy kiugró, bekarikázott pont, amelyet extrém esetként kezeltünk, és hatását figyelmen kívül hagytuk.)

3. ábra
Duplikációeloszlás a 3 százalékos mintán (automatikus algoritmussal és manuális kereséssel)



A kérdés tehát a következő. Ha az automatikus algoritmus a 3 százalékos mintán 0,8 százalékos duplikációt mutat (a *3. ábra* bal oldala), 100 százalékos mintán pedig 13 százalékosat, a manuális keresés a 3 százalékos mintán 1,3 százalékos duplikációt mutat (a *3. ábra* jobb oldala), akkor mennyi lenne a 100 százalékos mintán a manuális keresés által található duplikáció?

A szimuláció

A szimuláció során adott paraméterek mellett előállítjuk a teljes sokaság reprezentációját. Ezt követően veszünk belőle egy 3 százalékos mintát, és megvizsgáljuk, mennyire illeszkedik az általunk kapott valódi, manuális kereséssel kapott duplikációs eloszláshoz. Ezek után addig finomítjuk a teljes sokaságra vonatkozó paraméterek becslését, ameddig a mintavétel során a lehető legpontosabb illeszkedést nem kapjuk. Így végül megkapjuk a teljes sokaság duplikációját, valamint duplikációs eloszlását.

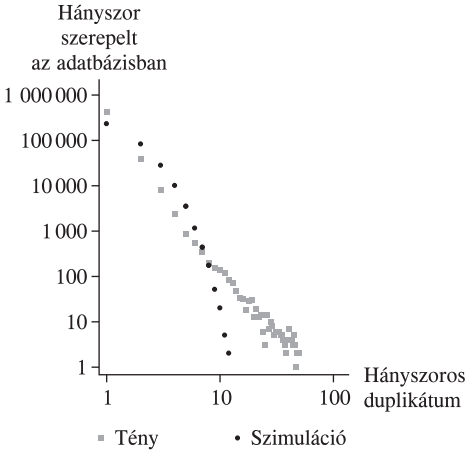
A teljes sokaság virtuális reprezentációja egyszerű feladat. A duplikációkat megfelelő algoritmus segítségével kell reprodukálni, és olyan algoritmust kell alkalmazni, amely megfelelő módon adja vissza a duplikációs eloszlást. Kutatásunk során három (a társadalmi hálózatelméletben is alkalmazott[1]) algoritmussal próbálkoztunk.

1. modell

Az első ügyfél adott. A következő ügyfél p valószínűséggel azonos, mint az előző, $1 - p$ valószínűséggel pedig különbözik minden eddigi ügyféltől.

A *4. ábra* alapján láthatjuk, hogy amikor ezzel az algoritmussal próbálkoztunk, akkor a duplikációs eloszlás nem lineáris csökkenést mutatott a log-log grafikonon, mint ahogy azt az az eredeti adatokon láttuk. Amennyiben ezzel az algoritmussal becsüljük a manuális keresés duplikációra legjobban illeszkedő paramétert, akkor azt kapjuk,

4. ábra
Duplikációeloszlás a teljes mintán (valóságos és szimulált duplikációk)



[1] A hálózatelméletben gyakran generálunk hasonló módszertan segítségével véletlen gráfokat. A hálózatelmélet robbanásszerű fejlődésével számos elméleti eredmény született, amelyek a jelen problémára is adnak analitikus eredményeket. E dolgozat célja tisztán az empirikus kutatás bemutatása.

hogy $p = 0{,}35$, emiatt a feltételezett teljes adatbázisunkban található teljes duplikáció manuális keresés esetén körülbelül 30 százalékot érne el. Ugyanakkor látható, hogy illeszkedésünk pontatlan. A 2-től 6-ig terjedő duplikációk valószínűségét erősen felülbecsüljük, az ennél többszörös duplikációkat viszont alul. Olyan nagymértékű a torzítás, hogy 11-nél többszörös duplikációt egyáltalán nem produkált az algoritmus. Feladatunk tehát egy olyan algoritmus megtalálása, amelyik képes reprodukálni egy lineárisan csökkenő duplikációs eloszlást.

Ezzel az algoritmussal az a gond, hogy a következő ügyfelek generálása független az előzően generált ügyfelektől. Ezért annak a valószínűsége, hogy valaki sokszoros duplikátum lesz, nagyon kicsi. A következő algoritmusunk olyan, hogy folyamatosan növeli a duplikáció valószínűségét, attól függően, hogy hány duplikátum van a generálás pillanatáig.

2. modell

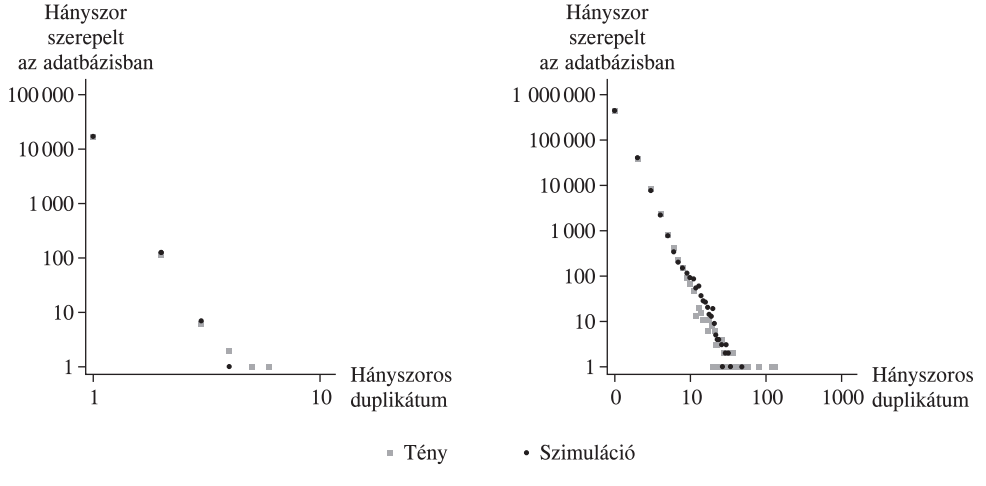
Folyamatosan generálunk ügyfeleket. Amennyiben az utoljára generált ügyfélből még csak egy van, akkor p_1 valószínűséggel generálunk egy azonosat, és $1 - p_1$ valószínűséggel egy olyat, amelyet eddig még nem generáltunk. Amennyiben az így generált ügyfél duplikátum, úgy p_2 valószínűséggel generálunk egy azonosat és $1 - p_2$ valószínűséggel egy olyat, amelyet eddig még nem generáltunk, és így tovább. Természetesen, ha új, nem duplikált ügyfelet generáltunk, akkor a következő lépésben visszatérünk a p_1 valószínűséghez. Feltételezzük, hogy $p_1 \leq p_2 \leq p_3 \leq \ldots \leq 1$.

Az első algoritmus ennek egy speciális esete, amikor $p_1 = p_2 = p_3 = \ldots = p$. Esetünkben a legjobb illeszkedést a $p_i = 0{,}145i - 0{,}025$ adta, egy olyan korlátozással, hogy ne érhessük el az 1 valószínűséget, éspedig $p_j = 0{,}8$, ha $j > 6$. Ezzel a módszerrel sikerült reprodukálni a duplikáció eloszlását a teljes mintán, mint ahogy azt az *5. ábra* mutatja.

Az ábra alapján jól látható, hogy az új algoritmus jól illeszkedik a sokaságra és a mintára is, ezért elvégeztük az illesztést a manuálisan vizsgált mintára (az ábra bal oldala). A paraméterek meghatározása után azt az eredményt kaptuk, hogy a teljes sokaságon a manuális keresés végrehajtása után 16,5 százalékos duplikációt várhatunk. Végül egy harmadik algoritmust is elkészítettünk.

5. ábra

Duplikáció eloszlás a mintán és a teljes sokaságon (valóságos és szimulált duplikációk)



3. modell

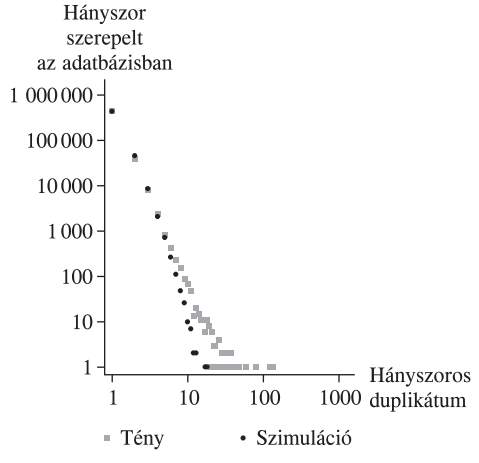
Az első ügyfél adott. A következő ügyfelek esetén p valószínűséggel egy már korábban generált ügyféllel azonosat generálunk, $1 - p$ valószínűséggel pedig egy újat. Ha egy régebbi ügyféllel azonosat generálunk, akkor az összes eddig generált ügyfél közül egyenletes valószínűséggel választunk, de az eddig generált mintában található számosságuk szerinti súlyozással.

Tegyük fel például, hogy a mintában már 6 ügyfelet generáltunk: *A; B; C; B; D; D*. A hetedik ügyfél generálásánál $1 - p$ annak a valószínűsége, hogy új, *E* ügyfelet generálunk és p annak, hogy egy korábbit. Méghozzá $p/6$ valószínűséggel *A*-t, $p/3$ valószínűséggel *B*-t, $p/6$ valószínűséggel *C*-t és $p/3$ valószínűséggel *D*-t.

A *6. ábrán* az utolsó modell által generált duplikációs eloszlást láthatjuk. A mintára a $p = 0{,}131$ érték illeszkedett a legjobban.

6. ábra

Duplikációeloszlás a teljes mintán (valóságos és szimulált duplikációk)

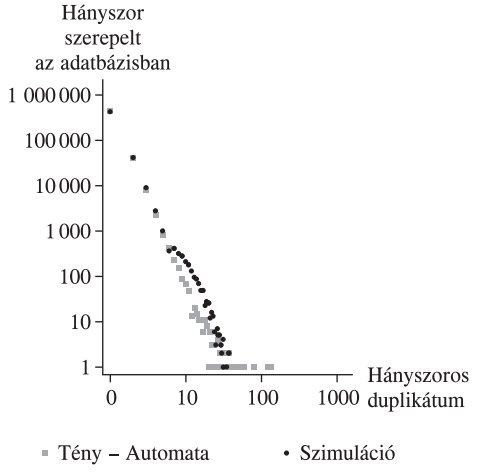


Következtetések

Az első modellünk tehát alkalmatlan volt a valós duplikáció becslésére. Míg az automatikus algoritmus 13 százalékos egyezőséget talált a teljes sokaságban, addig a manuális keresés esetén a duplikációt az első algoritmus alapján 30 százalékosnak becsültük volna. Ez azt jelentette volna, hogy az ügyfelek további 17 százaléka hibásan szerepel az adatbázisban, és ebben az esetben érdemes lenne egy költséges beruházással manuálisan megtisztítani a teljes ügyfélkört.

Láttuk azonban, hogy ez az egyszerű algoritmus erősen torzítja a megfigyelhető duplikációeloszlást. Ezért a második algoritmus segítségével olyan paraméterbeállításra törekedtünk, hogy az a lehető legjobban illeszkedjen a megfigyelhető duplikációs eloszláshoz. Ebben az esetben az automatikus 13 százalékos duplikáció vélhetően csak 16,5 százalékra növekedne, amennyiben a manuális eljárást végrehajtanánk. Ez mindösszesen 3,5 százalékpontos hibát jelent jelenleg, és ez már nem biztos, hogy indokolja a költséges kézi ügyfélkörtisztítást. Vegyük észre, hogy illesztés szempontjából semmi különbség nem volt az 1. és 2. modell között. Ugyanazt a 3 százalékos mintát használtuk, amelyen a manuális tisztítást végrehajtottuk. A különbség annyi, hogy az elsőben egyetlen para-

7. ábra
Duplikációeloszlás a teljes mintán (automatikus és manuális keresés)



1. táblázat
Duplikációeloszlás a teljes mintán (automatikus keresés)
(duplikácó = 13,0 százalék)

Darab	Méret	Összes eset
444 267	1	444 267
39 669	2	79 338
8 033	3	24 099
2 313	4	9 252
835	5	4 175
421	6	2 526
231	7	1 617
152	8	1 216
88	9	792
69	10	690

2. táblázat
Becsült duplikációseloszlás a teljes mintán (manuális keresés)
(duplikácó = 16,5 százalék)

Darab	Méret	Összes eset
419 871	1	419 871
42 006	2	84 012
8 968	3	26 904
2 793	4	11 172
987	5	4 935
366	6	2 196
416	7	2 912
315	8	2 520
281	9	2 529
213	10	2 130

métert (p) kellett megbecsülnünk, a másodikban négyet. (A p_i egyenlet két együtthatóját, j-t és p_j-t.) Végül ezeket az eredményeket fogadtuk el, és a *7. ábrán,* valamint az *1.* és a *2. táblázatban* láthatjuk a teljes sokaságra az automatikus (és egyben megfigyelhető), valamint a manuális (és ezért becsült) duplikációs eloszlást.

Módszertani szempontból azonban van jelentősége a 3. modellnek is, annak ellenére, hogy a végső következtetéseket nem ebből vontuk le. A harmadik modell ugyanis szintén egyparaméteres (p). Mégis, a megfigyelhető eloszláshoz jóval hasonlóbb eredményeket tudott produkálni, mint az első algoritmus. Ráadásul gyakorlati (közgazdasági) megfontolásból is könnyebben magyarázható, mint a 2. modell. A vizsgált vállalat ugyanis fokozatosan építette az ügyfélkörét. Amikor egy új ügyfél érkezett, nem tudta kellő pontossággal vizsgálni, hogy nem tévedett-e az ügyfél, és nem szerepel-e már az adatbázisban. Nyilvánvalóan egy régebben már regisztrálódott ügyfél nagyobb valószínűséggel jelenik meg újra, majd ismét, és így tovább. Időközben javult a vállalat informatikai környezete, nagyobb pontossággal volt képes ellenőrizni az esetleges duplikációt. Így strukturális törés következett be a modellben, valójában két p értéket kellene modellezni, egy strukturális törés előttit (p_1) és egy utánit (p_2). Ilyen mélységben azonban nem álltak rendelkezésre az adatok, így szimulációs vizsgálatokat sem kívántunk végezni.

Záró gondolatok

Sikerült tehát bebizonyítani, hogy a szimulációs módszertan és megfelelő modellválasztás esetén becsülhető a valós duplikációszám, így egy hosszú és költséges adattisztítási fázis megkezdése előtt el lehet dönteni annak hasznosságát, megtérülését. (A példában bemutatott vállalat döntése az volt, hogy nem éri meg az automatikus keresésnél mélyebben tisztítani az adatokat.) Az alapgondolat nem új, mintát veszünk a sokaságból, és a mintában fellelhető összefüggések felfedezésével állításokat fogalmazunk meg a teljes sokaságra. Az üzleti szféra és a gazdaságpolitika szereplői számára a statisztikai modellezés, az adatbányászat vagy a szimuláció bonyolult és költséges feladatnak tűnhet. Feladatunk az, hogy minél többször rávilágítsunk arra, ennél jóval költségesebbek lehetnek a hibásan, és nem elég körültekintően meghozott döntések.

Hivatkozások[2]

Bollobás, B. [1985]: Random Graphs. Academic, London.

Erdős Péter–Rényi Alfréd [1959]: On random graphs. Publicationes Mathematicae. Debrecen.

Knuth, D. E. [1987]: A számítógép-programozás művészete 1. Alapvető algoritmusok és 2. Szeminumerikus algoritmusok. Műszaki Könyvkiadó, Budapest.

Watts, D. J. [1999]: Small Worlds: The Dynamics of Networks between Order and Randomness. Princeton University Press, New Jersey.

[2] A kutatás során nem használtunk fel külső irodalmi forrásokat, de a téma iránt érdeklődők számára a felsorolt publikációkat javasoljuk.