

Összetett hálózatok vizsgálata a Python IGraph moduljával, egy eKurzus tapasztalatai

Horváth Árpád

Óbudai Egyetem, Alba Regia Műszaki Kar, Székesfehérvár

Email: horvath.arpad@amk.uni-obuda.hu

Kulcsszavak: Összetett hálózatok, Python, eLearning

Kivonat—A szerző több éve foglalkozik az összetett hálózatok vizsgálatával foglalkozó kurzus fejlesztésével. Ebben a hálózatok vizsgálatának nem csak az elméletét tanulhatja meg a hallgató, hanem saját maga vizsgálhat hálózatokat a Python nyelv IGraph moduljával. Pár éve kifejlesztettünk egy olyan kurzust, amely kontakt óra nélkül elvégezhető és ezt kipróbáltuk a gyakorlatban. Az eddigi tapasztalatokról és a további tervekről szól a cikk.

Abstract—The author teaches about complex networks several years. In the course he developed the students can learn not only the theory of the networks, but he can investigate networks by the IGraph package of the Python programming language. Some years ago we developed a course that can be learned by the students without he needs to travel to the tutor, and we have tried it. This article is about our observations and goals.

1 BEVEZETÉS

Az összetett hálózatoknak olyan rendszert nevezünk, amelyek sok hasonló összetevőből állnak, és a köztük meglevő vagy hiányzó kapcsolatoknak van fontos szerepe a rendszer működése szempontjából. Az egyes összetevőket általában csúcsoknak hívjuk. Sok tudományterületen vizsgáltak összetett hálózatokat a múlt században is. A szociológia az emberi kapcsolatok hálózatát vizsgálta. A biológia és az orvostudomány többek között az idegrendszer és az agy felépítését, a fehérjekölcsönhatások hálózatát és a táplálkozási láncokat. Mindegyik tudományterületen a másik területek-

től függetlenül, és eltérő elnevezéseket használva történtek előrelépések. A századfordulón jött létre annak a technikai feltétele, hogy könnyedén feltérképezzünk nagy hálózatokat. A webes keresőrendszerek, a közösségi oldalak, a kereskedelmi oldalak ajánló rendszerei, a mobiltelefon-társaságok keresztül történt hívásokat rögzítő archívumok nagy méretű összetett hálózatokat rögzítettek.

Barabási Albert-Lászlónak és munkatársainak nagy felfedezése volt 1999-ben, hogy ezek a hálózatok sok szempontból hasonlóak: a legtöbb úgynevezett skálafüggetlen hálózatot alkot, azaz nincs egy jellemző értéke annak,

hogy egy csúcs mennyi másikhoz kapcsolódik. Ezt követően történt meg a hálózatokban gondolkodó tudományágak egymásra találása is. A nagy hálózatok tulajdonságait nyilvánvalóan nem lehet a hálózat ábrájára ránézve megállapítani, mint ahogy a kisebb hálózatok esetén működik. A hálózatok különböző szempontú jellemzésére különböző mérőszámok és függvények alkalmasak. A hálózatokat jól jellemző mennyiségek és függvények megtalálása, valamint a hálózatban történő átalakulások vizsgálata, valamint a hálózatok vezérlése a hálózattudomány feladata. Ezt a tudományt 2005-ben az Egyesült Államok *Nemzeti Kutatási Tanácsa* új, önálló tudományos diszciplínának ismerte el, amely ugyan kapcsolódik a biológiához, az orvostudományhoz, és a szociológiához és főként a fizika, valamint a matematika fogalomrendszerén alapul, mégis önálló fogalomrendszer és kérdésfeltevés jellemzi.

A hálózattudomány alkalmazásának már ma is jelentős társadalmi és tudományos hatásai vannak[1, 1.5. és 1.6. fejezet], és általában az adatok feldolgozása sok informatikai problémát vet fel. Mivel várható, hogy egyre több informatikusra lesz szükség ezen a téren, érdemes az informatikus-hallgatóknak a hálózatok elemzéséhez szükséges eszközök alapjait megismerni. Nyilvánvalóan egy kurzus keretében nem lehet cél az összes eszköz megismerése, hiszen az elterjedt progra-

mozási nyelvek mindegyikéhez vannak kisebb-nagyobb tudású, hálózatelemzésre alkalmas programkönyvtárak. A szerző azt az utat választotta, hogy egy ilyen programkönyvtárral ismerteti meg hallgatóit: az IGraph-fal.[2, 3] Ez a programkönyvtár C nyelven íródott, és a függvényei C, C++, Python és R nyelvből is meghívhatóak, ezért széles körben alkalmazható.

A hálózatok interaktív vizsgálatára, azaz arra az esetre, amikor még nem igazán tudjuk, hogy milyen jellemzőjű hálózatunk van, nagyon alkalmas a Python nyelvű és az R nyelvű parancssor. Ezek közül a Python nyelvet[4] választottam a hálózati vizsgálatok eszközeül, hogy ne csak általánosságban tudjak beszélni a hálózatok tulajdonságairól, hanem eszközt adhassak a hallgatók kezébe, amellyel saját maguk is megvizsgálhatnak egyes hálózatokat és meggyőződhetnek a hálózatokban tapasztalható meglepő hasonlóságokról, de a különbségekről is.

2 A KURZUS TÖRTÉNETE

Az Összetett hálózatok vizsgálata kurzus először a 2011/12-es tanév első félévében indult laborgyakorlatként. Ez a kurzus a mérnök informatikusok számára lett kidolgozva, akik túl vannak a matematika- és szoftverszigorlataikon, tehát már megismertek az objektumorientált programozási paradigmával és valamennyire a gráfelmélettel is.

1. táblázat. Az összetett hálózatok vizsgálata kurzusok összes és sikeres résztvevőinek a száma az Óbudai Egyetem NEPTUN-rendszere szerint (*Aki több félévben is jelentkezett, többször számít.)

sor- szám	félév	feliratkozott	elvégezte	
			fő	arány
1	2011/12/1.	11	11	100%
2	2011/12/2.	9	7	77%
3	2012/13/1.	15	15	100%
4	2013/14/1.	13	9	69%
5	2013/14/2.	13	12	92%
6	2014/15/1.	18	14	77%
7	2014/15/2.	13	10	76%
8	2015/16/1.	19	17	89%
9	2015/16/2.	12	9	75%
Összesen:		123*	104	85%

Időközben egy teljesen kontakt óra nélkül elsajátítható kurzus kidolgozását kérték tőlem, amelynek témájául ezt a kurzust választottam. 2014/15 első félévé óta tehát ez a kurzus eKurzusként üzemel, és legtöbb félévben a KMOOC-ra is felkerült, azaz a magyarországi, felvidéki, erdélyi és délvidéki egyetemek hallgatói is felvehették elvileg eKurzusként. 2016 tavaszán vegyes módon működött a kurzus. A KMOOC-ra nem került fel a kurzus, tehát csak a saját hallgatóink láthatták az Óbudai Egyetem hagyományos Moodle-rendszerében[5], de a kurzus tananyagának a feldolgozása továbbra is otthonról, önállóan történt. Volt viszont kijelölve négy egy tanórás alkalom, amikor a fontosabb modulok véget értek, ahol élő szóban beszéltük meg a modullal és a számonkéréssel

kapcsolatos fontosabb tudnivalókat.

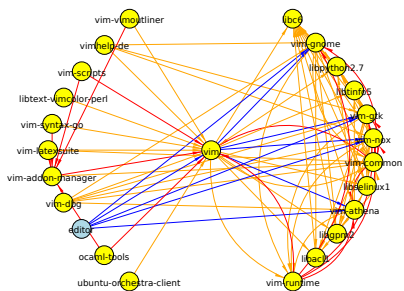
A tapasztalatok szerint egyetemünkről továbbra is főként a fehérvári műszaki informatikus hallgatók veszik fel a tárgyat. Az összes hallgató közül, akik felvették, ketten voltak villamosmérnökök, és egy mechatronika mérnöki szakos hallgató volt. A KMOOC-on persze többen is felvették, mint amit a NEPTUN feltüntet, de többnyire félbe is hagyták. Ennek az oka az lehet, hogy az érdeklődők többnyire nem tartották hasznosnak a területükön a hálózatos szemléletmódot, vagy nem ismerték az objektumorientált programozási szemléletet, amelynek hiányában nehezen tudták követni a tananyagot.

Látható az 1. táblázatból, hogy 104-en végezték el sikeresen a kurzust laborgyakorlaton vagy kontakt óra nél-

küli gyakorlaton. Összesen 123 jelentkezés történt, de ennél kevesebb hallgató jelentkezett a kurzusra, mert volt olyan, aki sikertelen félév után újra jelentkezett. Általában könnyen elválasztható volt az, aki kitartóan dolgozott a jobb jegyért, és az, aki csak feljelentkezett a kurzusra, és legfeljebb a lelegejéből csinált meg pár tesztet. Ritka volt az olyan eset, hogy valaki végigcsinálta, de nem volt meg az elég pontszáma a ketteshöz. Szabadon választható kurzus révén nem épül rá más tárgy, így akár későbbre halasztható, akár más szabadon választható tárggyal helyettesíthető ez a tárgy, ezért nem így sikerült mindenkinek.

3 A SZOFTVERCSOMAG-HÁLÓZAT

Az IGraph lehetővé teszi pár paranccsokkal, hogy egy internetes hálózatadattárazásban, a Nexusban[6] keresgéljünk, és onnan mások által feltérképezett hálózatot töltsünk le, és azt vizsgáljuk.

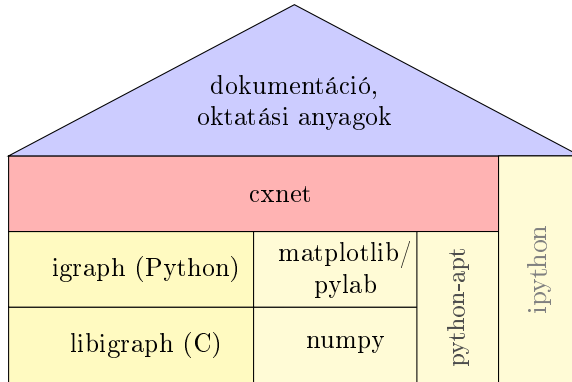


2. ábra. A szoftvercsomag-hálózat egy része, a Vim szövegszerkesztő szomszédjaival

Sokkal izgalmasabb viszont olyan hálózatot vizsgálni, amelynél a hálózatot ott helyben állítjuk elő.[7] A szoftvercsomag-hálózatot az általam fejlesztett CXNet programcsomag[8, 9] állítja elő egy korszerű gépen nagyszámrendileg tíz másodperc alatt (1. ábra). Ezt csak Linux alatt tudjuk előállítani, viszont így a hálózat arra a gépre jellemző, amin lefuttattuk. Más a hálózat a Linux különböző terjesztései esetén. Általában egy közel egyidőben kiadott Ubuntu több csomagot tartalmaz, mint egy Debian. A csomagok és kapcsolatok száma attól is függ, hogy a gépünkön mikor frissítettük, a fájlokat, amelyek a tároló csomagjainak adatait és függőségeit (valamint más függőségnél enyhébb kapcsolatait) tartalmazzák. Általában az idő haladtával új csomagok jelennek meg, vagy egyeseket szétbontanak több csomagra, vagy ritkán némelyik feleslegessé válik, mert van helyette jobb, és törlik a tárolóból.

A szoftvercsomagok esetén megvizsgálhatjuk a CXNet-tel, hogy melyikektől függ, melyikek függenek tőle, sőt erről egy áttekinthető ábrát is rajzoltathatunk (2. ábra).

A CXNet által létrehozott hálózat tárol a teljes hálózattal kapcsolatos tulajdonságokat, mint például a csomagokkal kapcsolatos adatok frissítési dátuma, a csúcsokkal (csomagokkal) kapcsolatos adatokat, mint amilyen a fájl méret, vagy a csomag fontossága, valamint az élekkel (függőségek...) kap-



1. ábra. A cxnet programcsomag alapjául szolgáló programcsomagok

csolatos adatokat, mint például a típusa (depends, suggests...).

Ez a hálózat irányított (ha az egyik függ a másiktól, általában a másik nem függ az egyiktől), és kiválóan alkalmas arra, hogy a szakirodalomban vagy a Nexusban található hálózatokéval összehasonlítsuk a jellemzőit.

4 VIDEÓKKAL SEGÍTETT TANÍTÁS A PYEDUVAL

Mióta eKurzusként működik az Összetett hálózatok vizsgálata tárgy, azóta tartoznak hozzá tájékoztató videók, amelyek az első komolyabb programozási feladaton vezetnek végig, illetve a szükséges szoftverek telepítését és használatát ismerteti meg.

A Moodle rendszerben a videók közötti navigáció elég nehézkes: egyesével meg kell nyitni azokat, és megnézni. Különösen nehéz, ha az egyes videók után kis kérdésekkel ellenőrizni szeretnénk a tananyag megértését, pe-

dig ez a figyelem fenntartása, és az érthetőség ellenőrzése szempontjából fontos.

A videók és tesztkérdések közötti könnyű navigáció problémájának a megoldására kezdtük el fejleszteni a pyEdu oktatási oldalt. Mintaképpül a Udacity[10] szolgált. A Udacity kurzusaiban kis lépésekben, rövid videókkal mutatják be az új tananyagot; legtöbbször új anyagot bemutató videó a kérdéssel zárul, amellyel teszteli a hallgatót, hogy megértette-e a tananyagot. A hallgató válasza után pedig egy újabb rövid videó összefoglalja a megoldást, vagy *egy lehetséges* megoldást.

A pyEdu hasonló felépítésű. Ebben is nagyon könnyen navigálhatunk a videók és a kérdések között. A jelenlegi állapotban a kurzus első fele döntően a Moodle rendszerben található, a másik része pedig a Moodle-ből elérhető pyEdu oktatási weboldalon lesz elérhető szeptembertől.

5 PROGRAMOZÁSI ÉS EGYÉB FELADATOK A PYEDUVAL

A pyEdu esetén az egyes feladatok jelenleg két csoportba tartoznak. Az egyikbe az egyszeres és többszörös választásos tesztek, ahol a kérdésre egy listából kell egyetlen, vagy tetszőleges számú válaszlehetőséget kiválasztani.

Ennél érdekesebb az a feladattípus, melynek során a hallgatónak egy programozási feladatot kell megoldania. Amennyiben a program értékelését automatizálni szeretném, nyilván nem működnek a szövegfeldolgozó módszerek, hiszen egy függvényben a kurzus résztvevője számtalan módon nevezheti el a változókat, egy mennyiséget egy vagy több lépésben is kiszámíthat, esetleg segédfüggvényt is alkalmazhat. A programozásifeladatmegoldó feladattípus úgy működik a pyEduban, hogy a megírandó függvény neve, és az egyes paraméterek sorrendje pontosan meghatározott, valamint a függvény visszatérési értéke, és az is, hogy hibásan megadott paraméterek esetén milyen kivétel lépjen fel.

A program kódját a pyEdu a kurzus kidolgozója által megadott egységteszttekkel teszteli. Ez alkalmat ad arra, hogy tesztelhessük a bemenő paraméterek egy halmazára, hogy megfelelő értékekkel tér-e vissza a függvény, valamint arra is, ha hibás (nem megfelelő típusú vagy az értelmes intervallumon kívül eső) argumentumokkal hívjuk meg a függvényt, akkor a megfelelő kivételek lépnek-e fel. Az egységtesztben esetleg megírhatja a tananyagfej-

lesztő magát a tesztelni szándékozott függvény egy változatát is, ilyen esetben egy ciklussal lehetősége van rá, hogy a paraméterek nagy halmazán végighaladva ellenőrizze, hogy a hallgató által megírt függvény ugyanazokat az értékeket adja-e vissza, mint a tananyagfejlesztő által megírt függvény.

6 A KURZUS FELÉPÍTÉSE

Az Összetett hálózatok vizsgálata kurzus ószre tervezett témái a következők:

1. A Python-nyelv, verziói, alapértelmezett shell és ipython, range függvény, szeletek, for ciklus, szöveges fájlok kezelése.

Python-szkriptek írása különböző szövegszerkesztőkkel és integrált fejlesztői környezetekkel.

2. Python: logikai hamis értékek, listával dolgozó függvények (all, any, max, min, sum), függvénydefiniálás, programok vizsgálata egyszerűbb tesztekkel.

3. A hálózatok vizsgálata és kirajzolása a Python igraph moduljával. Gráfelméleti alapfogalmak ismétlése (egyszerű gráf, teljes gráf fokszáma, komponens, fa élszáma, kapcsolat az átlagfokszám, a csúcsszám és élszám között).

Megjegyzés: A diákok a harmadik-negyedik héten egy-egy beadandó feladatot kapnak.

Például meg kell tudni határozatni egy hálózatban a komponensek számát, létrehozni szöveges fájlban tárolt éllistából hálózatot.

4. A hálózatok kirajzolásának finomhangolása, csúcs- és élattribútumok, fokszám, szomszédok, adott tulajdonságú csúcsok és élek keresése

Osztályok létrehozása a Pythonban, tesztek létrehozása a unittest modulal

Megjegyzés: Az IGraphban a hálózatok kirajzolásának tulajdonságait (szín, méret) él- és csúcattribútumokkal szabályozhatjuk, de ezek szolgálhatnak a csúcsok és élek adatainak, élek súlyának tárolására is.

5. Az Internet, mint hálózat vizsgálata (csúcs- és élszám, átlagfokszám, maximális és minimális fokszám, átmérő)

Függvényábrázolás Pylabbal. `linspace`, `plot`, `semilog(y|x)`, `loglog`, `legend`, `title`, `xlabel`, `ylabel`, `grid`, TeX-es feliratok. Grafikonok mentése.

Megjegyzések: A pylab függvényei nagyjából megegyeznek a MATLAB függvényeivel. Az itt tanultakkal együtt tanuljuk meg azt is, hogy hogyan érdemes a függvényeket ábrázolni, hogy észrevegyük azok hatványfüggvény, illetve exponenciális jellegét.

Ezen a héten már elkezdődik a beadandó feladatok ellenőrzése is. Amennyiben a kurzus létszáma megengedi, a hallgatók minden egyes feladat megoldását és tapasztalatait röviden bemutatják az órán.

6. Az Erdős–Rényi modell (véletlen gráf), a véletlen gráfok komponenseinek vizsgálata. Adott hálózatával egyező csúcyszámú és élszámú véletlen hálózat generálása.
7. A Price-modell és a Barabási-Albert modell. Az összetett hálózatok fokszámeloszlása és összegzett fokszámeloszlása. A fokszámeloszlás létrehozása és ábrázolása.
8. A szoftvercsomag-hálózat vizsgálata. Összehasonlítása az eddigi modellekkel. Az összegzett fokszámeloszlás. Skálafüggetlen hálózatok.
9. Gráfelméleti adatstruktúrák (szomszédossági és illeszkedési mátrix, szomszédossági tömb, láncolt szomszédossági lista). Hálózatok tárolása fájlokban (éllista, graphml, gml, pickle, dot, gml).
10. A fokszámeloszlás simításának módszerei: a beosztás megváltoztatása (bin smearing). A kitevő becslése.
11. A csoporterősségi együttható, annak fokszámfüggése. A hierarchikus modell.

12. A hálózatok sérülékenysége: véletlen hibák és célzott támadás.

Ezeket a témákat a kurzus 6 modulra osztva tartalmazza.

A 6 modul mindegyikének végén van olyan teszt, amelyet a hallgatónak meg kell oldania egy előre meghatározott pár napos időintervallumban. Ezek a tesztek a Moodle rendszerben találhatóak. Az első két kurzustól eltekintve beadandó feladatok is tartoznak a kurzushoz. Ezek olyan programozási feladatok, ahol a végeredményt fel kell tölteniük a Moodle-rendszerbe.

A tanítást segíteni fogja, hogy az összetett hálózatokról kiváló, átfogó tankönyv jelent meg idén magyarul Barabási Albert-László tollából *A hálózatok tudománya* címmel, ami az oktatásához sok segítséget nyújt az oktatóknak és a hallgatóknak.[1]

7 AZ EKURZUS ELŐNYEI ÉS NEHÉZSÉGEI

Az alábbiakban összefoglalom, milyen előnyei vannak az eKurzusnak, és milyen nehézségeket támaszt egy kontakt óra nélküli eKurzus a laborgyakorlatokhoz képest.

A legnagyobb előny természetesen az, hogy a hallgató otthon, a megszo- kott számítógépen dolgozhat a saját maga által kialakított időbeosztás szerint. A modul végén található számonkéréseket is egy adott időtartományon belül a neki legmegfelelőbb időpontban kezdheti el. (Természetesen a saját idővel gazdálkodás okozhat nehézséget is.)

A hátrányok közé tartozik, hogy nehéz olyan feladatokat készíteni, hogy gyakorolni is lehessen vele, mégis a másik hallgatókétól és a gyakorló feladattól eltérő feladatot kapjanak a modulzáró teszteken. Ebben segíthet a következő lehetőség. A Moodle-teszteket lehet egy viszonylag egyszerű szöveges fájlformátumból, az úgynevezett GIFT-formátumból importálni. Ez jóval egyszerűbb a Moodle XML-en alapuló formátumánál. Amennyiben a teszt olyan, hogy könnyen lehet egy programmal generáltatni sok változatot, akkor az oktatónak egyszerű dolga van: ír egy olyan programot, amely létrehozza a feladat sok változatát GIFT-formátumban, és azt importálja a Moodle-kurzusba.

Ha sok változata van a feladatnak, akkor nem lehet, vagy legalábbis nem érdemes megjegyezni az egyes változatok megoldásait, mert kisebb erőfeszítés megérteni, hogyan lehet megoldani a feladatot.

Ezt a szoftvercsomag-hálózattal kapcsolatos feladathoz alkalmaztam. Ezen esetben egy adott lementett hálózatot kell betölteni, és olyan csúcspontokat vagy éleket kell tudni megszámo- lítani, ahol valamelyik tulajdonság adott értékű, illetve, ahol számszerű érték van, adott értéknél nagyobb, vagy kisebb. Például a csúcspontok (csomagok) típusa lehet, valódi, vagy virtuális, a programcsomag bájtban mért méreténél pedig meg lehet kerestetni az adott értéknél nagyobbakat, kisebbeket. Megfelelő ciklussal végigmenve valamilyen alsó vagy felső korlát értékeken rengeteg feladatot lehet csinál-

ni, ahol szűrni kell valamilyen jellemzőre. Ilyenkor nem gond, ha a gyakorló feladatoknál is véletlenszerűen választunk, és a tesztnél is, hiszen kicsi az esélye, hogy pont ugyanazt kapja a hallgató.

A beadandó feladatok esetén elég nehéz helyzetben van az oktató, ugyanis elég nehéz annyiféle feladatot kiadni a hallgatóknak, ahányan vannak úgy, hogy a nehézségi szintjük is hasonló legyen, és mindegyik érdekes legyen. Ráadásul ezeket évről évre változtatni kellene. Megoldás lehet projektmunka kiadása hallgatók egy csoportjának, ami azonban olyan hallgatók esetén, akik a Kárpát-medence minden tájáról vannak, kihívást jelenthet, bár nem megoldhatatlant.

A kontakt órák nélküli kurzusok esetén nehezen lehet ellenőrizni, hogy a hallgató saját maga végezte el a munkát. A teszt esetén, ha odaülte-ti egy hallgatótársát, az nem derül ki. A beadandó feladatnál természetesen a teljesen egyező megoldások jelezhetik a másolást.

8 CÉLOK 2016 ŐSZI FÉLÉVÉRE

Az Összetett hálózatok vizsgálata kurzust ősszel újból a KMOOC keretében kerül meghirdetésre, tehát sok hallgatónak lesz lehetősége felvenni a tárgyat. A következő félévre ezért az a cél, hogy a tárgy segédanyagai segítséget nyújtsanak azoknak is, akik nem informatikai képzést kaptak. Ezt kiegészítő anyagok elkészítésével szándékozunk elérni. További cél a kurzus célközösségének megtalálása, és az,

hogy azok, akiknek ez a téma hasznos lehet, minél többen tudjanak a kurzusról, és annak lényegéről. Ezt szolgálja az idén lefogatott kurzusismertető videó.

Eddig a pyEdu oktató oldal nem került felhasználásra. A pyEdu használata két célt is meghatároz. Egyfelől egy tananyagfejlesztési célt, hogy a segédanyagok minél nagyobb része legyen lefedve magyarázó videókkal, és hozzájuk kapcsolódó automatikusan kiértékelődő tesztekkel. A következő félévre a tananyag végét, az utolsó két modult szándékozunk így átdolgozni. A másik cél programfejlesztési cél: a pyEdu átláthatóvá, könnyen kezelhetővé és elegánsá tételé.

9 ÖSSZEFOGLALÁS ÉS KITEKINTÉS

Az Összetett hálózatok vizsgálata kurzus a hálózattudomány elméleti alapjai mellett a gyakorlati alkalmazásra is komoly hangsúlyt fektet, megismerteti a hallgatókat az IGraph programcsomaggal, amely önmagában elérhetővé tesz számos hálózatelemzési módszert, és lehetőséget nyújt saját hálózatelemzési módszerek megvalósítására, vagy egy összetett vizsgálat összeállítására.

A kurzus tananyagok és a feladattípusok széles tárházát alkalmazza azért, hogy a hallgató könnyen elmélyedhessen a tananyagban.

A kontakt óra nélküli kurzusok esetén nehéz ellenőrizni, hogy a hallgató önálló munkája-e, amit csinált. Olyan esetekben viszont, ahol nincs jelentősége a megszerzett pontoknak, és feltelezhetjük, hogy a hallgató a tudás,

és nem a jegy megszerzéséért dolgozik, ez a képzési forma nagyon hatékony lehet.

HIVATKOZÁSOK

[1] B. Albert-László, *A hálózatok tudománya*. Libri Könyvkiadó, Budapest, 2016.

[2] G. Csárdi and T. Nepusz, „The igraph software package for complex network research,” *InterJournal Complex Systems*, p. 1695, 2006.

[3] —, „Igraph,” 2003–. [Online]. Available: <http://igraph.sourceforge.net/>

[4] „Official Python site.” [Online]. Available: <http://www.python.org>

[5] „The official site of Moodle.” [Online]. Available: <https://moodle.org>

[6] „A Nexus weboldala.” [Online]. Available: <http://nexus.igraph.org/>

[7] Á. Horváth, „The software package dependency networks of some linux distributions,” in *4th IEEE International Conference on Nonlinear Science and Complexity: NSC 2012.*, I. J. Rudas, Ed., Budapest, Hungary, 2012.

[8] „A CXNet weboldala.” [Online]. Available: http://django.arek.uni-obuda.hu/cxnet/doc/tutorial_en.html

[9] Á. Horváth, „The cxnet complex network analyzer software,” *Acta Polytechnica Hungarica*, vol. 10, no. 6, 2013.

[10] „A Udacity weboldala.” [Online]. Available: <https://www.udacity.com/>