Szegedi Tudományegyetem

Informatikai Intézet

Diplomamunka

Vass Máté

2023

Szegedi Tudományegyetem
Informatikai Intézet

Epidemiológiai kontroll és a fertőzés monitorozási probléma

Diplomamunka

Készítette:

Vass Máté
programtervező informatikus
MSc szakos hallgató

Témavezetők:

Krész Miklós
Szegedi Tudományegyetem
főiskolai tanár

Hajdu László
InnoRenew CoE & University of Primorska, Szlovénia
tudományos munkatárs

Dávid Balázs
InnoRenew CoE & University of Primorska, Szlovénia
tudományos munkatárs

Konzulens:

London András István
Szegedi Tudományegyetem
adjunktus

Szeged
2023

Feladatkiírás

A hálózatokon értelmezett diffúziós folyamatok számos jelenség modellezésére alkalmasak: járványok terjedése, véleményformálók detektálása szociális hálózatokban, valamint csődök kialakulása pénzügyi hálózatokban. A folyamatok kontrollja intervenciós stratégiákkal ezen alkalmazások mindegyikében releváns, aminek tipikus esete az epidemiológiában a vakcinázás. A diplomamunka témája a vakcinázás példája nyomán a diffúziós folyamatok hatékony blokkolása hálózatokban. A dolgozat keretében a szakirodalom tanulmányozása alapján ki kell fejleszteni legalább egy "state-of-the-art" eljárást, valamint ki kell dolgozni egy új algoritmust, amely az úgynevezett "Fertőzés monitorozási probléma" (L. Hajdu, M. Krész: The influence monitoring problem, Proceedings of SOR 2023) megoldásával ad közelítő megoldást. Az eljásásokat részletes tesztelés keretében össze kell hasonlítani és elemezni.

Tartalmi összefoglaló

A téma megnevezése:

Hatékony fertőzés blokkolási algoritmus kifejlesztése egyszerű, irányított, élsúlyozott gráfokra.

A megadott feladat megfogalmazása:

Napjainkban egyre nagyobb figyelem hárul valóélet-beli folyamatok hálózatokkal való modellezésére. Epidemiológiai szempontból különféle fertőzésterjedési folyamatokat vizsgálhatunk, két ismert probléma a fertőzés blokkolás, valamint a fertőzés monitorozás. A feladat ezek megismerése és kapcsolatuknak vizsgálata volt egy fertőzés blokkolási algoritmus kifejlesztésén keresztül.

A megoldási mód:

A szakirodalom fertőzés blokkolással, valamint fertőzés monitorozással foglalkozó területének alapvető gondolatait felhasználva egy költség megszorításos maximális folyam modell implementálása a fertőzés monitorozási problémára. Az algoritmus kipróbálása a fertőzés blokkolási problémára, valamint ennek összehasonlítása a tudományterület más módszereivel hatékonyság szempontjából.

Alkalmazott eszközök, módszerek:

A diplomamunkában megalkotott algoritmus hatékonyságát kis méretű generált gráfokon tesztelték. Az implementáció, automatizálás, fájlkezelés és szöveges fájlok formázása Python programozási nyelven történt.

Elért eredmények:

A tesztelési módszertanunk alapján az eredményeink alátámasztják, hogy a szakirodalomban fellelhető fertőzés blokkolási algoritmusok hatékonyabbá tehetőek új megközelítésekkel. Erre egy hatékony megoldás a fertőzés monitorozási probléma alapú módszerünk.

Kulcsszavak:

Epidemiológia, fertőzésterjedési modell, független kaszkád modell, fertőzés blokkolás, fertőzés monitorozási probléma, költség megszorításos maximális folyam modell

Tartalomjegyzék

Absztrakt

A hálózatokon terjedő diffúziós folyamatok széleskörű alkalmazást nyertek az elmúlt években; többek között a klasszikusnak tekinthető szociológiai vagy epidemiológiai területek mellett a közgazdaságtan, a kommunikációs hálózatok és a szemantikai hálózatok vizsgálatában is fontos szerepet játszanak. A témakör fejlődésében döntő szerepet játszott Kempe, Kleinberg és Tardos 2003-ban publikált tanulmánya [1], melyben adott diffúziós folyamat szerinti legbefolyásosabb csúcsok megtalálása egy diszkrét optimalizálási feladatként került definiálásra. Ezen eredeti feladat esetében adott k szám, valamint az éleken definiált valószínűségek és a csúcsokon értelmezett „fertőzési függvények" által meghatározott diffúziós folyamat. A legbefolyásosabb k csúcs alatt azon csúcsokat értjük, melyekből a diffúziós folyamatot elindítva várható értékben a legnagyobb számú csúcs fertőződik meg. Ezen munkában igazolást nyert, hogy a probléma NP-teljes, azonban a diffúziós folyamatok egy széles osztálya esetén a mohó eljárás garantált approximációt szolgáltat az optimum vonatkozásában.

Kempe és társainak klasszikus módszere számtalan kutatási eredményt indukált. Az utóbbi időben több megközelítés azt a kérdést vizsgálta, hogy egy ténylegesen végbemenő diffúziós folyamatot miként lehet egy adott méretű (vagy általánosabban értelemben egy adott költségű) halmaz segítségével maximális mértékben „blokkolni". Az általunk vizsgált változat Sambaturu et al. EpiControl modellje [2] alapján azt keresi, hogy melyik k csúcs hálózatból történő eltávolítása csökkenti a legnagyobb mértékben várható értékben a fertőzött csúcsok számát. A fenti modellre Sambaturu és társai egy hatékony randomizált kerekítési eljárást fejlesztettek ki.

Ezen dolgozat keretében a Hajdu és Krész által 2021-ben bevezetett fertőzés monitorozási probléma [3] segítségével egy hatékony közelítő eljárást ismertetünk, amelynek során azon k csúcs kiválasztása a cél, amelyen várható értékben a legtöbb fertőzési lánc halad keresztül. A kidolgozott eljárás egy sztochasztikus programozás szerinti hálózati folyam modellen alapul. A módszer hatékonysága teszteseteken keresztül nyer igazolást.

Bevezetés

Az elmúlt évek rohamos technológiai fejlődése, valamint a végeláthatatlan mennyiségben, nyilvánosan elérhető adathalmazok újabb és újabb lehetőségeket nyitottak különféle valóélet-beli folyamatok számítógépes modellezésére és szimulációjára. A napjainkban igen népszerű kutatási területnek tekinthető gráfelmélet klasszikus problémái is ebbe a kérdéskörbe tartoznak. Emellett számos egyéb feladat gyakran átfogalmazható megfelelő gráf reprezentáció választásával könnyen megoldható, kis komplexitású problémára. Így sokszor inkább előbbi feladat bizonyul nehezebbnek, az átalakítás után a megoldási módszer már adott.

Egy hagyományosan gráfokkal reprezentált probléma hálózatokon információ, befolyás vagy akár fertőzés terjedése. A tudományterület egyik megalapozó cikkében [1] különféle befolyásterjedési modelleket vizsgáltak a szerzők, legfőbb hangsúlyt a széles körökben elterjedt független kaszkád, valamint lineáris küszöb modellekre és azok általánosításaira fektetve. Ezen modellek a szakirodalomban korábban is felbukkantak, előbbi szerepéről 2001-ben [4], utóbbi jelentőségéről pedig már egészen 1978-ban [5] is értekeztek a terület kutatói.

A befolyás terjedésének témakörében egy gyakran vizsgált probléma a fertőzés maximalizálás, ahol cél egy adott hálózaton minél nagyobb várható értékű fertőzöttségi érték elérése bizonyos megszorításokkal. Az [1] cikk egyik fő témája is ez volt, ahol a szerzők részletesen körbejárták a kérdéskört, ennek bonyolultságáról és egy mohó heurisztikával való közelítéséről is hosszasan értekeztek.

Ezzel teljesen ellentétes feladat a diffúziós folyamatok minél hatékonyabb blokkolása, vagyis az előbbiekben említett fertőzés minimalizálása, esetleg megelőző szándékkal ennek monitorozása. Ezen problémák epidemiológiai szempontból a vakcinázással, valamint a teszteléssel vonhatók párhuzamba. Habár a két megközelítés látszólag távol áll egymástól, sejtésünk szerint bizonyos szempontból mégis találhatunk összefüggéseket, hasonlóságokat. Alapötletünk is innen ered, a dolgozat fő részében egy fertőzés monitorozási problémára megalkotott modellt mutatunk be, melynek hatékonyságát egy speciális fertőzés minimalizálási feladaton teszteljük. A dolgozatban célunk a vakcinázás által motivált epidemiológiai kontroll probléma optimalizálása volt, azaz meghatározott számú egyed eltávolítása a hálózatból a fertőzési szint maximális csökkentése érdekében, amit a [2] cikk alapján tárgyalunk. Az újszerű megközelítésünk szerint viszont az optimális tesztelés (fertőzési láncok korai felismerése) által motivált fertőzés monitorozási probléma [3] segítségével dolgozunk ki közelítő eljárást. A dolgozatban

az [1]-ben ismertetett megoldásnál hatékonyabb módszert szerettünk volna kidolgozni, ami eredményekben is versenyképes.

Fontos kiemelnünk, hogy a befolyás terjedésének fertőzés minimalizálási szempontból való vizsgálata mindössze egy kiragadott felhasználási terület szemléltetés céljából. Általánosságban a probléma lényege, hogy valamilyen hálózat ellenálló-képességét szeretnénk növelni korlátolt költségű beavatkozással. Ezt számos más kontextusban is értelmezhetjük, gondolhatunk akár ökológiai, gazdasági vagy bűnügyi hálózatokra is. A különféle beavatkozási módok és ezek költsége esetenként más és más lehet, de formálisan a feladat nagyon hasonló.

Célunk jelen kutatásban nem lesz más, mint a fertőzés blokkolási probléma hatékony approximációja egy fertőzés monitorozási feladaton keresztül bevezetett költség megszorításos maximális folyam modell megoldásával. A dolgozat fő eredménye ezen modell és ennek egy Python programozási környezetben megvalósított implementációja. Az így elkészült algoritmusunk hatékonysága kis méretű, mesterségesen generált gráfokon való teszteléssel nyert igazolást „*proof of concept*” jelleggel.

A diplomamunkában az alapfogalmak átfogó ismertetésével kezdünk, majd definiálásra kerül a fentebb említett két probléma, az *EpiControl* és a fertőzés monitorozási probléma. Ezek megoldására konkrét modelleket és ezek implementációit is bemutatjuk, az elkészült programkódok az elektronikus mellékletben megtalálhatóak. A kiértékeléshez ezen algoritmusok hatékonyságát néhány egyszerű baseline modellel vetjük össze. Természetesen ezek is megtekinthetőek az elektronikus mellékletben. A munka összegzéseképp pedig egyéb továbbfejlesztési lehetőségeket, valamint jövőbeli kutatási irányvonalakat, célokat prezentálunk.

Jelen diplomamunka alapját a Szegedi Tudományegyetem 2023. évi őszi Tudományos Diákköri Konferencia Informatikai Szekciójában bemutatott „Diffúziós folyamatok blokkolása és hálózati folyammodellek” című dolgozatom [6] képezte, melyet szintén jelenlegi témavezetőimmel készítettem.

1. Alapfogalmak

A dolgozat első fejezetében a témához kapcsolódó definíciókat, tételeket, valamint ezek szakirodalmi vonatkozásait tekintjük át. Definiáljuk a legalapvetőbb gráfelméleti alapfogalmakat, kezdve a gráfok és folyamok formális leírásával. A hálózatokon való diffúziós folyamatokat alapvető fertőzésterjedési modelleken keresztül vezetjük be, majd áttérünk a befolyásmaximalizálás problémakörére és ennek szimulációjára, végül pedig érintőlegesen szót ejtünk a lineáris programozásról. Minden tudományterületnek megemlítjük a fontosabb szakirodalmi vonatkozásait, továbbá valóélet-beli alkalmazásokat is prezentálunk. A fejezetben található fogalmak, definíciók ismerete a teljes munka értelmezéséhez elengedhetetlen, de nem teljes körű, részletesebb leírás a [7] könyvben található.

1.1. Gráfok és folyamok

A diffúziós folyamatok és azok hatékony blokkolásának vizsgálata során hálózatokkal fogunk foglalkozni, melyben a különböző entitásokat és a közöttük fennálló kapcsolatrendszert gráfokkal reprezentáljuk. Definiáljuk a $G = (V, E)$ gráfot egy halmazpárral, ahol V a csúcsok, E pedig az élek halmaza. A dolgozatban egyszerű gráfokkal fogunk dolgozni, melyekben nincsenek sem önmagukba vezető hurokélek, sem pedig többszörös élek. A hálózat éleinek irányítása szempontjából megkülönböztethetünk irányított, illetve irányítatlan gráfokat. A következőkben részletesen bemutatott diffúziós folyamatokban, valamint a hálózati folyamok vonatkozásában is az irányított esettel fogunk foglalkozni. Az irányítatlan gráfok tekinthetők ilyen szempontból speciális irányított gráfoknak, ahol mindkét irányban találhatóak élek a csúcsok között. Fontos még továbbá, hogy $\forall v \in V$ csúcs, valamint $\forall e \in E$ él súlyozott, ezen w_v és w_e értékek a $[0,1]$ intervallumból kerülnek ki. Ezen súlyok a diffúzió szempontjából az éleken terjedési valószínűségeket fognak jelenteni, a csúcsok esetén pedig azt, hogy adott csúcs mekkora valószínűséggel indíthat el diffúziót (azaz lehet kezdeti fertőzött) a gráfunkban. Hálózati folyamok tekintetében csúcsvalószínűségekről nem fogunk beszélni, az élek súlyairól viszont igen, ezek kapacitásokat jelölnek majd a későbbiekben.

A hálózati folyamok definiálásához vegyük a $G = (V, E)$ egyszerű, irányított gráfot. Két kitüntetett szerepű csúcsot különböztetünk meg, ezek az s (~ source) *forrás* és t (~ target) *cél* csúcsok. Fontos kikötések, hogy $s, t \in V$, valamint $s \neq t$. Az éleink súlyozottak, melyek ebben az esetben kapacitásokat jelölnek. Ez formálisan csak azt jelenti, hogy a hálózaton adott egy

$c: E \to \mathbb{R} \geq 0$ kapacitásfüggvény, mely megadja, hogy az egyes éleken legfeljebb mekkora lehet az áramlás. Az éleken ténylegesen felvett $f: E \to \mathbb{R}$ értékeket figyelembe véve *folyamról* beszélhetünk, amennyiben az alábbi feltételek teljesülnek:

- kapacitási feltételek: $\forall e \in E: 0 \leq f(e) \leq c(e)$, vagyis az éleken folyó anyag mennyisége nem lépheti át a kapacitást
- megmaradási feltételek: $\forall v \in V - \{s, t\} : \sum_{e \in E^{ki}(v)} f(e) = \sum_{e \in E^{be}(v)} f(e)$, azaz a forrás és nyelő csúcson kívüli csúcsokban nem keletkezhet vagy veszhet el anyag

A fentieken kívül definiálnunk kell még a folyam értékét, mely a következőképpen néz ki formálisan: $é(f) = \sum_{e \in E^{ki}(s)} f(e) - \sum_{e \in E^{be}(s)} f(e)$. A folyamok esetében a célunk a folyam értékének maximalizálása, azaz az előbb definiált $é(f)$-re cél-függvényként is tekinthetünk.

1.2. Diffúzió hálózatokon

A hálózatokon végbemenő diffúziós folyamatokat diffúziós modellek segítségével írhatjuk le. Mivel jelen dolgozatban fertőzésterjedéssel foglalkozunk, ezért a terminológiánkban diffúzió alatt mindig fertőzés terjedését fogjuk érteni. Természetesen léteznek általánosabb felhasználású modellek is, mi azonban olyanokra fókuszálunk, melyek hatékonysága fertőzésterjedés vizsgálati szempontból korábban már bizonyítást nyert a szakirodalomban.

A diffúziós modellek bemenete jelen esetünkben egy $G = (V, E)$ egyszerű, irányított gráf, melynek w_v csúcs- és w_e élsúlyai a $[0,1]$ intervallumból kerülnek ki. A modellben a csúcsokon lévő súlyok alapján meghatározhatjuk az A_0 kezdeti fertőzött csúcshalmazt. Amint ezzel végeztünk, iterációnként haladunk, és a modellben definiált szabályok alapján terjesztjük tovább a fertőzést a hálózaton. A modell kimenete pedig nem más, mint az utolsó, t. iteráció után kapott A_t fertőzött állapotban lévő csúcsok halmaza.

1.2.1. Független kaszkád modell

A fertőzések terjedését viszonylag pontosan szemléltető, egyszerű, széleskörben alkalmazható modell a független kaszkád modell. Hatékonysága az epidemiológiai felhasználások [1] mellett közgazdaságtani alkalmazásokban [4, 8] is bizonyítást nyert. Ahogy azt az előző bekezdésben már említettük, a diffúziós modellünk bemenete a $G = (V, E)$ egyszerű, irányított gráf lesz, adott w_v csúcs- és w_e élsúlyokkal, melyek a $[0,1]$ intervallumba esnek. Az A_0 kezdeti fertőzött halmazt megadhatjuk direkt módon egy konkrét csúcshalmazként, vagy akár a csúcsokon lévő súlyokat felhasználva is kiszámolhatjuk (lásd később a modell általánosításánál). Az élsúlyok a független kaszkád modellben terjedési valószínűségeket reprezentálnak. Az A_0 kezdeti fer-

tőzött csúcshalmaz elemeiből kiindulva minden élen a rajta lévő valószínűséggel fog egymástól függetlenül terjedni a fertőzés. Az újonnan megfertőződött csúcsok a diffúzió végéig fertőzöttek maradnak és ezek természetesen fertőzhetnek tovább. Amikor már egyik csúcs sem tud további szomszédokat megfertőzni, a diffúziónak vége, a modell kimenete pedig az összes fertőzött csúcsot tartalmazó A_t halmaz lesz.

A független kaszkád modell pszeudokódja a következőképpen néz ki:

$INPUT: G(V, E), A_0$

$WHILE\ A_i \neq 0$:

$A_i \leftarrow$ $újonnan\ megfertőződött\ csúcsok$

$\forall v \in A_i\ megfertőzi\ a\ nem\ megfertőzött\ szomszédait\ w_e\ valószínűségekkel$

$IF\ sikeres\ a\ fertőzés\ adott\ u\ szomszédra$:

$A_{i+1} = A_i \cup u$

$END\ IF$

$END\ WHILE$

$A_t = \bigcup_i A_i$

$OUTPUT: A_t$

1.1. pszeudokód: Független kaszkád modell

A modellnek a szakirodalomban létezik általánosított változata is, melyre általánosított független kaszkád modellként szoktak hivatkozni [8]. A bemutatásra került független kaszkád modellnél feltételeztük az A_0 halmaz ismeretét. Azonban ez a való életben a legtöbb esetben nem adható meg egzakt módon. Az általánosított modellben a kezdeti fertőzött csúcsok megadása nem konkrét megadással történik, hanem $\forall w_v$ csúcssúly p_v a priori fertőzési értékként jelenik meg. Ezek alapján a modell $\forall v \in V$ csúcsra kiszámolja a p'_v végső a posteriori fertőzési értékeket. Az egész úgy is értelmezhető, hogy a $G = (V, E)$ gráf valószínűségi eloszlásait alakítjuk át végső fertőzési értékekké.

A fertőzési folyamat természetesen továbbra is a függet-len kaszkád modellel írható le. Minden v csúcson egymástól függetlenül adott egy kezdeti p_v a priori fertőzés. Egy fertőzési

folyamat ezért nem egy konkrét S_0 halmazból indul, hanem minden egyes csúcs az a priori fertőzési értékének megfelelően egymástól függetlenül megfertőződik, vagy nem. Így az adott fertőzés ezen kezdeti véletlen fertőzés alapján megfertőzött csúcsokból indul (ezek a csúcsok alkotják a véletlen S_0 halmazat), majd a folyamat a független kaszkád modell szerint megy végbe. Ennek alapján az összes lehetséges a priori értékek által előállítható S_0 halmazokra egy eloszlás adódik. Ezen eloszlás szerint a $\sigma(S_0)$-ban való előfordulásuk szerint az egyes v csúcsok esetében egy p'_v a posteriori fertőzés számolható. Így a gráf $\sigma(G)$ fertőzési értéke az összes csúcsán kapott a posteriori fertőzési értékek összegeként áll elő. Ilyen módon könnyen látható, hogy az általánosított független kaszkád modell az eredeti független kaszkád modellnek valóban az általánosítása.

1.2.2. SI, SIR, SIRS modellek

Ahogyan az előző alfejezetben láthattuk, a független kaszkád modell viszonylag általános megközelítést alkalmaz, így széles körben alkalmazható, azonban bizonyos epidemiológiai szempontokat teljesen figyelmen kívül hagy. Ilyen lehet például a fertőzött és fertőzőképes állapot időtartama, a felépülés lehetősége, az ideiglenes vagy akár a teljes immunitás megszerzésének lehetősége. Az epidemiológiai területek fokozatos térnyerésével egyre és egyre erőteljesebb igény jelentkezett a fent említett szempontokat is figyelembe vevő diffúziós modellekre. Így születhetett meg az SI, SIR, SIRS modellek csoportja [9]. Az elnevezések mozaikszavakként értelmezhetőek, melyek a csúcsok lehetséges állapotaira és időbeli lefolyásukra utalnak. Ezek az állapotok a Susceptible (S) ~ fogékony, Infected (I) ~ fertőzött és Recovered (R) ~ felépült.

Az SI modellben fogékony és fertőzött állapotú csúcsokat különböztetünk meg. A fertőzött csúcsok a többi fertőzésre fogékony csúcsot fertőzhetik meg egymástól függetlenül adott β valószínűségi eloszlás szerint. A fertőzött állapot a folyamat végéig megmarad, felépülésre nincs lehetőség.

AZ SIR modell az SI modellt egészíti ki, a fertőzött csúcsok ugyanúgy adott β valószínűségi eloszlások szerint fertőzhetik meg egymástól függetlenül a többi csúcsot. Azonban itt már ezek a csúcsok fel is épülhetnek egy másik, de ugyanúgy adott γ valószínűségi eloszlás szerint. A felépült állapot a folyamat végéig megmarad, újrafertőződésre nincs lehetőség.

Az SIRS modell a SIR modellt egészíti ki, minden hasonlóan zajlik, azonban itt a felépült csúcsok már adott ξ valószínűségi eloszlással újból fertőzésre fogékonnyá válhatnak. Így természetesen a fertőzött állapot nem tekinthető a folyamat végéig fennmaradó tulajdonságnak, ahogyan a felépült és a fogékony sem.

Látható, hogy a fent bemutatott változatok már pontosabb és szofisztikáltabb fertőzésterjedési modellezést tesznek lehetővé, azonban sokkal bonyolultabbak és kevésbé hatékonyan skálázhatóak. Ezen dolgozatban azonban a célunk, hogy „*proof of concept*" jelleggel az egyik legegyszerűbb, legáltalánosabb modellen, az általánosított független kaszkád modellen keresztül mutassuk be megközelítésünk hatékonyságát. Ez a módszertan természetesen kiterjeszthető az alfejezetben ismertetett modellekre is.

1.3. Befolyás maximalizálás

A hálózati diffúzió tanulmányozásával kapcsolatos egyik legismertebb probléma a befolyás maximalizálás. Már a 2000-es évek elején is jelentős eredményeket értek el a terület kutatói [1], ennek ellenére napjainkban továbbra is sok nyitott kérdés van a témában. Kiindulásképp adott egy $G = (V, E)$ gráf, melyen egy szintén adott fertőzési modell értelmezhető. A továbbiakban mi a független kaszkád modellre fogunk fókuszálni, de természetesen a módszerek más modellekre is értelmezhetők, kiterjeszthetők. Adott még továbbá egy $k \ll n$ szám, ahol $n = |V|$. A célunk pedig a teljes hálózaton számolt fertőzöttség várható értékének maximalizálása azzal a feltétellel, hogy $|A_0| \leq k$, azaz a kezdeti fertőzött csúcshalmazunk legfeljebb k elemet tartalmazhat.

A problémára egy kézenfekvő és egyszerűen kivitelezhető módszer a mohó heurisztika. Eszerint induljunk ki egy üres halmazból, melyet iterációnként bővítsünk egy-egy új csúccsal. Az első ilyen iterációban legyen $|A_0| = 1$ a feltételünk, és keressük meg azt a csúcsot, amiből adott diffúziós modell szerint fertőzést elindítva a teljes hálózatra számolt fertőzöttség várható értéke maximális lesz. A rákövetkező második iterációban $|A_0| = 2$ a feltétel, és az előző iterációban kapott csúcshoz vehetünk hozzá egy újabbat ugyanazzal a feltétellel. Vagyis a cél, hogy az immár két elemű kezdeti fertőzött csúcshalmazból a teljes hálózatra számolt fertőzöttség várható értéke maximális legyen. Ezt addig ismételjük, ameddig meg nem kapjuk az $|A_0| = k$ elemszámú kezdeti fertőzött csúcshalmazt.

A szakirodalomban a problémával kapcsolatban több fontos tétel is kimondásra került, melyek természetesen bizonyítást is nyertek. Az egyik legfontosabb ilyen állítás az [1] cikkben, hogy a befolyás maximalizálási probléma NP-nehéz a független kaszkád modellre. Továbbá azt is belátták a szerzők, hogy a mohó heurisztika a befolyás maximalizálási problémára $(1 - {}^{1}/_{e})$ közelítési pontosságot garantál. Ez azért volt jelentős előrelépés a területen, mert így tesztesetekeken keresztül bármely más heurisztika pontosságának korlátját megbecsülhetjük a mohó módszer eredményével összehasonlítva.

1.4. Hálózati szimuláció

Az előző fejezetben bemutatott fertőzésterjedési modellek esetében érezhető, hogy a megfelelő számítások elvégzése sokszor túlzottan időigényes lehet. A független kaszkád modell esetében a [10] cikk szerzői belátták, hogy bármely kezdeti fertőzött csúcshalmazra a fertőzési függvény kiszámítása #P-teljes. Ez korábbi kutatási eredmények alapján várható volt, hiszen az [1] cikkben már arról értekeztek, hogy a fertőzés maximalizálási probléma elegendő mennyiségű szimulációval tetszőleges pontossággal közelíthető a független kaszkád modell esetében.

Mivel a fertőzési függvények kiszámítása #P-teljes a független kaszkád modell esetében, ezért mi is szimulációk futtatása mellett döntöttünk. Az [1] cikk alapján a független kaszkád modell közelítése interpretálható s darab szimuláció futtatásával. Ezt megvalósíthatjuk olyan módon, hogy az élvalószínűségeknek megfelelően s darab súlyozatlan gráf példányt generálunk. Minden egyes ilyen példányban az egyes éleket a rajtuk értelmezett w_e valószínűségekkel egymástól függetlenül megtartjuk, egyébként eldobjuk. Ezen gráf példányokban megvizsgálhatjuk, hogy a kiindulási fertőzött csúcsokból mely további csúcsok érhetőek el. Ezeken a példányokon számolt átlag fogja a fertőzési értékeket közelíteni. Ennek a technikai megvalósítását a következő bekezdésben részletesen ismertetem.

A teljes szimulációhoz bemenetként szokásosan szükségünk van a $G = (V, E)$ gráfra, valamint egy s példányszámra (szimulációhoz szükséges minták száma). Minden csúcshoz rendelünk egy f_v fertőzési változót, melyet kezdetben 0-ra inicializálunk. Az algoritmus törzsében egy ciklusunk lesz, ami s-szer fog lefutni. Az A_0 halmaz vagy adott, vagy a csúcsokon lévő w_v súlyokat felhasználva minden csúcs a rajta lévő valószínűséggel lesz kezdeti fertőzött adott gráf példányban. Minden iterációban minden $e \in E$ élt a rajta lévő w_e valószínűséggel megtartunk, egyébként eldobjuk a hálózatból. A gráf megtartott éleivel, valamint az eredetileg adott vagy kiszámolt A_0 halmazzal elő is állt egy ilyen gráf példányunk. Ezek után $\forall v \in V$ csúcsra megvizsgáljuk, hogy ebben a konkrét példányban elérhető-e valamely kezdeti fertőzött csúcsból. Ez alatt azt kell érteni, hogy létezik-e valamely A_0-beli csúcsból út adott $v \in V$ csúcshoz. Ha igen, akkor a csúcshoz rendelt f_v értéket megnöveljük eggyel. Értelemszerűen $\forall a \in A_0$ csúcs fertőzési értékéhet hozzáadunk egyet, azaz nulla hosszú utakat is engedélyezünk e tekintetben. Az összes iteráció lefutása után ezen f_v értékeket $\forall v \in V$ csúcsra leosztjuk az s példányszámmal, azaz megkapjuk, mely csúcsok átlagosan hányszor fertőződtek meg a szimulációk futtatása során. Az algoritmus visszatérési értéke pedig ezen f_v értékek együttese lesz.

Az [1] cikkben szintén bizonyítást nyert, hogy mennyi szimuláció futtatása szükséges adott pontosság adott valószínűséggel való eléréséhez. Ezt a módszertant a [8] cikk alapján

továbbfejlesztve az általánosított független kaszkád modellre a következő állítások beláthatóak. Legyen p_{avg} a $G = (V, E)$ gráf a priori fertőzési értékeinek átlaga, továbbá legyen $\varepsilon, \delta \in (0,1)$. Ekkor az a posteriori fertőzési értékek összege $(1 \pm \varepsilon)$ pontossággal közelíthető $(1 - \delta)$ valószínűséggel, amennyiben a futtatott szimulációk száma legalább $\ln\left(\frac{2}{\delta}\right) \Big/ (2 \cdot p_{avg}^2 \cdot \varepsilon^2)$.

Azaz, ha tudjuk, milyen pontosak szeretnénk lenni, valamint mekkora valószínűséggel, ki tudjuk számolni, ehhez hány szimulációt kell futtatnunk. A továbbiakban hálózati szimuláció alatt mindig az általánosított független kaszkádon értelmezett változatot fogjuk érteni, de erre gyakran csak egyszerűen független kaszkádként fogunk hivatkozni.

Az előző bekezdésben bemutatott teljes szimulációs algoritmust pszeudokód formájában a következőképpen írhatjuk fel:

$BEMENET$: $G(V, E)$, s

$j \leftarrow 0$

$\forall v \in V$: $f_v \leftarrow 0$

$AMÍG\ j < s$:

    $A_0 \leftarrow kezdeti\ fertőzött\ csúcsok\ (adott\ vagy\ kiszámolás\ adott\ példányra)$

    $\forall e \in E$ $él\ megtartása\ w_e\ valószínűséggel$

    $HA\ v \in V\ elérhető\ valamely\ a \in A_0 - ból$:

        v: $f_v \leftarrow f_v + 1$

    $HA\ VÉGE$

    $j \leftarrow j + 1$

$AMÍG\ VÉGE$

$\forall v \in V$: $f_v \leftarrow \frac{f_v}{s}$

$KIMENET$: $\forall v \in V$ $csúcs$ f_v $értéke$

1.2. pszeudokód: Teljes szimuláció

1.5. Lineáris programozás

A második és harmadik fejezetben bemutatott két probléma megértéséhez szükséges a lineáris programozás (röviden LP) fogalmának érintőleges ismerete. Egy matematikai modellt építünk fel, melyben különböző változóink vannak, ezek értékei lehetnek binárisak, egészek vagy folytonosak. Amennyiben csak egész értékű megoldások a megengedettek, úgy egész értékű programozási feladatról beszélhetünk (röviden IP). A változóinkra lineáris feltételeket írhatunk fel, így egy lineáris feltételrendszert kapunk. Adott továbbá egy lineáris célfüggvény, melyet maximalizálni vagy minimalizálni kell. Bizonyos feladatokat érdemes ilyen formában felírni, hiszen a szakirodalomban egészen hatékony megoldások születtek különféle tulajdonságokkal rendelkező LP, vagy éppen IP modellekre. A megértéshez szükséges ismeretek nem haladják meg az alapvető egyetemi tananyagot, így ezekre bővebben nem térünk ki, részletesebb leírás a [11] könyvben olvasható.

2. EpiControl probléma

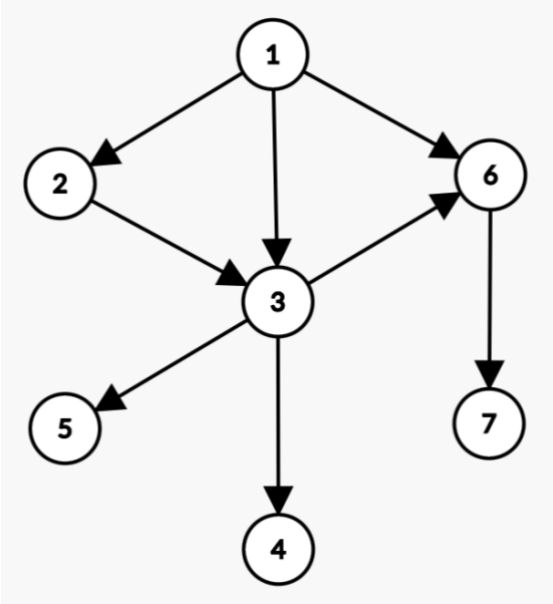
Ahogy azt már a korábbi fejezetekben említettük, a fertőzés terjedésének vizsgálatát a független kaszkád modell alapján fogjuk bemutatni a dolgozatban. Jelen fejezetben a [2] cikkben található *EpiControl* problémát dolgozzuk fel, értelmezzük az alapfeladatot, majd egy saját implementációt is adunk, hogy a későbbiekben tesztelni tudjuk a módszer hatékonyságát.

Az *EpiControl* célja, hogy egy adott $G = (V, E)$ hálózatra olyan vakcinázási stratégiát dolgozzon ki, amely egy járvány esetén a teljes hálózatra számolt fertőzöttség várható értékét a lehető legjobban igyekszik minimalizálni. A cikk írói a feladatot az SIR modellre fogalmazták meg, azonban mi a dolgozatban az „*SAAROUND for 1sEpiControl*”-ként hivatkozott algoritmusra fogunk csak fókuszálni, amiben megfertőződés utáni felépülésről nem esik szó, valamint a korábban bemutatott hálózati szimuláció a független kaszkád modell szerint történik. Fontos hangsúlyoznunk, hogy az *1sEpiControl* a megoldandó problémát, az *SAAROUND* pedig magát a szerzők által kidolgozott algoritmust jelenti.

Az *1sEpiControl* feladat tekintetében adott B_0 mennyiségű vakcina áll rendelkezésre, melyet a fertőzési folyamat kezdete előtt használhatunk fel. Az *SAAROUND* néven hivatkozott algoritmus ezt a problémát egy lineáris programozási feladatként definiálja, majd annak egy relaxált, valós változókat tartalmazó változatát oldja meg, utána pedig ennek eredményeit egy úgynevezett „*randomized rounding*” lépéssel kerekíti egész értékekre. A cikk későbbi fejezeteiben a módszer pontosságáról értekeznek, melyet formálisan be is bizonyítanak.

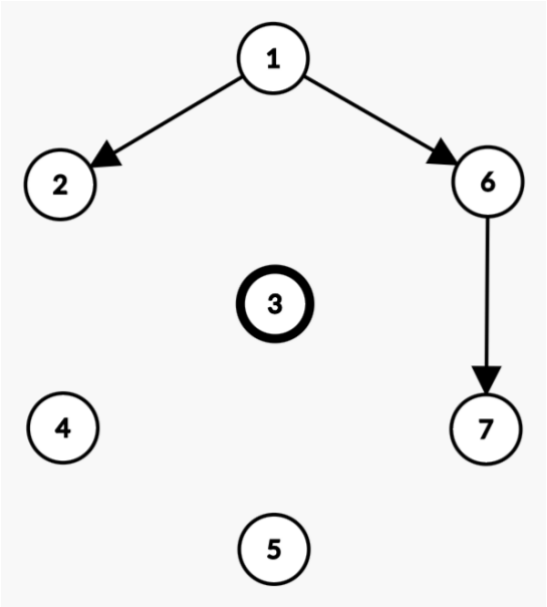
Az *1sEpiControl* probléma formális definíciója a következő: adott egy $G = (V, E)$ gráf, melyen s_v csúcs- és p_e élvalószínűségek is szerepelnek. Ezeket rendre s és p vektorok tartalmazzák. Egy tetszőlegesen választott $v \in V$ csúcsra s_v azt adja meg, hogy az mekkora valószínűséggel lesz kezdeti fertőzött egy gráf példányban. A p_e valószínűségek pedig megmutatják, hogy mekkora valószínűséggel terjedhet fertőzés adott éleken egy ilyen bizonyos példányban. $EInf(G, s)$ je-lölje a folyamatban megfertőződött csúcsok számának várható értékét, ezt szeretnénk minimalizálni. A vakcinázás értelmezéséhez vezessük be az x_{vt} változókat, amiknek értéke 1, ha adott v csúcs t időpillanatban kap oltást. Ezek vektora az $X_t = \{v: x_{vt} = 1, v \in V\}$. Így már $EInf(G, s, X_0)$ jelentése is definiálható, ez a fertőzések számának várható értéké jelöli, ha X_0 szerint oltunk, vagyis $t = 0$ időpillanatban beoltjuk azon csúcsokat, melyekre $x_{v0} = 1$. Végül adott még egy B_0 költség, ami az elérhető vakcinák számát jelöli. Célunk tehát nem más, mint $X_0 \leq B_0$ megválasztása úgy, hogy $EInf(G, s, X_0)$ minimális legyen.

Az *1sEpiControl* probléma szemléltetéséhez tekintsük az alábbi egyszerű példát:



2.1. ábra: Szemléltető gráf

A *2.1. ábrán* 7 darab csúcsunk van, itt most az egyszerűség kedvéért a priori valószínűségekkel nem foglalkozunk. A csúcsok között összesen 8 darab irányított él fut, ezeken szintén nincsenek súlyok. Egy csúcs beoltása szemléletesen azt jelenti, hogy azt lényegében kiemelve a hálózatból a bele befutó és belőle kifutó éleket törölhetjük (hiszen az a csúcs a továbbiakban nem tud megfertőződni, illetve másnak sem képes átadni a fertőzést). Azaz például a 3-as csúcs beoltása így nézne ki:



2.2. ábra: 3-as csúcs beoltása

A *2.2. ábrán* jól látható, hogy a 8 élből csupán 5 darab maradt meg. Természetesen ez nem jelenti következésképp azt, hogy ez a legjobb oltási stratégia $B_0 = 1$ darab rendelkezésre álló vakcina esetén. Igaz, hogy így tudunk egy csúcs kiemelésével a legtöbb élt törölni, de előfordulhat az, hogy a 3-as csúcs minden kapcsolata nagyon gyenge (p_e élvalószínűségei kicsik) és azokon az esetek döntő többségében nem is terjedne fertőzés. Így lehet, hogy valami másik csúcs vakcinázása lenne a legjobb döntés.

Az összes élen azonosan $p_e = 1$ terjedési valószínűségeket feltételezve a fertőzésmaximalizálási probléma megoldása a $k = 1$ esetre az 1-es csúcs kiválasztása lenne. Ha csak egy csúcsból indíthatunk el fertőzést, az 1-est kiválasztva kezdeti fertőzött csúcsnak az összes többi csúcs megfertőződik, és ez másképp nem érhető el (feltéve, hogy csak 1 darab kezdeti fertőzött csúcsunk lehet). Ilyen szempontból ez is tűnhet hatékony választásnak.

A fenti egyszerű példa azt is jól szemlélteti, hogy konkrét számolások nélkül már kis méretű gráfokon sem egyértelmű az *1sEpiControl* feladat megoldása. Bár itt még az s_v csúcs- és p_e élvalószínűségeket ismerve direkt módon ki tudnánk számolni a tényleges fertőzési függvény értékeket, azonban ez nagy gráfok esetében már jóval időigényesebb feladatnak ígérkezne. Ez adta az alapvető motivációnkat, a problémát valamilyen logikusnak tűnő gondolat alapján egy teljesen más, azonban sokkal hatékonyabban számolható probléma megoldásával szerettük volna közelíteni. A [2] cikk szerzői is egy közelítő eljárást dolgoztak ki, azonban ők a feladatot lineáris programozási modellként írták fel, és annak egy relaxált változatát oldották meg. Így ők valós értékkészletű eredményeket kaptak, azonban vakcinázás szempontjából egész megoldásoknak lenne értelme (hiszen egy embert nem lehet „félig beoltani”). Annak érdekében, hogy egész megoldásokat kapjanak, egy véletlent is használó kerekítési eljárást dolgoztak ki, melynek közelítési pontosságát elméleti számításokkal igazolták.

Az *1sEpiControl* feladat megoldására a cikk szerzői által írt *SAAROUND* algoritmust pszeudokód segítségével szemléltetjük:

$BEMENET$: $G(V,E)$, s, B_0

1: $H_j = (V, E_j)$ *példányok elkészítése;* $j = 1, \dots, M \implies M$ *darab példány*

$\forall j$: ($\forall v \in V$: $v \in src(H_j)$ s_v *valószínűséggel és* $\forall e \in E$: $e \in E_j$ p_e *valószínűséggel*)

2: *Oldjuk meg a következő lineáris programozási feladatot* ($LPsaa$):

(1) $$min \frac{1}{M} \sum_j \sum_v y_{vj}$$

(2) $$\forall j, \forall u \in V\colon\ y_{uj} \leq 1 - x_{u0}$$

(3) $$\forall j, \forall u \in V, (w,u) \in E_j\colon\ y_{uj} \geq y_{wj} - x_{u0}$$

(4) $$\forall j, \forall s \in src(H_j)\colon\ y_{sj} = 1 - x_{s0}$$

(5) $$\sum_{u \in V} x_{u0} \leq B_0$$

(6) $$Minden\ változó \in [0,1]$$

3: *Az* $LPsaa$ *valós megoldásai legyenek* x *és* y *vektorok, ezeket a következőképpen kerekítjük* X *és* Y *egész értékeket tartalmazó vektorokká*:

(1) $Ha\ y_{vj} \in \{0,1\}, akkor\ Y_{vj} = y_{vj}$, *hasonlóan ha* $x_{v0} \in \{0,1\}, akkor\ X_{v0} = x_{v0}$

(2) $Ha\ y_{vj} \geq \frac{1}{2}, akkor\ Y_{vj} = 1, egyébként\ Y_{vj} = 0$

(3) $\forall v$: $X_{v0} = 1$ *a következő valószínűséggel*: $\min\{1, 2x_{v0} \log(4nMN_v)\}$,
ahol $n = |V|$ *és* N *a maximális utak száma* H_j *példányban bármelyik* $src(H_j)$ *beli csúcsból az adott* v *csúcsba*

(4) $X_0 = \{v: X_{v0} = 1\}$

$KIMENET$: X_0

2.3. pszeudokód: SAAROUND algoritmus

Az x_{v0} változók jelentése már definiálásra került az előző alfejezetben, ezek értéke 1, ha a $t = 0$ időpillanatban v csúcs kapott vakcinát. Új változókat is bevezetünk, y_{vj} azt jelöli, hogy v csúcs megfertőződött-e adott H_j gráf példányban. Vagyis minden ilyen y_{vj} azt mutatja, hogy ezen példány kezdeti fertőzött $src(H_j)$ csúcsainak valamelyikéből vezet-e a v csúcshoz út oltatlan csúcsokon keresztül. A pszeudokód 1-es pontjában szintén leírtuk a gráf példányok generálásának menetét formálisan, egy ilyen példányban minden csúcs s_v valószínűséggel lesz kezdeti fertőzött, és minden él p_e valószínűséggel kerül bele. M pedig nem más, mint az ilyen módon legenerált példányok száma.

Azt már korábban láthattuk, hogy a fertőzési függvények kiszámítása igencsak idő- és erőforrás-igényes feladat, ezért a cikk írói is szimulációs eszközökhöz nyúltak, valamint ennek pontosságát is megvizsgálták. A problémát LP feladatként írták fel, a minimalizálandó célfüggvény (1) a szimulációk során átlagosan megfertőződött csúcsok száma. Ha egy csúcs be van oltva, nem fertőződhet meg (2), egyébként a nem beoltott szomszédos csúcspárok között terjedhet a fertőzés (3). Ha egy kezdeti fertőzött csúcsot beoltunk, akkor ő nem lehet fertőzött, ha pedig nem oltjuk be, akkor valóban fertőzött lesz (4). Legfeljebb B_0 darab csúcsot olthatunk be a fertőzési folyamat előtt (5).

Még egy érdekes gondolat a (6) feltétel, miszerint minden változó valós értékeket vehet fel a $[0,1]$ intervallumon. Természetesen egy csúcs nem tud „félig” megfertőződni, és természetesen a beoltás is csak egész értékekkel működhet. Azonban a modell bonyolultsága láthatóan a példányok számának növelésével együtt rohamosan nő. Ezen megfontolásokból döntöttek egy relaxált probléma megoldásán, lényegében feltételeket engedtek el a változókra vonatkozóan, azok nem voltak lekorlátozva kizárólag egész értékekre. Hogy ténylegesen egész értékű megoldásokat kapjanak, egy úgynevezett „*randomized rounding*” kerekítési stratégiát találtak ki. Az eredetileg egész értékű megoldásokat változatlanul hagyták. Ha egy csúcs legalább az esetek felében megfertőződött, az Y változóját 1-re állították, egyébként 0-ra. Ez csak a megoldás fízibilitásának szempontjából fontos, mivel az algoritmus kimenete az X vektor alapján képződik. Az x_{v0} változók kerekítése valószínűségi alapon történik, innen ered a „*randomized*” jelző, a kerekítéshez pedig több paramétert felhasználnak, magát az x_{v0} értékét, a csúcsok és szimulációk számát, valamint egy N_v számot is. Ennek számolásában rejlik a kerekítés pontosságának kulcsa, ez a szám minden egyes csúcsra megadja, hogy azt maximálisan hány különböző úton lehetett elérni az adott gráf példány bármelyik $src(H_j)$ csúcsából. A cikk második harmada lényegében a kerekítés helyességének bizonyításával foglalkozik. A fejezetben leírtak alapján mi is elkészítettük az algoritmus implementációját, eredményeinket a későbbiekben ismertetjük.

3. Fertőzés monitorozási probléma

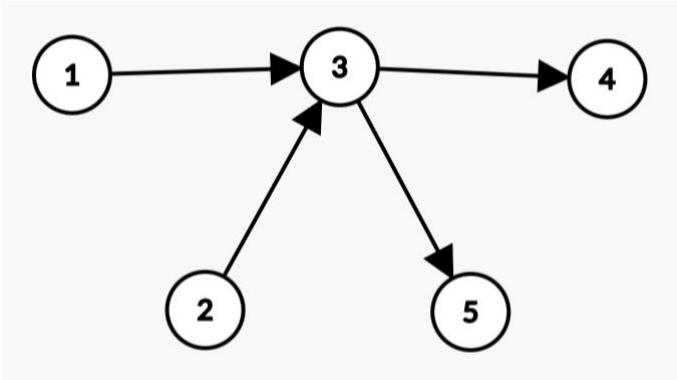
Ebben a fejezetben a [3] cikkben bevezetett fertőzés monitorozási problémát fogjuk definiálni, bizonyos tulajdonságait megvizsgálni, majd egy vele ekvivalens folyam modell segítségével közelíteni. A folyam modellt egy viszonylag jól skálázható, a szimulációk számától független méretű, egész értékű lineáris programozási feladatként fogjuk megoldani.

Először a fertőzés monitorozási problémát fogjuk formálisan definiálni. Szintén adottnak tekintjük a $G = (V, E)$ gráfunkat, az éleink szokásos módon irányítottak lesznek. Továbbra is súlyozott gráfokról beszélünk, az élsúlyok itt is ugyanúgy fertőzésterjedési valószínűségeket jelentenek a független kaszkád modell szerint. A csúcsainkon pedig a priori fertőzési értékek lesznek, melyek segítségével meghatározható a kezdeti fertőzött csúcsok eloszlása. Az így megkapott S_0 kezdeti fertőzött csúcshalmazból fog a fertőzés elindulni, és a megfertőzött csúcsok számának $\sigma(S_0)$ várható értékét fogjuk vizsgálni. Ugyanúgy érvényes minden korábban bemutatott eredmény a fertőzés maximalizálási problémával, valamint az általánosított független kaszkád modellre való áttéréssel kapcsolatban is.

A fertőzés monitorozási probléma alapötlete, hogy egyes csúcsok lokális fertőzési tulajdonságait próbáljuk kifejezni. Egy konkrét fertőzési folyamatot leírhatunk egy fákból álló irányított erdő segítségével, ahol a fák gyökerei a kezdeti fertőzött csúcsok és a fák szintjei azt reprezentálják, hogy melyik csúcs hány lépés után fertőződött meg. Hogy valóban fákat kapjunk és ne legyenek köreink, fontos megkötés, hogy egy adott szinten a fertőzési sorrendet lerögzítsük, és a továbbiakban mindig csak fogékony csúcsok fertőződhetnek meg. Ezeket fogjuk a későbbiekben erdő példányoknak nevezni, azaz a fertőzési folyamat egy konkrét megvalósulását irányított fák alkotta erdővel szemléltetjük. Szintén jegyezzük meg, hogy ezek az erdő példányok már súlyozatlanok lesznek, bennük sem a csúcsokon, sem az éleken nem lesznek súlyok. Ezt a konkrétan megvalósult fertőzési folyamatot a [8] könyvben „*branching process*” néven ismertetik, azaz amikor a fertőzés ténylegesen elindul egy csúcshalmazból és bizonyos útvonalakon további csúcsok fertőződnek meg. Így egy S csúcshalmaz lokális fertőzése alatt a továbbiakban érthetjük egy konkrét F erdő példányában az S leszármazottjainak $\mu_F(S)$ számát. Mivel sztochasztikus diffúziós folyamatokról beszélünk, vehetjük ezen $\mu_F(S)$ lokális fertőzések eloszlását. Így definiálhatjuk a $\mu_G(S)$ lokális fertőzési indexet, ami megadja a G gráf egy véletlen F erdő példányában a $\mu_F(S)$ lokális fertőzés várható értékét. A fertőzés monitorozási problémában éppen ezt a $\mu_G(S)$ lokális fertőzési indexet szeretnénk majd maximalizálni. Ehhez szokásosan adott egy $G = (V, E)$ gráf, valamint egy k szám, amire $|S| = k$.

A feladat tehát azon k méretű S csúcshalmaz meghatározása, amire $\mu_G(S)$ maximális. Tehát összegzve az előbbiekben leírtakat, egy gráf példányon belül számtalan erdő példány lehet, azonban a gráf példányokon keresztül ezen erdő példányok egyszerűen kiszámolhatók és jól becsülhető velük a teljes hálózat fertőzöttsége. Ezen erdő példányoknak az eloszlását pedig az a priori fertőzések és az élvalószínűségek határozzák meg.

A fertőzés monitorozási probléma szemléltetéséhez tekintsük az alábbi példát:



3.1. ábra: Szemléltető gráf

Tegyük fel, hogy a *3.1. ábrán* egy konkrét erdő példány látható. Ha $|S| = k = 1$, akkor ebben az F erdő példányban a $\mu_F(S)$ értékek könnyen meghatározhatóak. Minden egyes csúcsra külön ki kell számolnunk a leszármazottak számát, ami valójában annak meghatározását jelenti, hogy adott csúcsból hány másik csúcsot tudunk elérni F-ben. Technikai megjegyzés, hogy számolás közben elérhetőnek tekintjük a kiindulási csúcsot is. Így például a 3-as csúcsból elérhető önmaga, valamint a 4-es és 5-ös csúcsok, azaz $\mu_F(3) = 3$. Hasonlóan $\mu_F(1) = 4$, mivel elérhető belőle a 3-as csúcs, onnan pedig tovább a 4-es és 5-ös.

A gráf $\mu_G(S)$ lokális fertőzési indexének kiszámolásához szükségünk van minden csúcs a priori fertőzési értékére, valamint az élek súlyaira is. Előbbi legyen minden csúcsra 0.5, utóbbi pedig 1, valamint $|S| = k = 1$ továbbra is. Ekkor például $\mu_G(2) = \frac{1}{2} \cdot 4 = 2$, mivel a 2-es csúcs 0.5 valószínűséggel lesz kezdeti fertőzött, és ekkor mindig 4 darab leszármazottja lesz. Hasonló módon számolva $\mu_G(3) = \frac{7}{8} \cdot 3 = \frac{21}{8} = 2.625$, hiszen a 3-as csúcs csak akkor nem lesz fertőzött, ha sem ő, sem pedig az 1-es és 2-es csúcs nem fertőzött, minden más esetben igen. Komplementert számolva a 3-as csúcs $1 - \frac{1}{2} \cdot \frac{1}{2} \cdot \frac{1}{2} = 1 - \frac{1}{8} = \frac{7}{8}$ valószínűséggel lesz fertőzött, és ekkor minden esetben 3 darab leszármazottja lesz. Vagyis itt a legnagyobb $\mu_G(S)$ értékkel a 3-as csúcs rendelkezik, őt lenne érdemes választani.

Ha a fertőzés maximalizálási problémára gondolunk vissza, ebben a gráfban az 1-es vagy 2-es csúcsot lenne érdemes kiválasztani. Hiszen ekkor várható értékben 4 darab csúcs lesz fertőzött, egyébként ennél mindig kevesebb. Az imént, valamint az előző fejezetben is vizsgált példák is jól mutatják, hogy bizonyos fertőzésterjedéssel kapcsolatos problémák megoldása már kis méretű gráfokon sem biztos, hogy egyértelmű, a kapott eredmények jelentősen eltérhetnek egymástól.

A fertőzés monitorozási problémára a cikk szerzői egy mohó algoritmust mutattak be, melynek pontosságára garanciát is adtak. A bizonyításokban sokszor nyúltak vissza a terület egyik megalapozó cikkéhez [1], főként a $\mu_G(S)$ szubmodularitási tulajdonságára hivatkozva. Jelen dolgozatban az volt a célunk, hogy ezen problémára egy folyam modellt konstruáljunk, ezt implementáljuk, majd vizsgáljuk meg a hatékonyságát az *1sEpiControl* problémán.

3.1. Folyam modell

Ebben az alfejezetben a fertőzés monitorozási problémára fogunk egy folyam modellt megadni. Ehhez először definiáljuk a $G = (V, E)$ gráf G^* lezártját. A G^* gráf csúcsai már súlyozatlanok lesznek, éleire viszont új súlyok kerülnek. Ezek a c_{uv} súlyok az (u, v) élen keresztül elérhető csúcsok számának várható értékét adják meg egy véletlen erdő példányban. Ezt a [3] cikkben bemutatott szélességi bejárás alapján létrejött szinteken visszafelé haladva ki tudjuk számolni. Minden élre összegezzük ezeket az értékeket a példányok alapján, végül pedig egy átlagot számolunk a példányok számával való leosztással. Hogy ezen élsúlyok kivétel nélkül a $[0,1]$ intervallumba essenek, egy normalizáló lépést fogunk végrehajtani. Ehhez szükségünk lesz a $C_{in}(v)$ és $C_{out}(v)$ értékekre: $\forall v \in V: C_{in}(v) = \sum_{(u,v)\in E} c(u, v)$ és $C_{out}(v) = \sum_{(v,w)\in E} c(v, w)$. Ezek után vesszük a $C_{in}(v)$ értékek maximumát, és minden c_{uv} súlyt leosztunk vele. Az eredmények ismertetésénél majd láthatjuk, hogy ez a normalizáló lépés mekkora hatékonyságbeli javulást hozott a teszteseteinkben.

A folyam modellünk teljességéhez szükségünk van továbbá egy s forrás és egy t nyelő csúcsra is. A forrás csúcsból húzunk egy élt minden másik csúcsba a nyelőt leszámítva, valamint a forrás csúcsot kivéve minden más csúcsból húzunk egy élt a nyelő csúcsba. A nyelő csúcsba tartó élek kapacitásai rendre a normalizálás után kapott $C_{in}(v) - C_{out}(v)$ különbségek lesznek. A forrás csúcsból induló élekre ezek után nem is lényeges kapacitás korlátot megadni a folyam megmaradás törvénye miatt.

Az előbbiekben definiáltuk a folyam modellünket, most pedig összegezzük, hogyan is fogjuk ezzel közelíteni a fertőzés monitorozási problémát. Generáljuk a G_1, ... , G_i gráf példányokat, majd rendre meghatározzuk ezek G_1^*, ... , G_i^* lezártjait. A lezártak segítségével tudjuk magát a G^* gráfot közelíteni, amire aztán megoldjuk a költség megszorításos maximális folyam modellünket.

A G^* gráf approximációja a következőképpen történik. Egy $BF[G']$ redukált szélességi erdő bejárást hajtunk végre, a kezdeti fertőzött csúcsokból indulva azokat éleket tartjuk meg, melyek szomszédos szinteket kötnek össze, a többit eldobjuk. Így kapjuk meg a $BF[E']$ éleket. Innen az alábbi pszeudokóddal leírt algoritmus adja kimenetként a folyam modellhez szükséges c_{uv} értékeinket:

$INPUT$: $BF[G'], r\ szinttel$ és $S \subseteq V\ kezdeti\ fertőzött\ csúcshalmazzal$

$FOREACH\ v\ csúcs\ az\ r.szinten$:

    $\mu_{G'}(v) = 1$

$FOR\ i = r\ DOWNTO\ 2\ DO$:

    $FOR\ v\ IN\ szint_i$:

        $FOREACH\ (u, v)\ bemenő\ él$:

            $c_{uv} = \frac{1}{d} \cdot \mu_{G'}(v)\ (ahol\ d\ a\ v\ csúcs\ befoka\ BF[G'] - ben)$

    $FOR\ u\ IN\ szint_{i-1}$:

        $\mu_{G'}(u) = 1 + \sum_{u' \notin S \& (u,u') \in BF[E']} c_{uu'}$

$END\ FOR$

$OUTPUT$: $\forall (u, v) \in E(G') - re\ c_{uv}\ kapacitás\ érték$

3.2. pszeudokód: G^* approximációja

3.2. Költség megszorításos maximális folyam

Az előbbi alfejezetben bemutatott folyamon a célfüggvény természetesen a nyelő csúcsba eljuttatott folyam értéke, ezt szeretnénk maximalizálni. Azonban itt lesz egy olyan kiegészítő feltételünk, hogy a forrás csúcsból legfeljebb adott k darab csúcs felé indulhat el a folyam. Így ez nem egy szokványos maximum folyam probléma lesz, hanem annak egy költség megszorításos változata. Ha egy valós értékű maximális folyam problémát kellene kiszámolnunk, az polinom időben megvalósítható lenne, azonban itt a költség megszorítás miatt a feladat érezhetően bonyolultabb. Egy másik fontos megjegyzés, hogy valós és egész értékű változóink is vannak a feladatban, ezért a problémát vegyes értékű programozási feladatként (MILP) fogjuk kezelni.

A pszeudokód bemutatása előtt emeljük ki, hogy a feladat komplexitása miatt szintén szimulációk futtatása mellett döntöttünk. Természetesen ezekből elegendő mennyiség esetén itt is elérhető kívánt pontosság adott valószínűség mellett. Viszont az már azonnal látható, hogy az *SAAROUND* algoritmussal szemben a vegyes értékű programozási feladat mérete független a példányaink számától, ez csupán a folyam modell létrehozásában játszik szerepet. Egy másik megjegyzés, hogy mivel a forrás és nyelő csúcson kívüli más csúcsokban nem keletkezhet vagy veszhet el fertőzés, ezért a folyam maximalizálást úgy is érthetjük, hogy minél több fertőzés induljon ki a forrásból. A MILP modellben is ezt fogjuk használni.

A fentiekben bemutatott vegyes értékű programozási feladat pszeudokódja a következőképpen néz ki:

$BEMENET$: $G(V,E)$, s, k

1: $G^* = (V^*, E^*)$ *lezárt gráf létrehozása* $G(V,E)$ *– ből* s *darab példánnyal*

2: *Oldjuk meg a következő vegyes értékű programozási feladatot* ($MIPfolyam$):

$$(1) \qquad max \sum_v x_{sv}$$

$$(2) \qquad \forall (u,v) \in E^*: \; x_{uv} \leq c_{uv}$$

$$(3) \qquad \forall v \in V^*: x_{sv} \leq y_v$$

$$(4) \qquad \forall v \in V^*: \sum_v y_v \leq k$$

$$(5) \qquad \forall v \in V^* \text{ és } (i,v),(v,j) \in E^*: \sum_i x_{iv} = \sum_j x_{vj}$$

$$(6) \qquad x_{uv},\, x_{sv},\, x_{vt} \in [0,1]$$

$$(7) \qquad y_v \in \{0,1\}$$

3: $Y_v = \{y_v : y_v = 1\}$

$KIMENET$: Y_v

3.3. pszeudokód: Költség megszorításos maximális folyam

A célunk, mint említettük, a forrásból kiinduló fertőzés maximalizálása (1). A feltételeink, hogy az éleken a kapacitást nem léphetjük túl (2), és természetesen anyag se nem keletkezhet, se nem veszhet el a forrás, illetve nyelő csúcson kívül máshol (5). Ha valamennyi anyag befolyik egy csúcsba a forrásból, azt kiválasztottnak tekintjük (3) és legfeljebb k darab csúcsba mehet fertőzés innen (4). Az élek változói folytonosak (6), míg a csúcsoké binárisak (7).

4. Eredmények

Az előző fejezetekben bemutatásra került az *1sEpiControl*, valamint a fertőzés monitorozási probléma. Mint azt a bevezetőben már említettük, célunk a dolgozattal az volt, hogy a fertőzés blokkolási problémakörre fejlesszünk ki egy hatékony, hálózati folyammodell alapú módszert. Habár a hálózati diffúzió blokkolása és monitorozása látszólag teljesen különböző feladatok, tesztelési eredményeink szerint bizonyos szempontból ezen két feladat mégis hasonló jelleget mutat. Az algoritmusunk hatékonysága mellett azt is fontos kiemelnünk, hogy futásidő tekintetében kiemelkedően jól teljesített, lényegesen gyorsabban produkált kimenetet az *SAAROUND* implementációhoz képest. A soron következő alfejezetekben az ezeket igazoló eredményeinket fogjuk szemléltetni.

4.1. Tesztelési és kiértékelési módszertan

Ebben az alfejezetben a tesztelésünk során használt gráfok előállítását, továbbá az összehasonlított algoritmusokat mutatjuk be. Az ismertetett módszertanunkkal célunk „*proof of concept*” jelleggel a tudományág egy kevésbé kutatott területének feltérképezése, valamint jövőbeli kutatási irányvonalak megadása.

4.1.1. Tesztgráfok előállítása

A különböző módszerek teszteléséhez kis méretű, mesterségesen előállított gráfokat használtunk. Ezek csúcsaira és éleire adott eloszlások szerint generáltunk súlyokat. A kiértékelésünk 108 darab 1000 csúcsú gráfon történt, melyekben egyenként körülbelül 7000 él található, normális eloszlású csúcs- illetve élvalószínűségekkel. Valós alkalmazásokban természetesen jóval nagyobb méretű gráfokkal találkozhatunk, azonban az algoritmusok paramétereinek finomhangolása és tesztelése szempontjából érdemesebb egyszerűbb példák vizsgálatával kezdeni.

Ezen gráfok egy témavezetőimmel közös korábbi kutatás során lettek legenerálva [12]. Ehhez a [13] cikkben bemutatott, Andrea Lancichinetti és Santo Fortunato által megalkotott módszert használtuk a következő paraméterekkel:

- csúcsok száma: $N = 1000$
- átlagos fokszám: $k = 7$
- maximum fokszám: $maxk = 9$
- fokszám exponense: $t1 = -2$
- közösségméret exponense: $t2 = -1.5$
- minimum közösségméret: $minc = 10$
- maximum közösségméret: $maxc = 50$
- mixing paraméter: $mu \in (0.1, 0.2, 0.3, 0.4, 0.5, 0.6)$
- csúcsok közötti átfedések aránya: $on \in (0.1, 0.2, 0.3, 0.4, 0.5, 0.6)$
- különböző közösségekbe tartozás maximuma: $om \in (2, 3, 4)$

Az utolsó három paraméternél több értéket is felsoroltunk. A gráfokat generáló algoritmus ezek minden lehetséges kombinációját megkapja, vagyis összesen $6 \cdot 6 \cdot 3 = 108$ különböző gráfot szolgáltat eredményül. Ezek mindegyikére elvégeztük a következő alfejezetben bemutatott tesztelést, eredményeinket a későbbiekben részletesen ismertetjük.

4.1.2. Alkalmazott módszerek

Mind az *1sEpiControl*, mind pedig a fertőzés monitorozási probléma folyam modelles változata adott méretű csúcshalmazt ad vissza. Azt szerettük volna megvizsgálni, hogy az eredményül kapott csúcsok kivétele a hálózatból mennyiben képes a teljes hálózaton elindított fertőzési folyamatot blokkolni. Tesztelésünk során futtattuk a gráfjainkra a két módszert (a későbbi diagramon *SAA ROUND* és *MONITOROZÁS NORM*), majd az eredményül kapott csúcsokat kivettük a gráfból és fertőzési szimulációk futtatásával azt vizsgáltuk, mennyi a folyamat során a teljes hálózaton megfertőződött csúcsok számának várható értéke. A kivenni kívánt csúcsok számát minden esetben 50-re, azaz a gráfban található darabszám 5%-ára állítottuk.

Az előző fejezetben említett két módszernek egy-egy módosított változatát is teszteltük. A fertőzés monitorozási probléma esetében kipróbáltuk, mi történik akkor, ha a folyam modell megalkotásánál a normalizáló lépést teljesen kihagyjuk (diagramon *MONITOROZÁS*). A másik feladat, az *1sEpiControl* esetében pedig azt vizsgáltuk meg, mit eredményezne az *SAAROUND* algoritmus lineáris programozási modelljének IP változata (diagramon *SAA IP*), azaz relaxáció

és randomizált kerekítés helyett a feladat direkt megoldása. Ez azt jelenti, hogy egy nehezebb, egész értékű programozási modellt kell megoldanunk. Fontos megjegyeznünk, hogy nagyobb feladatok esetében az IP megoldó általában nem az optimumot adja, hanem egy ezt jól közelítő fízibilis megoldást. Természetesen minél hosszabb ideig engedjük a megoldót futni, annál pontosabb közelítést érhetünk el.

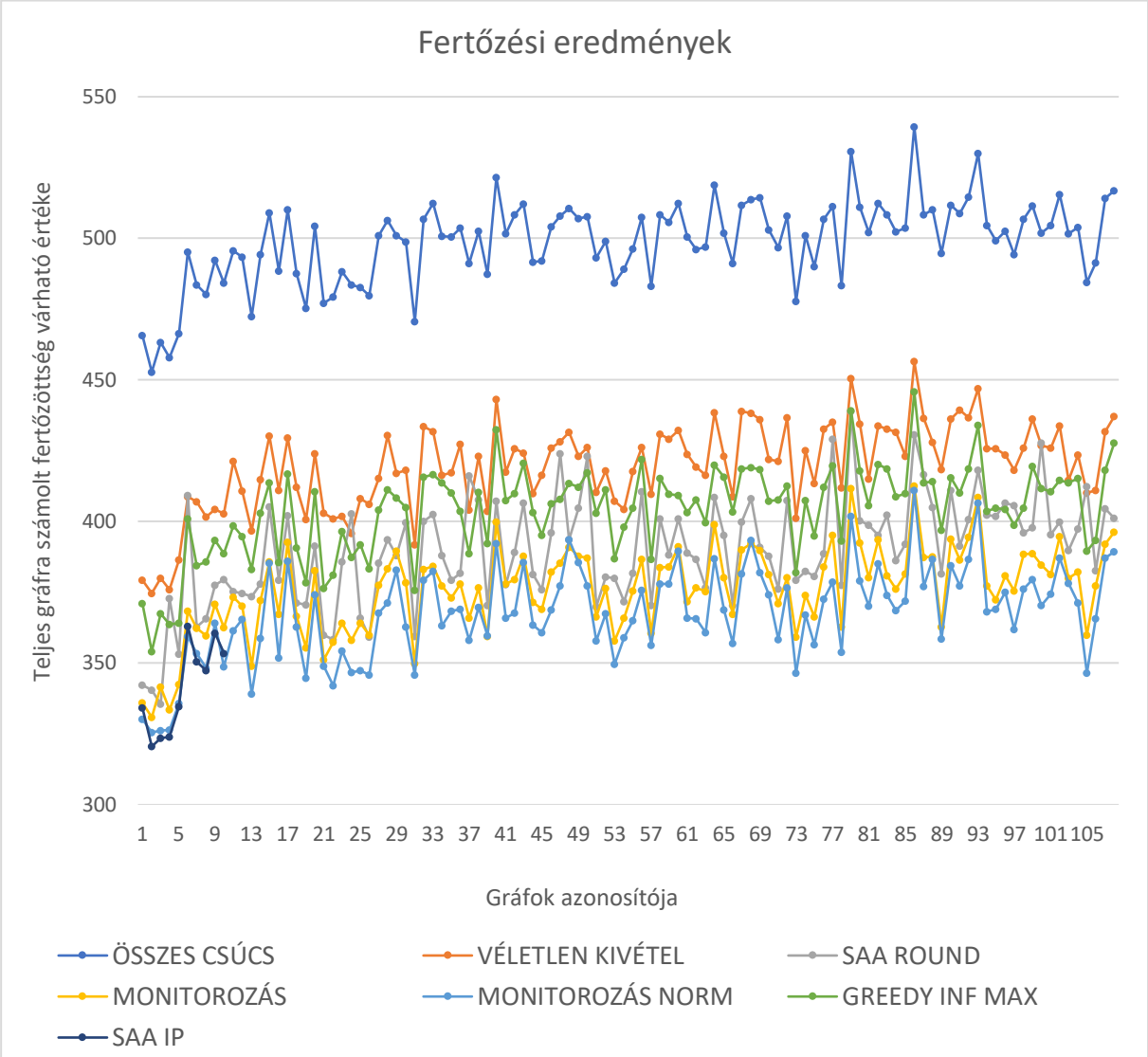
Témavezetőimmel közös korábbi kutatómunkáink során foglalkoztunk fertőzés maximalizálással, ahol szintén ezeken a gráfokon teszteltünk. Ezekben a kutatásokban egy mohó heurisztikával igyekeztünk a teljes hálózatra számolt maximális fertőzöttséget elérő adott elemszámú kezdeti fertőzött csúcshalmazt meghatározni. Kíváncsiak voltunk, mit tapasztalunk, ha ezeket a maximális fertőzöttséget előidéző csúcsokat távolítjuk el a hálózatból és így futtattunk fertőzési szimulációkat (diagramon *GREEDY INF MAX*).

Hogy legyen viszonyítási alapunk, úgynevezett „baseline” modelleket is implementáltunk. Egyrészt futtattunk fertőzési szimulációkat a teljes gráfra csúcsok kivétele nélkül, hogy lássuk, beavatkozás nélkül milyen jelleget mutat a hálózat (diagramon *ÖSSZES CSÚCS*). Azt is megvizsgáltuk, mi történik akkor, ha az 50 darab csúcsot véletlenszerűen választjuk ki (diagramon *VÉLETLEN KIVÉTEL*). Így egy olyan viszonyítási alapot is kaptunk, hogy egy tudományos alapokon nyugvó módszer valójában mennyivel eredményez nagyobb csökkenést egy bonyolultabb számításokat nem igénylő megközelítéshez képest.

4.2. Teszteredmények

Az előző alfejezetben leírt módokon kapott tesztelési eredményeinket az alábbi diagram szemlélteti. Az ábra vízszintes tengelyén a tesztgráfok azonosítói láthatók 1-től 108-ig, a függőleges tengelyen pedig a szimulációk során a teljes gráfra számolt fertőzöttség várható értékei a kiválasztott 50 darab csúcs vakcinázása után. Ez alól kivétel az *ÖSSZES CSÚCS* eset, hiszen ekkor nem történik vakcinázás, ehhez hasonlíthatjuk a többi módszer hatékonyságát. A diagram jobb átláthatósága miatt az egy algoritmushoz tartozó adatpontokat összekötöttük, de természetesen az eredményeket diszkrét módon kell értelmezni, a helyes megjelenítés kategóriánként különböző színű pontfelhőket eredményezne. Ezt az ábrát azonban átláthatatlannak gondoltuk, így döntöttünk az adatpontok összekötése mellett. Fontos még kiemelnünk, hogy az *SAA IP* módszert csak 10 darab tesztgráfon futtattuk, ennek okairól bővebben a futásidők bemutatásánál értekezünk majd.

A fentebbi alfejezetben leírt módokon kapott tesztelési eredményeink tehát a következő diagramon láthatóak:



4.1. diagram: Fertőzési eredmények

A fenti diagramon együtt látható az összes futtatási eredményünk. A zsúfolt megjelenés ellenére egyértelműen leolvasható, hogy a teszteléshez használt gráfjaink esetében a véletlenszerűen kiválasztott 50 darab csúcs kivétele is jelentős csökkenést hozott a teljes hálózatra számolt fertőzöttség várható értékében. Az is megfigyelhető, hogy az *SAAROUND* jobban teljesít, mint a véletlenszerű választás, azonban a mi általunk megalkotott költség megszorításos maximális folyam módszer bizonyult a legjobbnak. Ezeken felül a saját módszerünk futásidő és komplexitás szempontjából is jóval egyszerűbb volt, mint az *SAAROUND* megközelítés.

A teljes hálózatra számolt fertőzési eredményeinkről néhány számszerűsített információ az alábbi táblázatban olvasható. Itt a „*CSÖKKENÉS*” értékek az „*ÖSSZES CSÚCS*” átlagához mérten százalékos arányban értelmezhetőek.

	ÖSSZES CSÚCS	VÉLETLEN KIVÉTEL	GREEDY INF MAX	SAA ROUND	MON.	MON. NORM
ÁTLAG	498.63	419.46	403.96	389.95	376.02	367.64
SZÓRÁS	14.92	15.46	15.92	19.20	15.54	16.63
CSÖKKENÉS		15.88%	18.98%	21.80%	24.59%	26.27%

4.2. táblázat: Fertőzési eredmények összehasonlítása

Ahogy az a felsorolásból leolvasható, az általunk megalkotott módszer számottevően jobban teljesített, mint a véletlenszerű csúcskivétel vagy az *SAAROUND* algoritmus szerint legmegfelelőbb csúcsok. Természetesen ebben fontos szerepet játszik, hogy a monitorozási megközelítés lényegesen kisebb modellt hoz létre, mint az *SAAROUND*, így ott pontosabb szimulációs közelítést is reméltünk. Az is észrevehető, hogy az *SAAROUND* sokkal nagyobb szórást produkált, mint a másik kettő blokkolási módszer, ez feltehetően annak tudható be, hogy az algoritmust nem sikerült megfelelően finomhangolnunk a tesztelésünkhöz használt mesterségesen generált gráfjainkra.

Az *SAAIP* módszer azért nem került be a táblázatba, mert ott kevesebb futtatást sikerült csinálnunk. Futásidő tekintetében olyan jelentős növekedés volt tapasztalható az IP modell megoldásakor, hogy 1 órás időkorlátot kellett megszabnunk. Így 10 darab gráfra sikerült lefuttattuk ezt a módszert, a futások alatt pedig minden esetben 7-8% IP GAP-et értünk el vele. A *4.1. diagram* bal alsó sarkában jól látszik, hogy sikerült igen közel kerülnünk a fertőzés monitorozási problémára megalkotott költség megszorításos maximális folyam módszerünkkel az IP modell korlátolt futásidejű megoldásához, és ami azt illeti, nagyságrendekkel gyorsabban (körülbelül 60-ad annyi idő alatt). A futásidők összehasonlításával a következő bekezdésekben részletesebben is foglalkozunk.

Futásidők tekintetében megvizsgáltuk az egész folyamatot a bemeneti gráfok átadásától kezdve a teljes hálózatra számolt blokkolt fertőzési eredmények produkálásáig. A teszteléshez egy 2020-as, M1 processzoros, 8GB RAM-mal és 512GB SSD-vel felszerelt MacBook Pro-t használtunk. A különböző részfolyamatok átlagos futásidejei (másodpercben) az alábbi táblázatban olvashatók:

	ÖSSZES CSÚCS	VÉLETLEN KIVÉTEL	GREEDY INF MAX	SAA ROUND	MON.	MON. NORM
ALGORITMUS		14.56	201.26	326.37	56.44	56.84
FERTŐZÉS	104.02	106.58	109.71	105.81	108.40	104.95

4.3. táblázat: Futásidők

A fenti táblázatban az „*ALGORITMUS*” sor a fertőzési folyamat blokkoláshoz használt $k = 50$ darab csúcsot meghatározó algoritmus, a „*FERTŐZÉS*” sor pedig az utána következő fertőzési szimuláció futásidejét jelöli. Ahogy azt már korábban említettük, a fertőzési függvény számolása #P-teljes, ezért történt a tesztelés szimulációk futtatásával. Látható, hogy a véletlen kivétel a leggyorsabb, azonban a fertőzés monitorozás alapú megoldás sem hozott nagyságrendileg nagyobb futásidőt a teszteseteinkre. Ez nem mondható el az *SAAROUND* algoritmusról, itt már az általunk megalkotott módszerhez mérten is jelentős ugrás tapasztalható. Ezt azért is fontos kiemelnünk, hiszen mi kis méretű gráfokon teszteltünk, melyeknél a való életben jellemzően jóval nagyobb és sűrűbb hálózatok fordulnak elő.

A különböző hatékonyságok végső szemléltetéséhez minden módszert jellemezhetünk egy arányszámmal, ami megmondja, hogy 1 százaléknyi fertőzés csökkenéshez átlagosan hány másodpercre volt szüksége adott algoritmusnak. Minél kisebb ez a szám, annál hatékonyabb módszerről beszélhetünk, de a skála nem feltétlenül lineáris (például kétszer olyan hatékony algoritmusnál akár négyszer akkora futásidőt is tolerálhatunk, ekkor négyzetes lenne a skála).

A kapott hányadosokat az alábbi táblázatban szemléltetjük:

	VÉLETLEN KIVÉTEL	GREEDY INF MAX	SAA ROUND	MON.	MON. NORM	SAA IP
FUTÁSIDŐ	14.56	201.26	326.37	56.44	56.84	3602.14
CSÖKKENÉS	15.88%	18.98%	21.80%	24.59%	26.27%	28.06%
HÁNYADOS	0.92	10.60	14.97	2.30	2.16	128.37

4.4. táblázat: FUTÁSIDŐ / CSÖKKENÉS hányadosok

A *4.4. táblázatban* jól látszik, hogy a megalkotott módszerünk normalizálást is használó változata a definiált *HÁNYADOS* szempontjából a második legkisebb érték, ennél kisebb szám csak a *VÉLETLEN KIVÉTEL* oszlopában található. Azonban a *CSÖKKENÉS* különbsége annyira jelentős, hogy természetesen érdemesebb a *MON. NORM.* implementációt választani, ha vakcinázni szeretnénk.

Összefoglalva elmondható az eredményeink tekintetében, hogy az általunk megalkotott algoritmus szolgáltatta a második legnagyobb csökkenést az *1sEpiControl* fertőzés blokkolási problémára. Ennél csak az *SAAIP* adott kicsivel hatékonyabb vakcinázási stratégiát, azonban jelentős, 60-szor nagyobb futásidővel. Kijelenthetjük tehát, hogy a módszerünk és annak implementációja más szakirodalom-beli megközelítésekkel összehasonlítva is versenyképes a vizsgált epidemiológiai vonatkozású problémára.

Összefoglalás

Jelen diplomamunkában hálózati diffúziós folyamatokat vizsgáltunk, főként epidemiológiai és fertőzésterjedési szempontból. Szemléltetésképp a széles körökben elterjedt független kaszkád modellen keresztül mutattuk be a terjedési folyamatokat. A tudományterület alapvető ismereteinek és kérdéseinek felvázolása után két konkrét problémát, a [2]-ben bevezetett *EpiControl*, valamint a [3]-ban tárgyalt fertőzés monitorozási feladatot mutattunk be. Célunk az volt, hogy előbbi problémát közelítsük egy, az utóbbi feladatra megalkotott költség megszorításos folyam maximalizálási modellel.

A fentebb említett két problémára implementációt készítettünk el, hogy teszteseteken keresztül mutathassuk be, a két látszólag teljesen különböző feladatnak bizonyos szempontból mégis köze lehet egymáshoz.

A teszteléshez kis méretű, mesterségesen generált gráfokat használtunk fel. A kapott eredményeink a hipotézisünket alátámasztják, a fertőzés monitorozási probléma megoldását diffúzió blokkolási szempontból felhasználva valóban jelentősen sikerült csökkenteni a teljes hálózatra számolt fertőzöttség várható értékét.

Ezeken felül az is elmondható, hogy futásidő tekintetében is hatékonyabbnak bizonyult a megközelítésünk, mint az *SAAROUND* algoritmus. Ez a modellünk jobb skálázhatóságának köszönhető, hiszen ennek mérete nem függ közelítés során futtatott szimulációk számától.

A fertőzés monitorozási problémára megalkotott módszerünk hasonló eredményeket ad, mint az *1sEpiControl* egész értékű feladat időkorlátos változata, ami az esetünkben legfeljebb 7-8%-kal tért el az optimumtól. Ez felveti annak a lehetőségét, hogy a monitorozás során kapott eredményt kezdeti megoldásként beadva az *SAAROUND* vagy az *SAAIP* modellnek további javulást érhetünk el hatékonyság tekintetében.

Eredményeink jövőbeli továbbfejlesztési célokat is indukálnak, többek között egyéb diffúziós modelleken való tesztelést. Érdekes kérdés lehet még a módszerek skálázhatóságának és közelítési pontosságainak vizsgálata, valamint nagy méretű, valós hálózatokon való kísérletezés is. Összességében elmondható, hogy jelenlegi eredményeink is alátámasztják a terület mélyebb kutatásának szükségességét.

Irodalomjegyzék

[1] D. Kempe, J. Kleinberg, E. Tardos. Maximizing the Spread of Influence through a Social Network. KDD ’03: Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 137-146. (2003)

[2] P. Sambaturu, B. Adhikari, B. A. Prakash, S. Venkatramanan, A. Vullikanti. Designing Effective and Practical Interventions to Contain Epidemics. AAMAS ’20: Proceedings of the 19th International Conference on Autonomous Agents and MultiAgent Systems, pp. 1187-1195. (2020)

[3] L. Hajdu, M. Krész. The Influence Monitoring Problem. SOR ’21: Proceedings of the 16th International Symposium on Operational Research, pp. 551-556. (2021)

[4] P. Domingos, M. Richardson. Mining the Network Value of Customers. KDD ’01: Proceedings of the 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 57-66. (2001)

[5] M. Granovetter. Threshold Models of Collective Behavior. American Journal of Sociology, Volume 83, Issue 6, pp. 1420-1443. (1978)

[6] M. Vass. Diffúziós folyamatok blokkolása és hálózati folyammodellek. Szegedi Tudományegyetem 2023. évi őszi Tudományos Diákköri Konferencia, Informatikai Szekció (2023)

[7] P. Hajnal. Gráfelmélet - Polygon jegyzet (2003)

[8] A. Bóta, M. Krész, A. Pluhár. Approximations of the Generalized Cascade Model. Acta Cybernetica, Volume 21, Issue 1, pp. 37-51. (2013)

[9] M. Marathe, A. Vullikanti. Computational Epidemiology. Communications of the ACM, Volume 56, Issue 7, pp. 88-96. (2013)

[10] W. Chen, C. Wang, Y. Wang. Scalable Influence Maximization for Prevalent Viral Marketing in Large-Scale Social Networks. KDD ’10: Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1029-1038. (2010)

[11] E. Bajalinov, B. Imreh. Operációkutatás - Polygon jegyzet (2001)

[12] L. Hajdu, M. Krész, A. Bóta. Evaluating the Role of Community Detection in Improving Influence Maximization Heuristics. Social Network Analysis and Mining, Volume 11, Issue 1, Article 91 (2021)

[13] A. Lancichinetti, S. Fortunato. Benchmarks for Testing Community Detection Algorithms on Directed and Weighted Graphs with Overlapping Communities. Physical Review E, Volume 80, Issue 1, 016118 (2009)

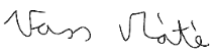
Nyilatkozat

Alulírott Vass Máté, programtervező informatikus MSc szakos hallgató kijelentem, hogy szakdolgozatomat a Szegedi Tudományegyetem, Informatikai Intézet Számítógépes Optimalizálás Tanszékén készítettem, programtervező informatikus MSc diploma megszerzése érdekében.

Kijelentem, hogy a diplomamunkámat más szakon korábban nem védtem meg, saját munkám eredménye, és csak a hivatkozott forrásokat (szakirodalom, eszközök stb.) használtam fel.

Tudomásul veszem, hogy jelen dolgozatot a Szegedi Tudományegyetem Informatikai Intézet könyvtárában, a helyben olvasható könyvek között helyezik el.

Dátum: 2023.12.16.



Aláírás

Köszönetnyilvánítás

Ezen kutatás az Európai Unió támogatásával valósult meg, az RRF-2.3.1-21-2022-00004 azonosítójú, Mesterséges Intelligencia Nemzeti Laboratórium projekt keretében.

Elektronikus mellékletek

[A] GitHub repository:
https://github.com/material1999/Diplomamunka_2023/
(Utolsó megtekintés: 2023.12.16.)