A Filter-Placement probléma és felhasználása az információ multiplicitás csökkentésére

Erdős Dóra, Vatche Ishakian, Andrei Lapets, Evimaria Terzi, Azer Bestavros

The Filter-Placement Problem and its Application to Minimizing Information Multiplicity c. cikk feldolgozása

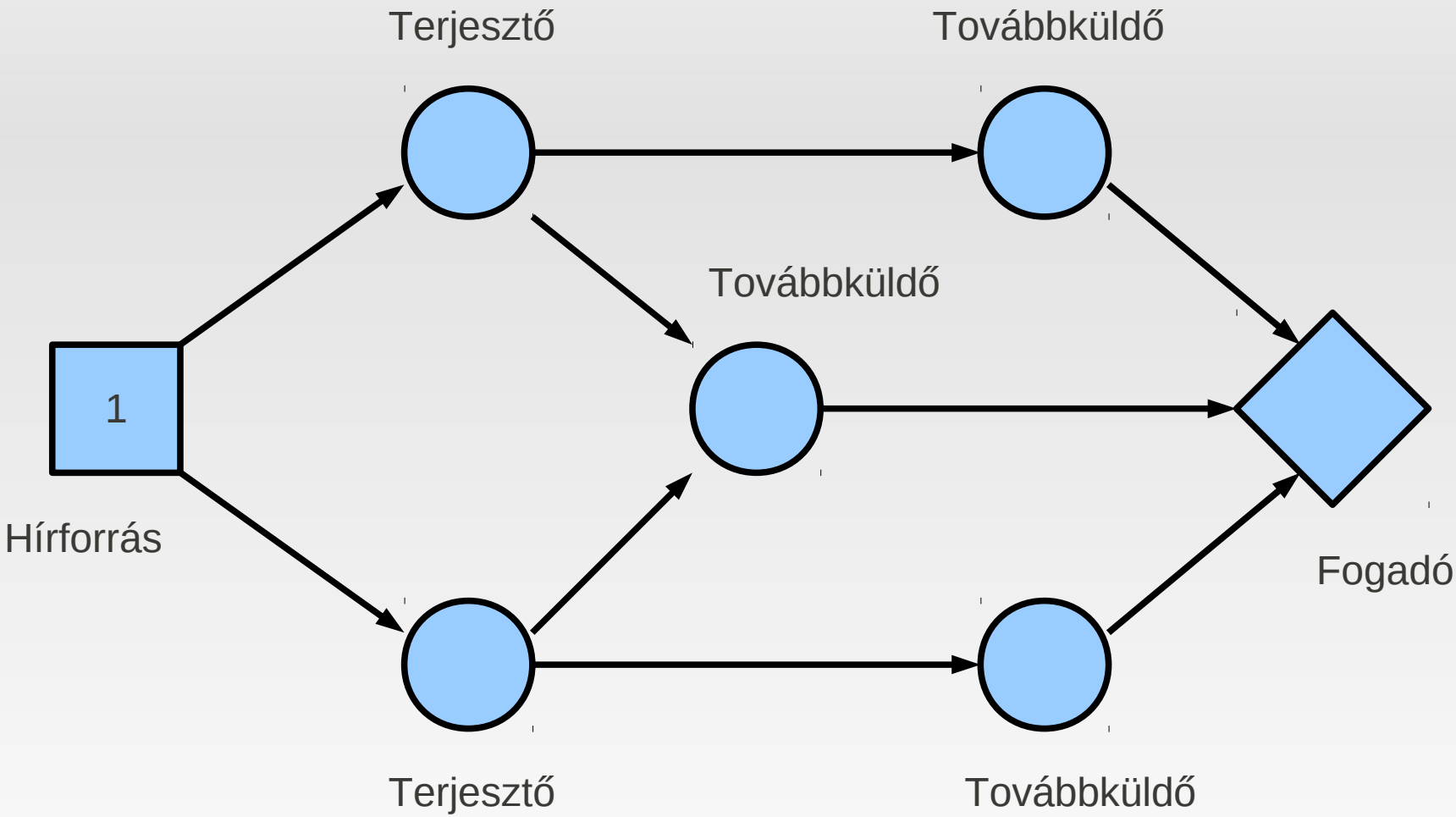
Bevezetés

- Egyes információs hálózatokban az adatok koordinálatlanul terjednek
  - Szociális hálók állapotfrissítései
  - Ad-hoc hálózatok útvonalfrissítései
- Az egyes csúcsok minden kapott információt továbbküldenek szomszédaiknak, még akkor is, ha az már rendelkezik vele
- Ez a jelenség redundáns adatok küldésével terheli a rendszert és csökkenti annak használhatóságát

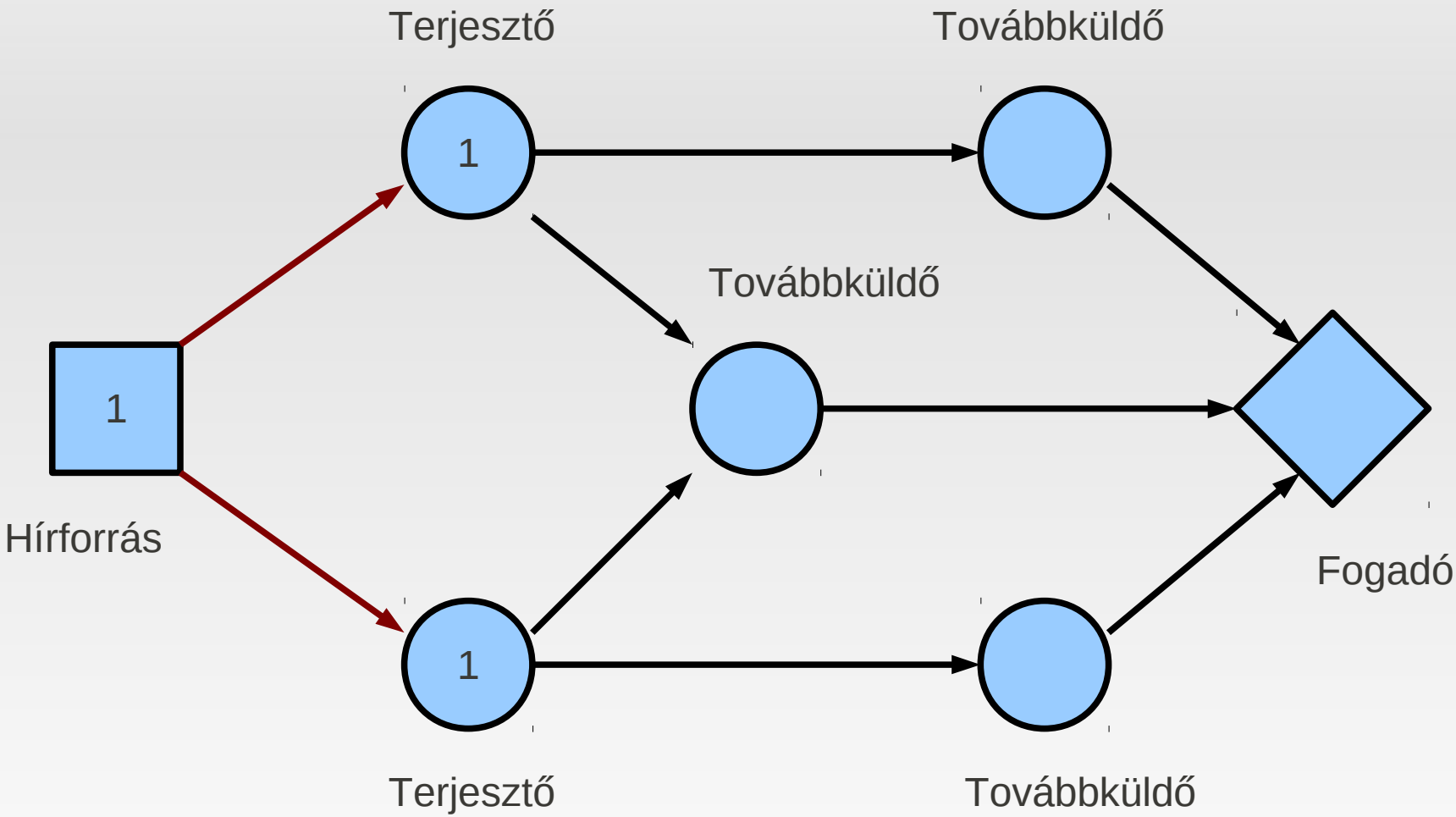
Probléma felvetés

- Jó lenne csökkenteni a redundáns információ mértékét
- Ezen célból kijelölhetünk speciális szerepű csúcsokat
  - Ezek szűrő szerepet látnak el
  - Nem küldenek tovább gondolkodás nélkül mindent, amit kaptak

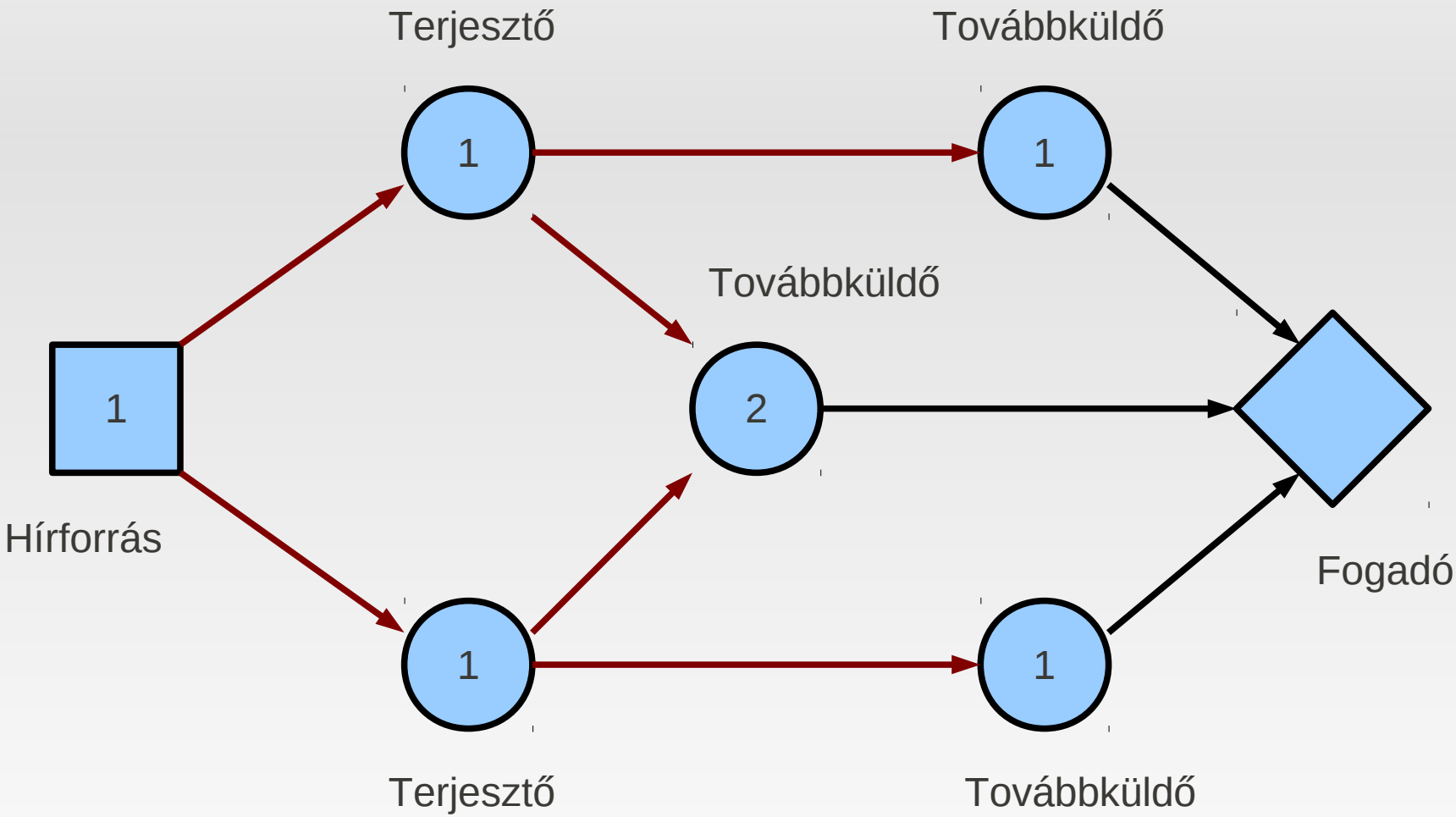
Az információ multiplicitás szemléltetése



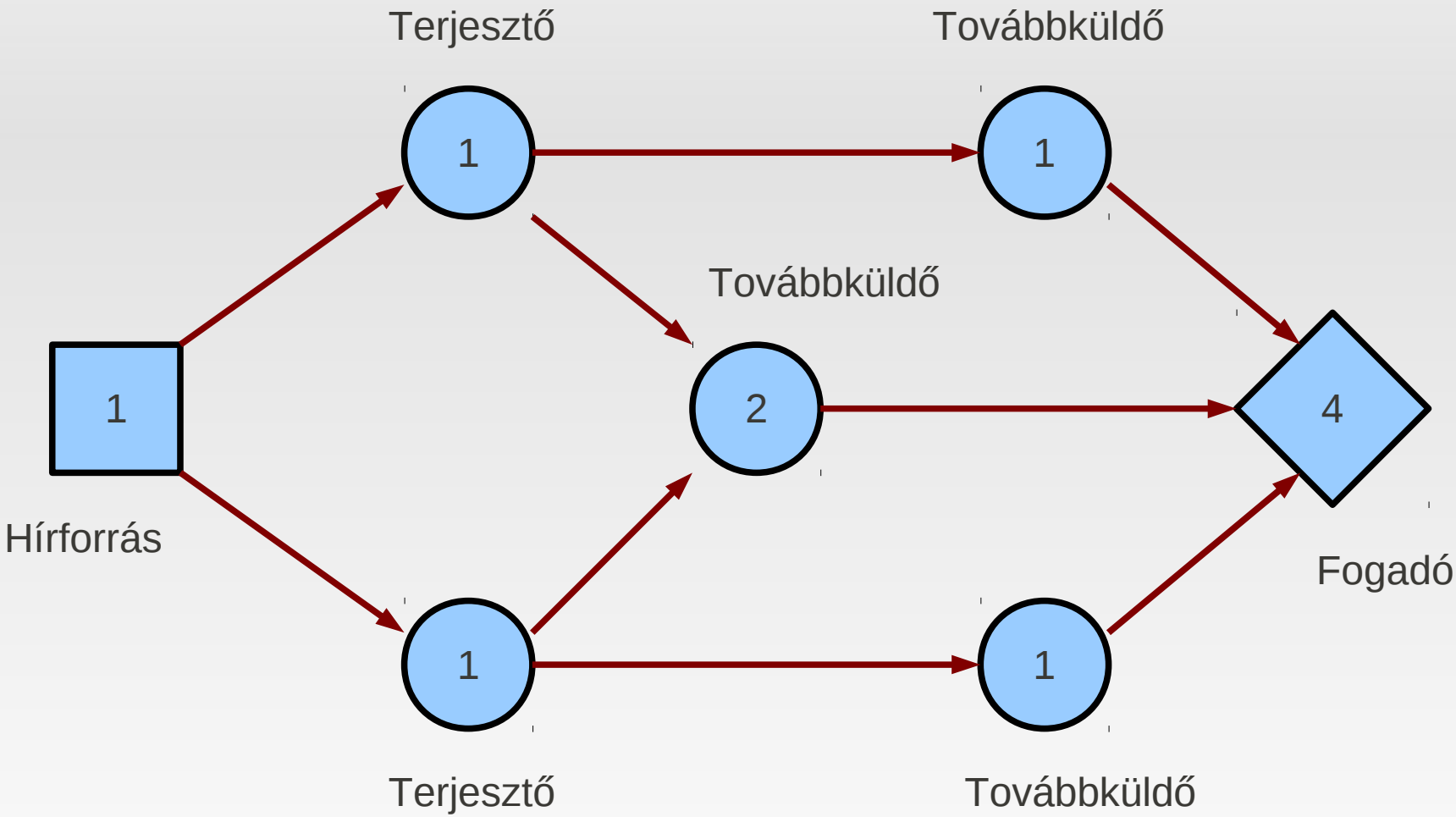
Az információ multiplicitás szemléltetése



Az információ multiplicitás szemléltetése



Az információ multiplicitás szemléltetése



Kapcsolódó munkák

- Az első cikk, ami konkrétan ezzel a problémával foglalkozik
- Hasonló munkák
  - Centralitás a hálózatokban
  - Szociális hálók esetén
    - a befolyásoló erővel rendelkező pontok meghatározása
    - kártákony pontok meghatározása
  - Ad-hoc hálózatokban elárasztásos terjesztési technikák
- Nem megfelelőek erre a célra

Filter-Placement probléma

- Rendszermodell
  - Egymáshoz kapcsolódó entitások halmaza
  - Az entitások minden információt, amit kapnak továbbküldenek a szomszédaiknak
- Reprezentáció
  - G(V,E) irányított gráf, melyet c-gráfnak nevezünk
  - A hálózat résztvevői a csúcsok
  - Két csúcs között akkor van irányított él, ha küldhetnek egymásnak, az irány a kommunikáció iránya
  - Néhány csúcs állít elő új információt, ezek a források
  - Ha egy csúcs információt kap, továbbküldi azt a gyerekeinek

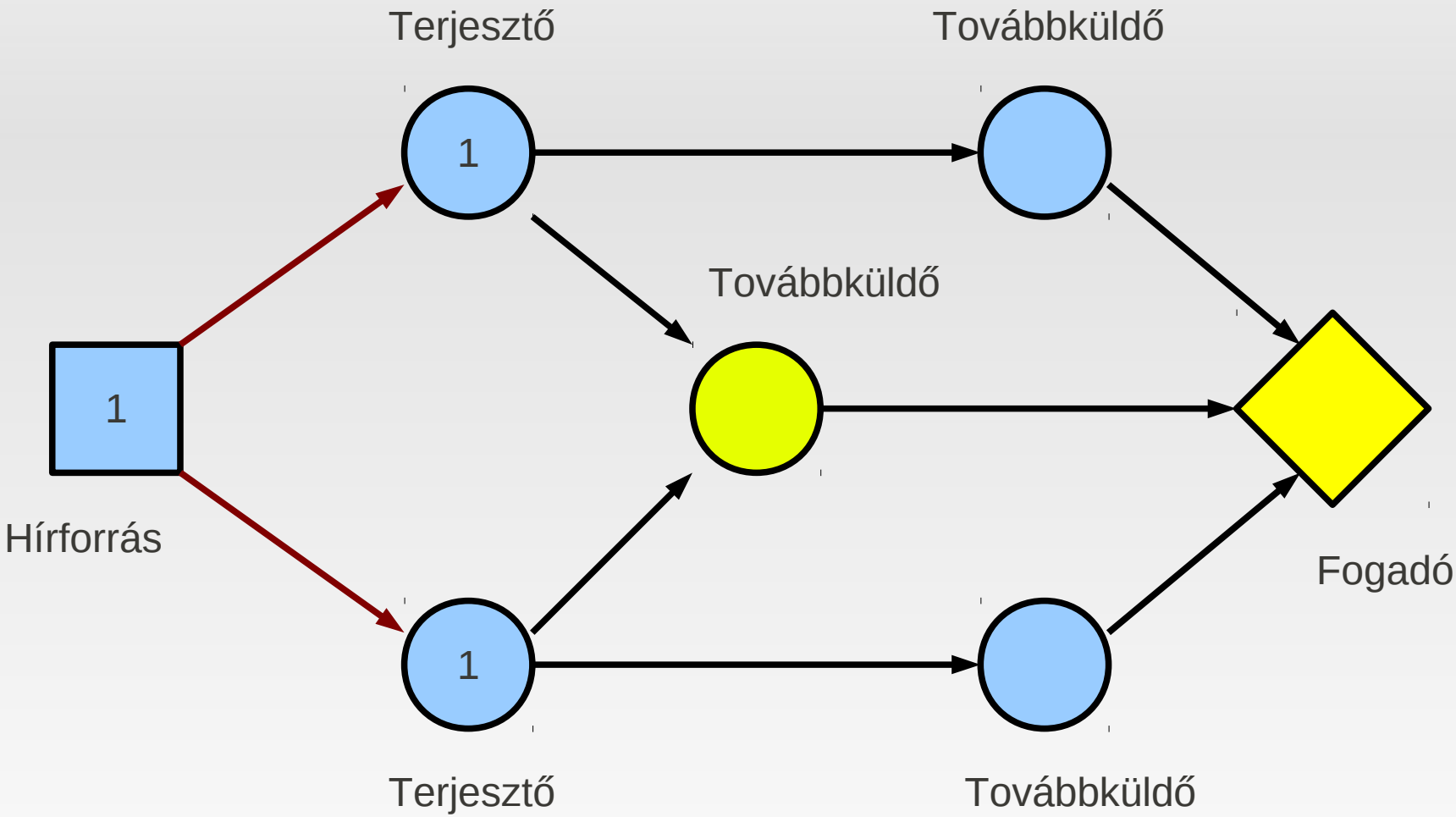
Filter-Placement probléma

- Néhány csúcs speciális szerepet lát el, ezek a szűrők
  - A szűrést tekinthetjük függvénynek, mely bemenetként egy információhalmazt kap, kimenetként pedig egy olyan információhalmazt ad vissza, melyben a duplikátumokat csökkentjük
  - Pontos feladata függ a felhasználástól
  - A bemutatásra kerülő módszerekben a szűrő minden duplikátumot eltávolít

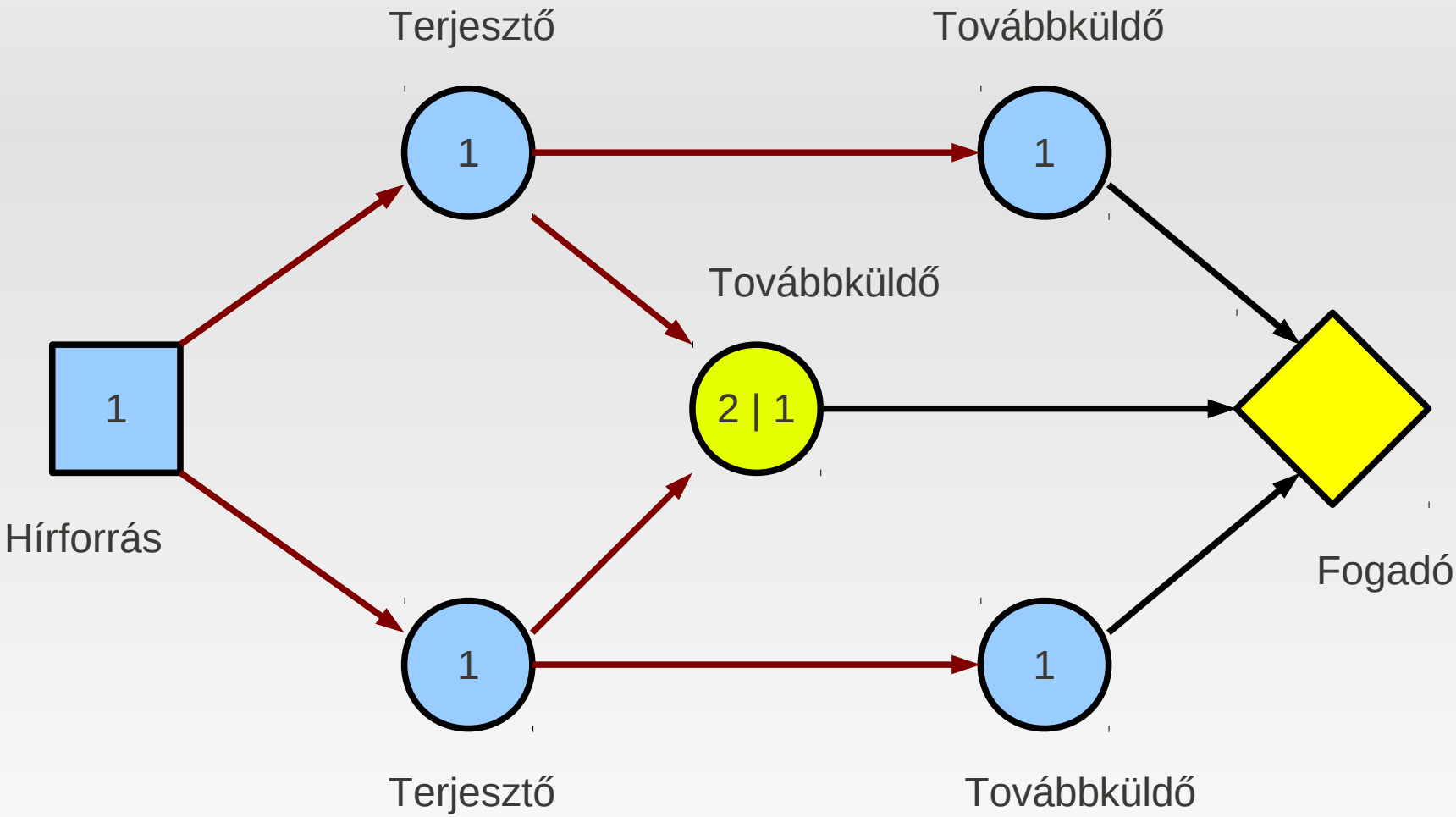
Filter-Placement probléma

- Objektív függvény
  - Egy tetszőleges v csúcsra $\phi(0,v)$ jelentse az adatok számát, melyet v akkor kap, ha nincs egyetlen szűrő sem a hálózatban
  - Jelentse továbbá $\phi(A,v)$ az adatok számát melyet v kap, ha A a szűrő feladatot ellátó csúcsok halmaza
  - Általánosítás csúcsok egy részhalmazára
- Filter-Placement probléma
  - Adott egy G(V,E) irányított gráf és egy k egész szám, keressük a csúcsok azon A részhalmazát, melynek számossága maximum k és az $F(A) = \phi(0,V) – \phi(A,V)$ függvény értéke maximális

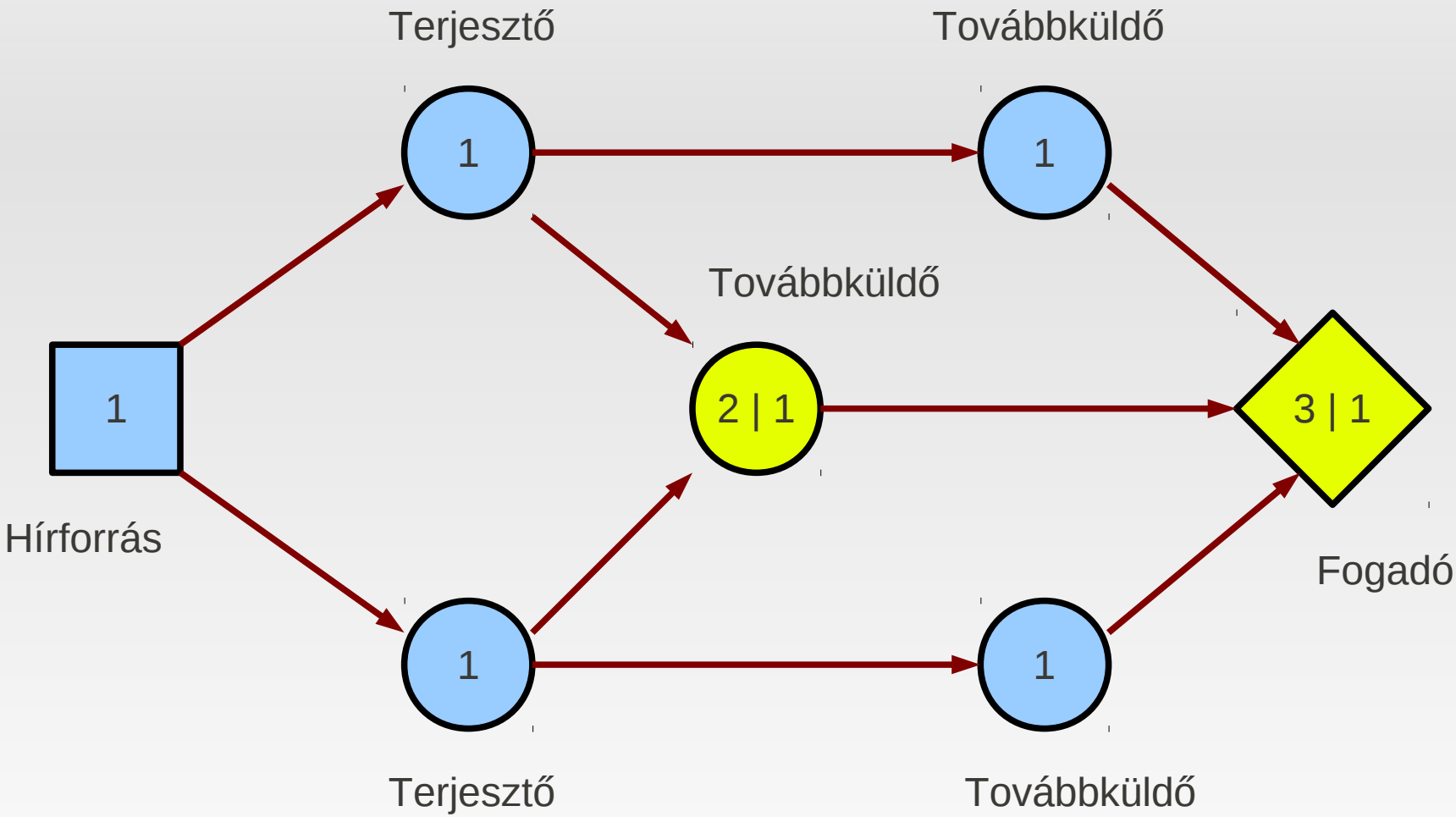
A szűrő csúcsok működésének szemléltetése



A szűrő csúcsok működésének szemléltetése



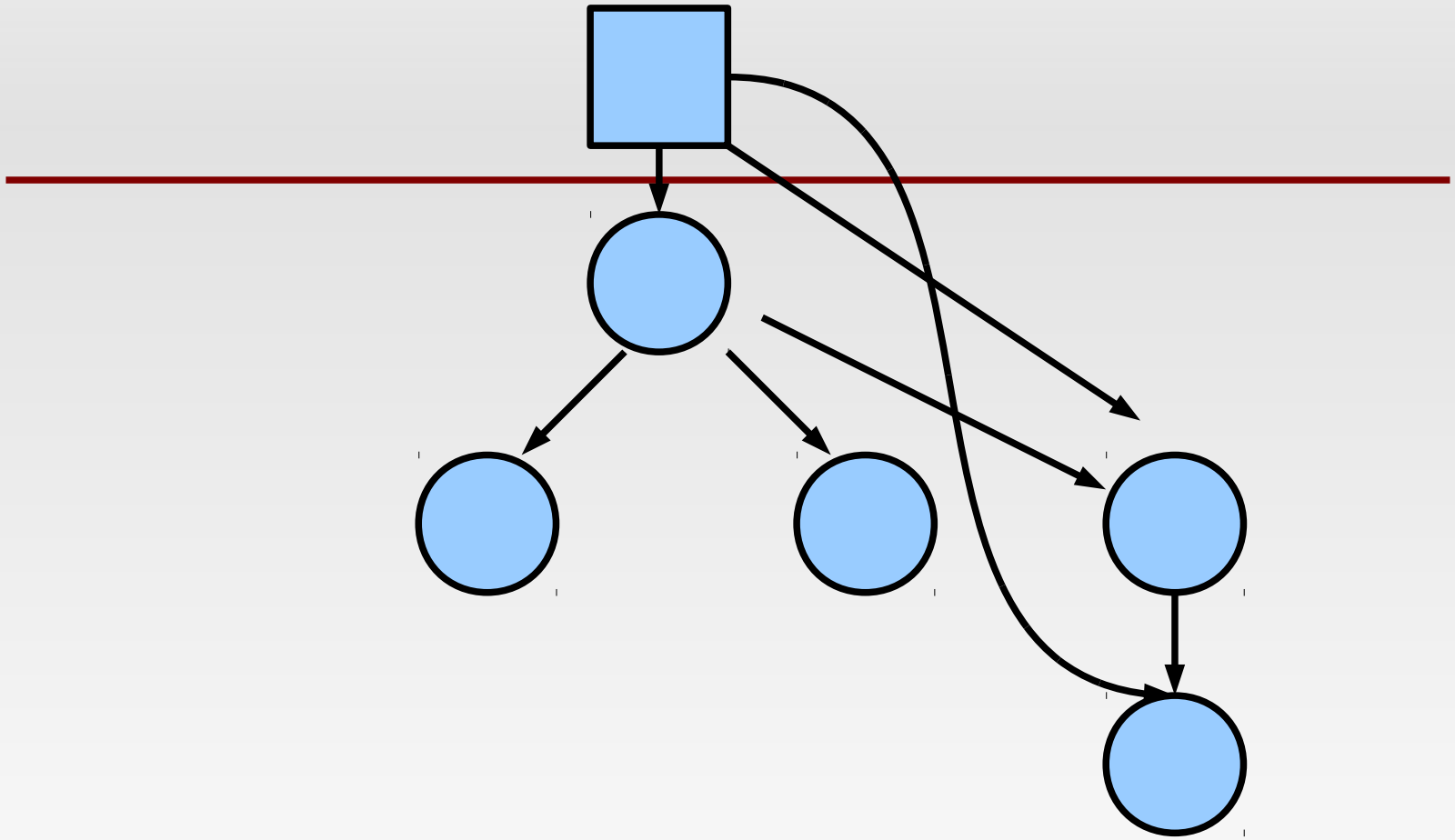
A szűrő csúcsok működésének szemléltetése



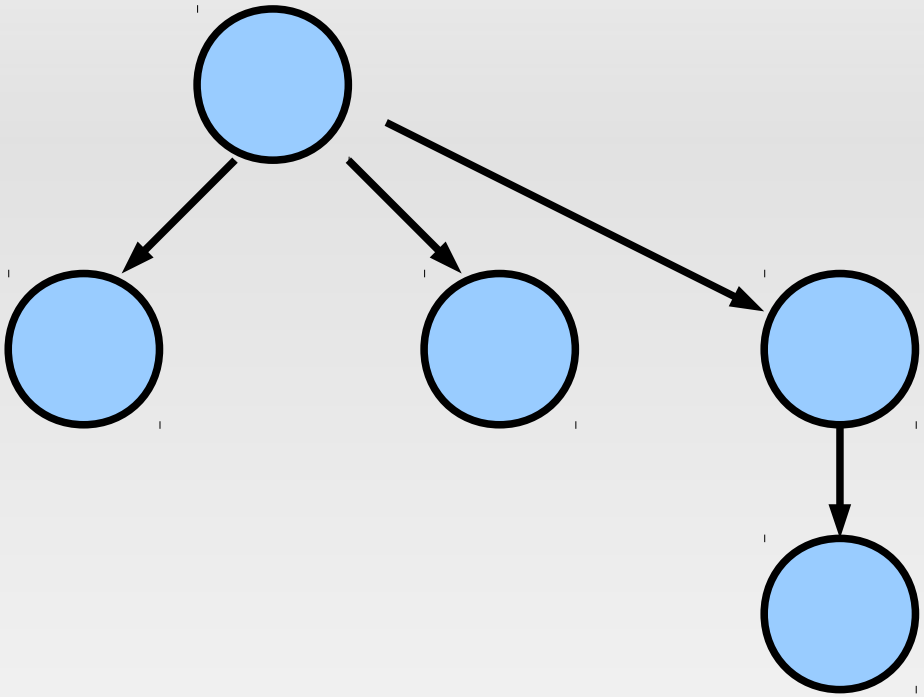
Filter-Placement c-fákon értelmezve

- Polinomiális idő alatt megoldható
- Legyen a G gráf egy c-fa, ha a forrás csúcsot eltávolítva fa
- Alakítsuk át a c-fát bináris fává
  - Ha egy csúcs foka max. 2 akkor nincs dolgunk
  - Különben vegyük a csúcs gyerekeinek egy tetszőleges sorrendjét, majd iterálva a gyerek csúcsokon az éppen jövő csúcs legyen a szülő bal-gyereke és hozzunk létre egy új csúcsot, mely a jobb-gyerek lesz, a megmaradó csúcsokat pedig helyezzük ez alá és így tovább

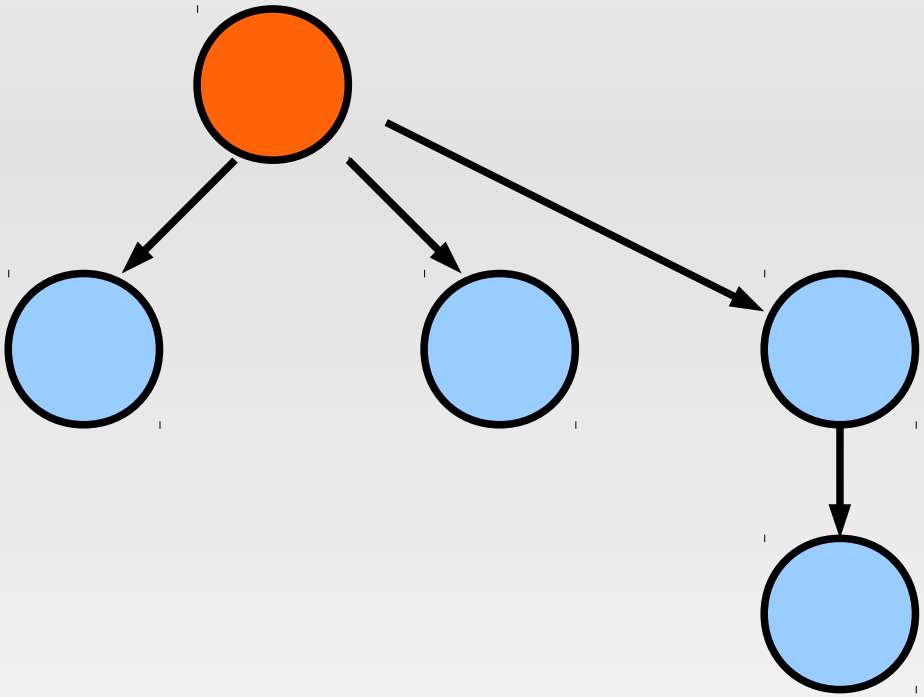
Filter-Placement c-fákon értelmezve



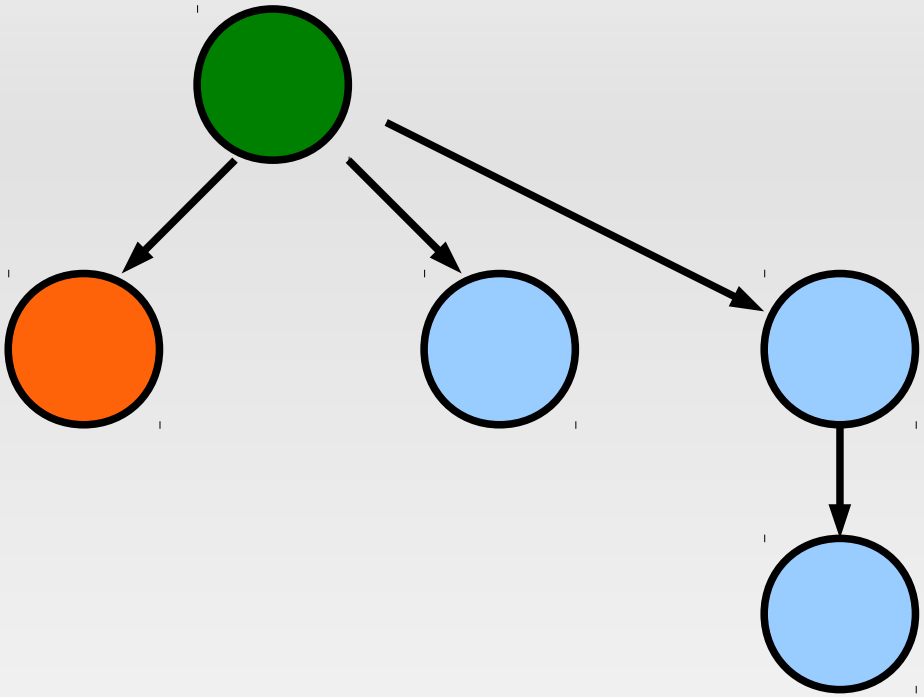
Filter-Placement c-fákon értelmezve



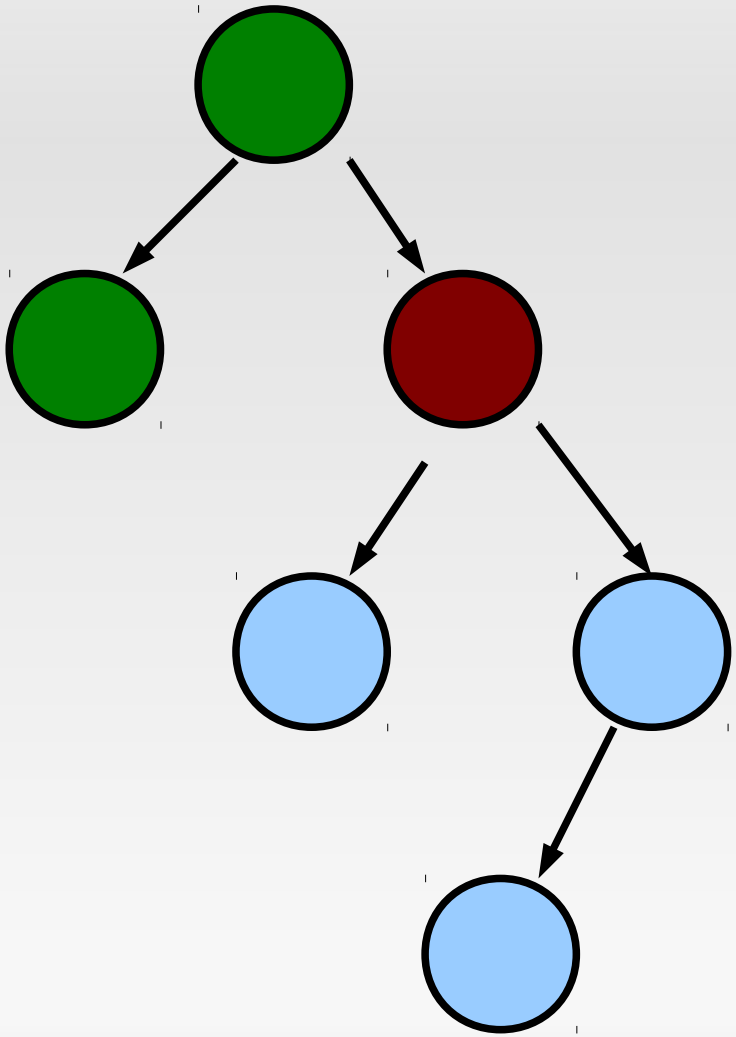
Filter-Placement c-fákon értelmezve



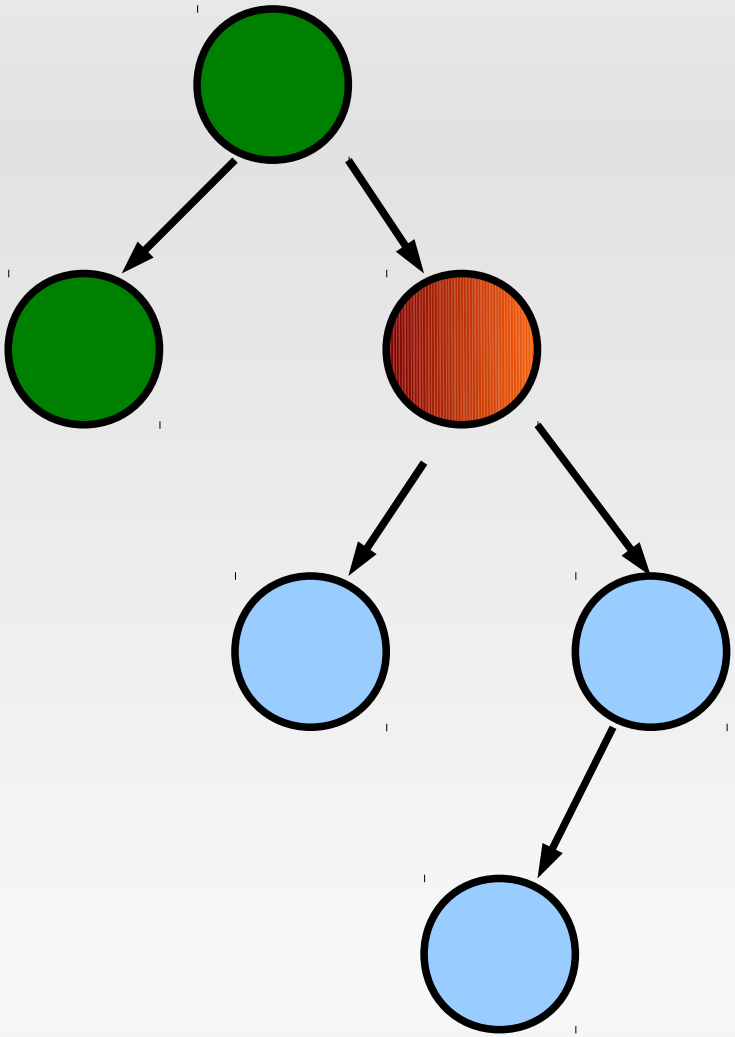
Filter-Placement c-fákon értelmezve



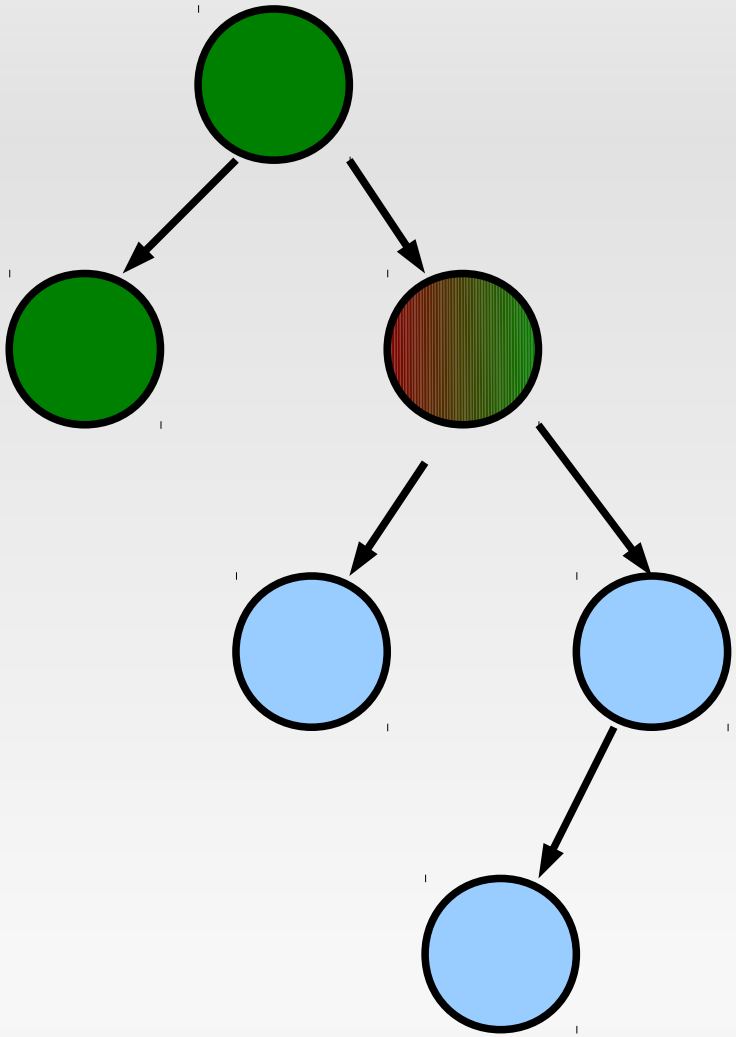
Filter-Placement c-fákon értelmezve



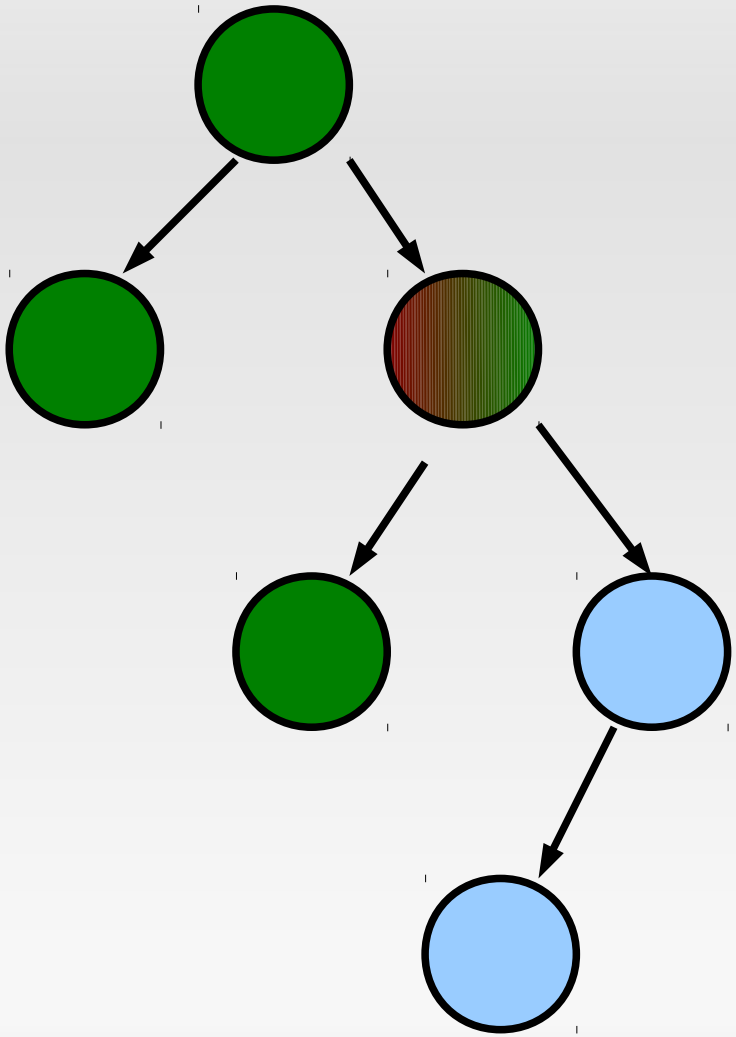
Filter-Placement c-fákon értelmezve



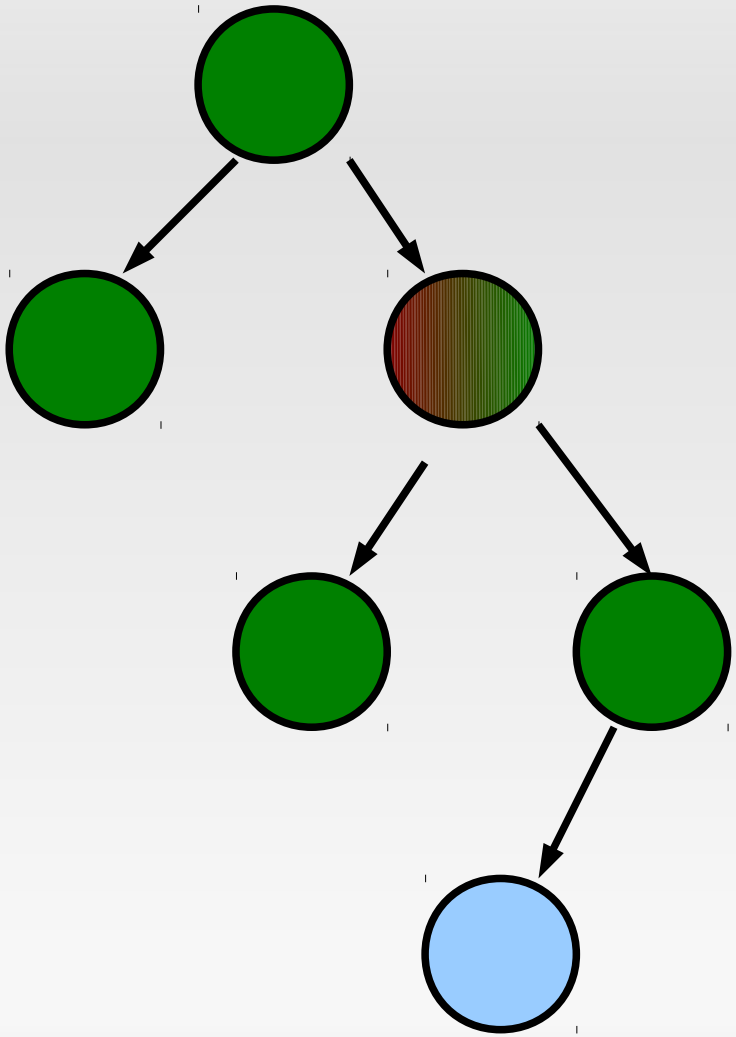
Filter-Placement c-fákon értelmezve



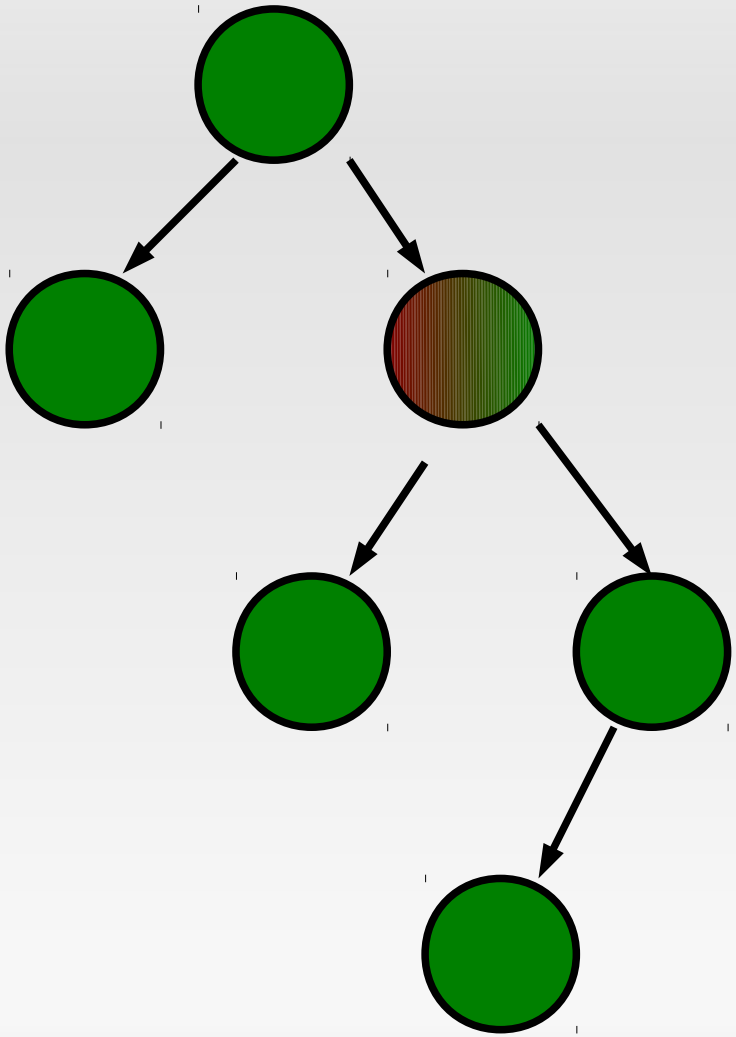
Filter-Placement c-fákon értelmezve



Filter-Placement c-fákon értelmezve



Filter-Placement c-fákon értelmezve



Filter-Placement c-fákon értelmezve

- Megkonstruáljuk az OPT algoritmust
  - Bemenetként kap egy v csúcsot és egy i számot
  - Kimenetként visszaadja az A szűrőcsúcsok halmazát, melynek számossága maximum i és az optimális szűrőhalmaz a v-ből induló részfára

$$\begin{aligned}\mathrm{OPT}(v,i,A) = \max\{ & \\ & \max_{j=0\ldots i}\{\mathrm{OPT}(v_l,j,A)+\mathrm{OPT}(v_r,i-j,A)\}, \\ \max_{j=0\ldots i-1}\{\mathrm{OPT}(v_l,j,A\cup\{v\})+\mathrm{OPT}(v_r,i-1-j,A\cup\{v\})\}&\}.\end{aligned}$$

Filter-Placement c-fákon értelmezve

- Lényege
  - Rekurzióval dolgozik, két részre bontható az alapján, hogy v-be helyezünk-e filtert
  - Az egész fára a gyökérből kell indítani az algoritmust
  - Nem zárja ki a felesleges szűrők elhelyezését

Filter-Placement DAG-on értelmezve

- Egy c-gráf DAG ha
  - Irányított
  - Körmentes
- A Filter-Placement probléma NP-teljes ezen esetben
- Több polinomiális idejű algoritmus, melyek folyamatosan optimálisabbak valamilyen szempontból

Filter-Placement DAG-on értelmezve Greedy_1 algoritmus

- Alapötlet
  - Minden csúcs a bemenő élein kap információt, melyek mindegyikét minden kimenő élén továbbítja
  - A továbbítandó adatra számolhatunk egy alsó korlátot ezt felhasználva minden csúcshoz úgy, hogy m(v) = bejövő élek * kimenő élek
- Lényege
  - Minden csúcsra kiszámolja a továbbküldendő adat alsó korlátját
  - Ezek közül kiválasztja a k darab legnagyobbat és ezen csúcsokba helyez el szűrőket

Filter-Placement DAG-on értelmezve Greedy_1 algoritmus

- Műveletigény szempontjából jó és egyszerű
- Egyes esetekben azonban hibásan működik

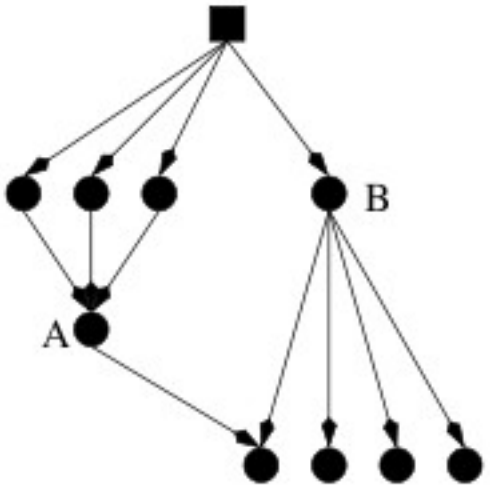


Figure 2: For $k = 1$ Greedy_1 places a filter in B while the optimal solution would be to place a filter in A.

Filter-Placement DAG-on értelmezve Greedy_All algoritmus

- Alapötlet
  - Minden csúcsra számoljuk ki, hogy egy információból hány másolatot fogünk küldeni miatta
  - Az információ a forrástól egy csúcsba bejár egy utat s-ből v-be. Erre számoljuk ki a különböző utak számát, hiszen ennyi másolatot fog kapni v az eredeti információról - Prefix(v)
  - Számoljuk ki továbbá, hogy mennyi duplikátum lesz a gráfban miután v továbbküldi, amit kapott - Suffix(v)
  - A másolatok száma a v-n áthaladás miatt a fentiekkel I(v) = (Prefix(v) – 1) * Suffix(v)

Filter-Placement DAG-on értelmezve Greedy_All algoritmus

- Lényege
  - Az algoritmus először kiválasztja a legnagyobb I értékkel rendelkező csúcsot, és rak oda egy szűrőt
  - Újraszámolja az I értéket minden csúcsra (úgy, hogy a kiválasztott a Prefix-et 1-re állítja)
  - Újra kiválasztja a legnagyobb I értékkel rendelkező csúcsot
  - Mindezt addig folytatja, amíg ki nem választ k csúcsot
- Műveletigénye $O(k * n * \log n)$

Filter-Placement DAG-on értelmezve Greedy_All algoritmus

- Optimálisan egy szűrő kiválasztására működik
- Szemléltetés a programmal
- Nagy adathalmazon (nagy gráfon) költségessége miatt lassan működik

Algorithm 1 `Greedy_All` algorithm

Input: DAG $G(V, E)$ and integer k.
Output: set of filters $A \subseteq V$ and $|A| \leq k$.

1: find topological order σ of nodes
2: **for** $i = 1 \ldots k$ **do**
3: **for** $j = 1 \ldots n$ **do**
4: compute $I(v_j)$
5: $A \leftarrow \text{argmax}_{v \in V} I(v)$
6: **return** A

Filter-Placement DAG-on értelmezve Greedy_Max algoritmus

- A Greedy_All gyorsítása
- Lényege
  - Csak egyszer számolja ki az I értéket minden csúcsra
  - Kiválasztja a k darab legnagyobbat
  - Műveletigénye csak O(n * |E|)
  - Jól közelíti a Greedy_All eredményeit

Filter-Placement DAG-on értelmezve Greedy_L algoritmus

- A Greedy_All gyorsítása
- Lényege
  - Az I érték helyett egy egyszerűbb I'-t számol csak ki minden csúcsra, melyben a Suffix helyett a kimenő élek számát használjuk (mennyi adatot küld az adott csúcs a közvetlen gyerekeinek)
  - Az I'-t minden iterációban újra számoljuk, mint a Greedy_All esetén az I-t
  - Műveletigénye O(k * |E|)

Filter-Placement általános gráfon értelmezve

- Visszavezetjük a DAG-on értelmezett Filter-Placement problémára
- Ehhez meg kell határozni a maximális körmentes részgráfot
  - Acyclic algoritmus
  - Mélységi bejárást használ
- Ezek után a korábban bemutatott stratégiák felhasználhatóak

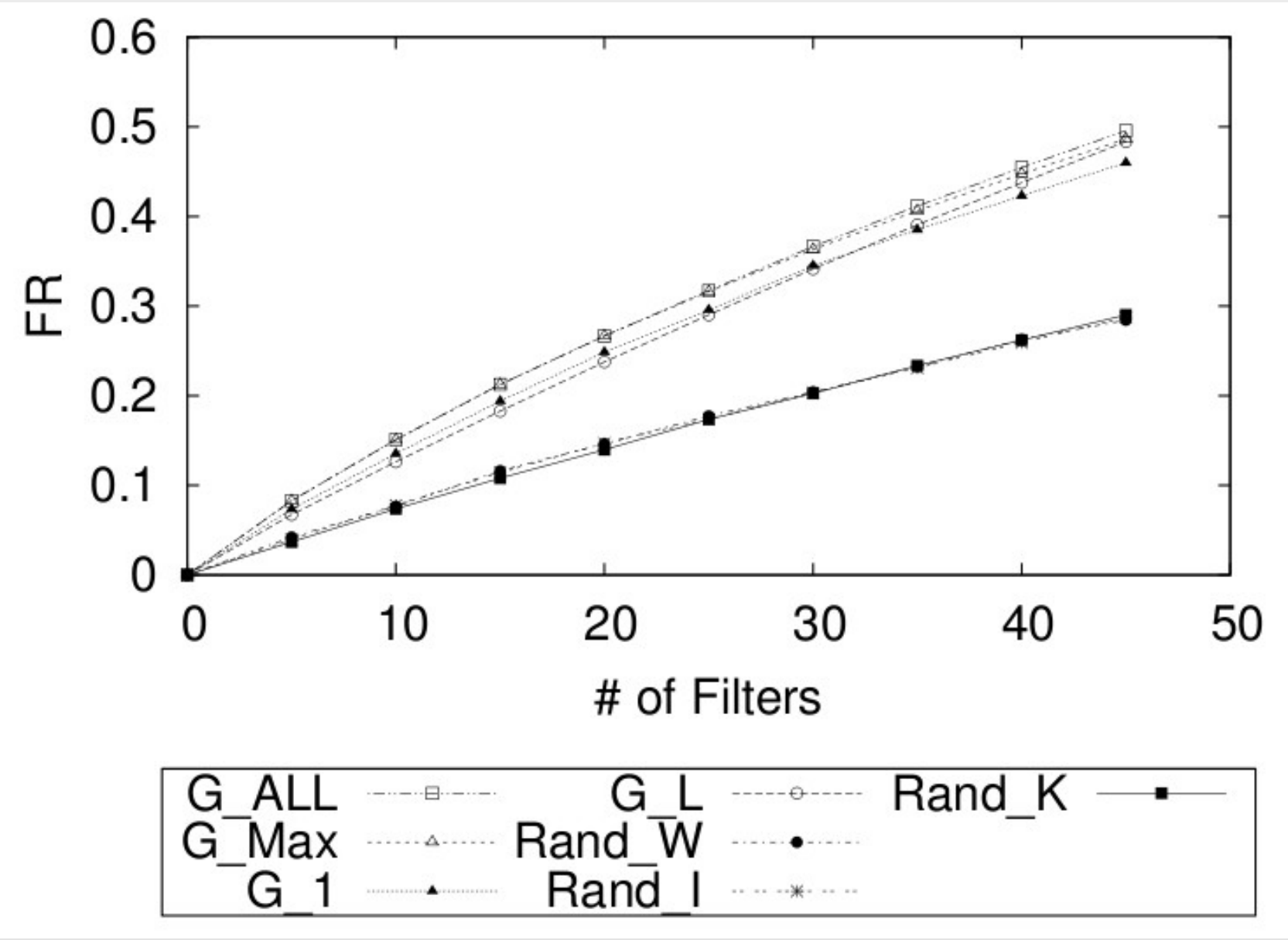
Eredmények

- Generált és valós adatok
- A mérés tárgya
  - Filter Ratio (FR)
  - Megegyezik a szűrők mellett küldött adatok és a szűrő nélkül küldött adatok számának hányadosával (erre az objektív függvényt használhatjuk)
- Még néhány stratégiát bevezet a mérésekhez
  - Random_k – véletlenszerűen választ k csúcsot
  - Random_Independent – bármely csúcs k/n valószínűséggel lesz szűrő
  - Random_Weighted – súlyt rendeleünk a csúcsokhoz a gyerekeik és a bejövő élek száma szerint, majd ezt szorozzuk a fentivel és ilyen valószínűséggel lesz az adott csúcs szűrő

Eredmények – generált adatokon

- Kb. 1000 csúcs
- Meghatározott szintek a gráfban
- Kb. 32000 él
- A szűrők számának növekedésével jól nő a FR értéke

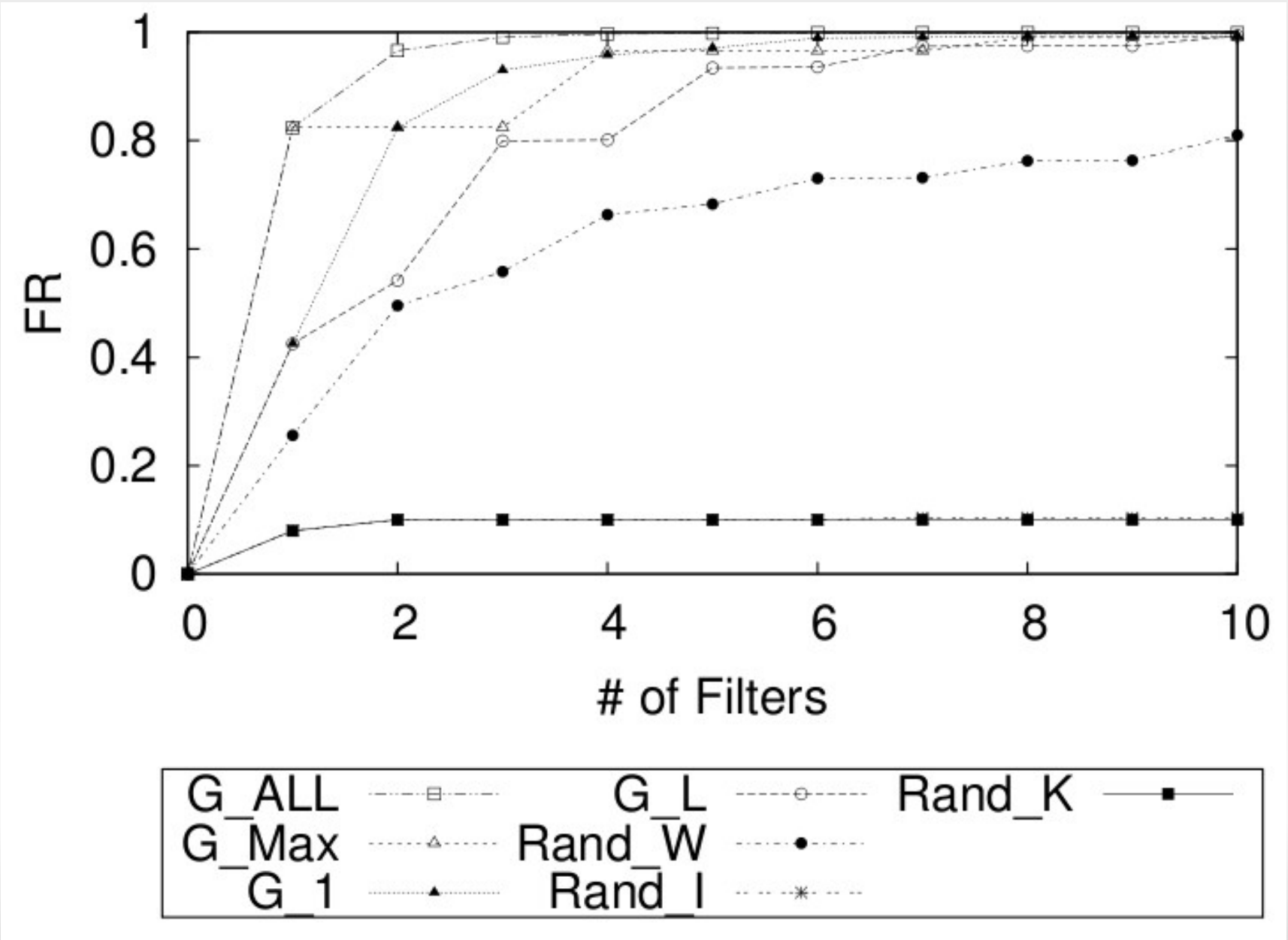
Eredmények – generált adatokon



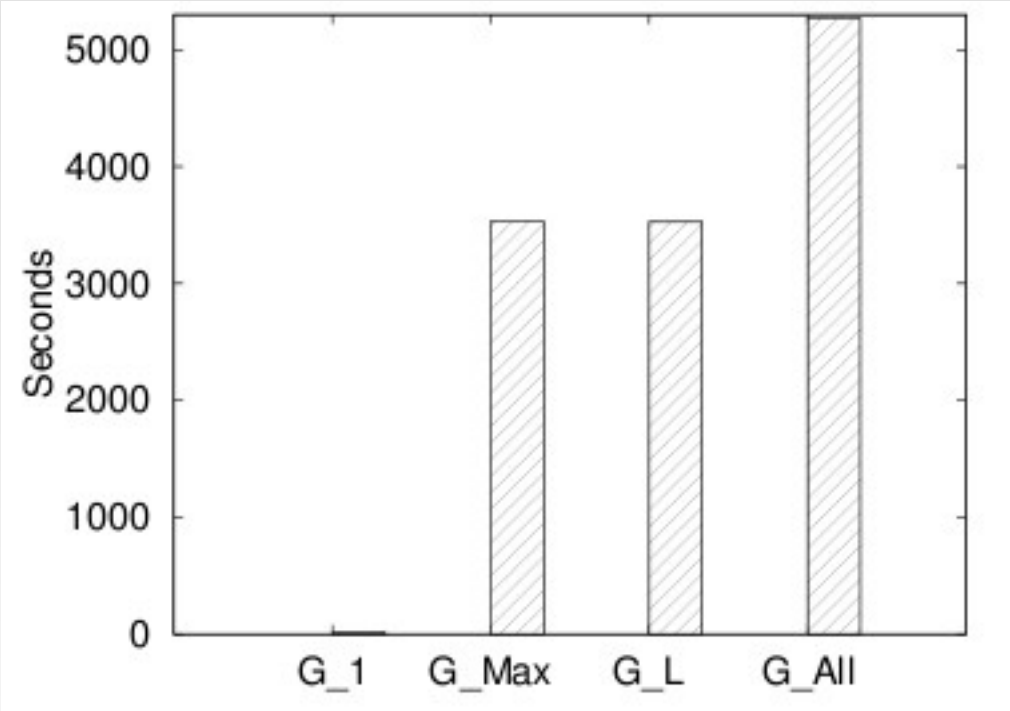
Eredmények – Twitter adatokon

- Adatgyűjtés a Kwak felhasználásával
- Kb. 41 millió felhasználói profil
- UserID-ket és köztük lévő kapcsolatokat tartalmaz (irány a felhasználótól a follower felé)
- Ennek egy részén zajlott csak a mérés, kb. 90000 csúcs és 120000 él bevonásával
- Látható, hogy
  - A Greedy_All 6 szűrő elhelyezésével minden redundanciát kiszűr
  - A többi Greedy algoritmus is 10 szűrő elhelyezésével eléri ezt az eredményt
  - Leglassabban a Greedy_L éri el, mert ez a forrástól messzebbi csúcsokat választ előbb

Eredmények – Twitter adatokon



Eredmények – Twitter adatokon



- Futási idők
- Greedy_1 a leggyorsabb
- Greedy_All a leglassabb (83 perc)
- A Greedy_Max és a Greedy_L kb. 60 perc alatt lefutott

Eredmények értékelése

- A Greedy_All algoritmus nagyon lassú
- Valós adatokon a többi Greedy algoritmus is jól szerepelt a redundancia kiszűrésében
- Jól használható eszközök a Filter-Placement probléma megoldására

Összefoglalás

- Bemutattuk a Filter-Placement problémát és felhasználási lehetőségeit
  - Redundáns információk küldésének csökkentése a hálózatban
- Különböző típusú gráfokon adott algoritmusok
  - C-fa, DAG, általános gráf
- Mérési eredmények valós és generált adatokon

Kérdések