

Hálózati Algoritmusok

Adatfolyam és véletlen elhelyezés elmélete

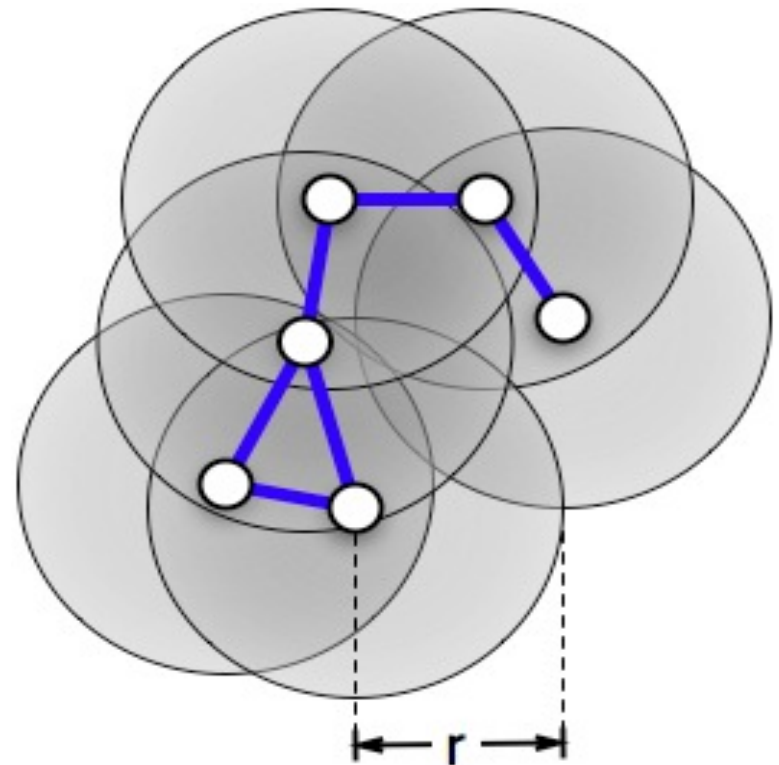
Unit Disk Graph

- Motiváció:
 - A fogadott jel erőssége d távolságban arányos $d^{-\gamma}$ -val, ahol γ a *path loss exponent*
 - Kapcsolat csak akkor van, ha a jel-zaj arány nagyobb, mint egy küszöb

- Definíció
 - Adott pontok egy véges halmaza $V \subset \mathbf{R}^2$ vagy $V \subset \mathbf{R}^3$,
 - akkor a V ponthalmazhoz a *Unit Disk Graph* (UDG) r sugárral $G=(V,E)$ a következő irányítatlan élhalmaz által definiált:

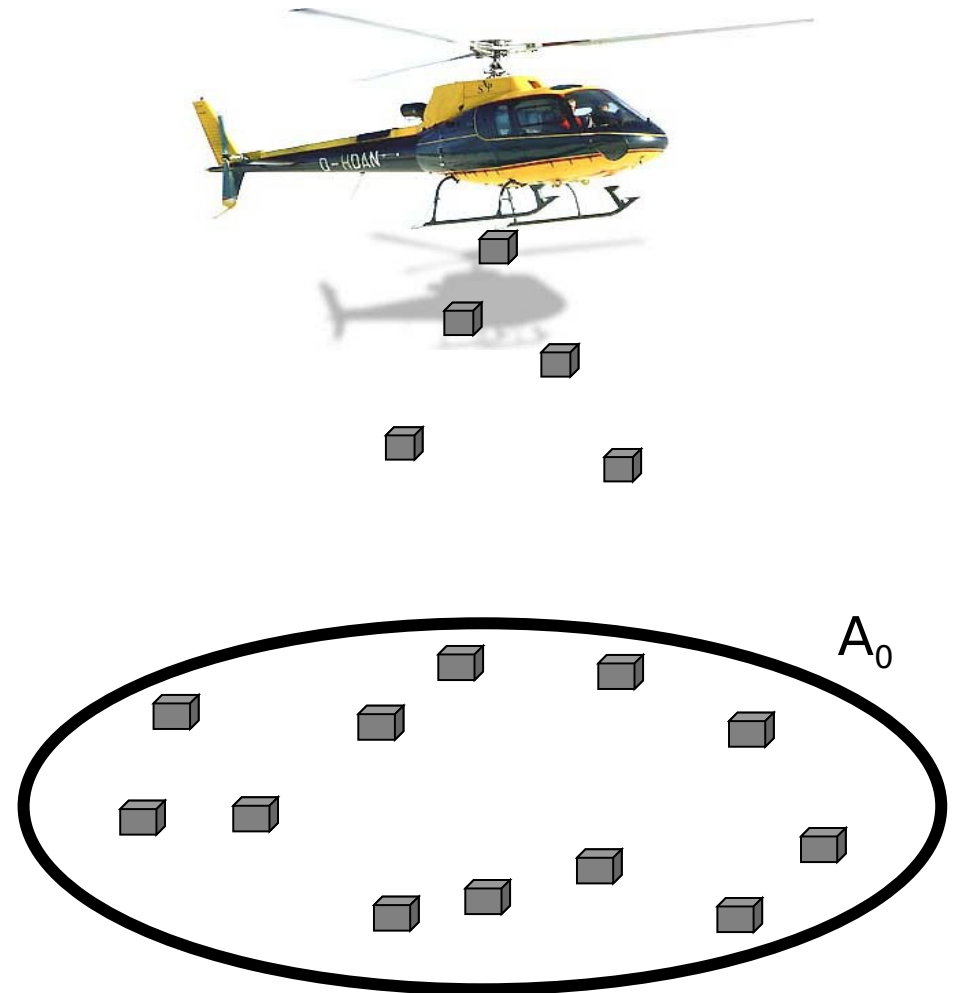
$$E = \{ \{u, v\} \mid \|u, v\|_2 \leq r \}$$

- ahol $\|u, v\|_2$ az Euklideszi távolság:
$$\|u, v\|_2 = \sqrt{(u_x - v_x)^2 + (u_y - v_y)^2}$$



Véletlen elhelyezés model

- Motiváció
 - Repülőről szórjuk le a csomópontokat
- Definíció
 - Pontok egy halmaza véletlenül van elhelyezve egy A_0 területen, ha minden pozíció egyforma valószínű, azaz
 - A sűrűségi függvény (pdf) $f(x)$ konstans



A véletlen elhelyezés tulajdonságai

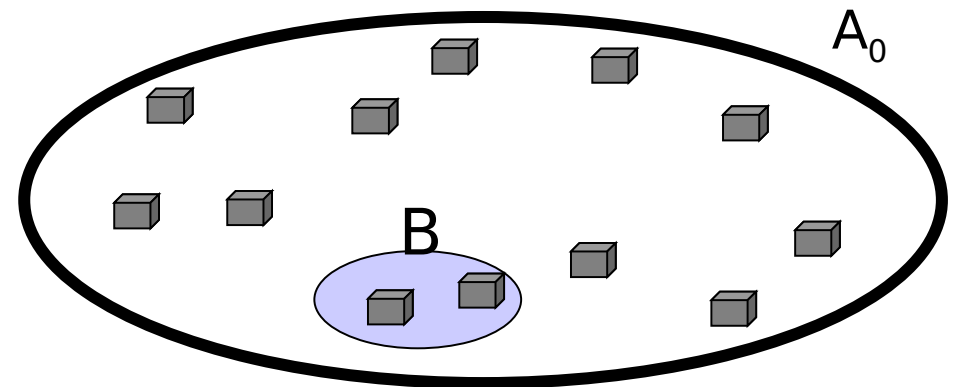
- Annak a valószínűsége, hogy egy csomópont egy adott B területre esik A_0 -on belül:

- $\Pr[\text{csomópont B-be esik}] = |B| / |A_0|$
- ahol $|B|$ B területe

- Lemma

- Legyen $p = |B|/|A_0|$. Annak a valószínűsége, hogy n csomópont közül k esik B-be:

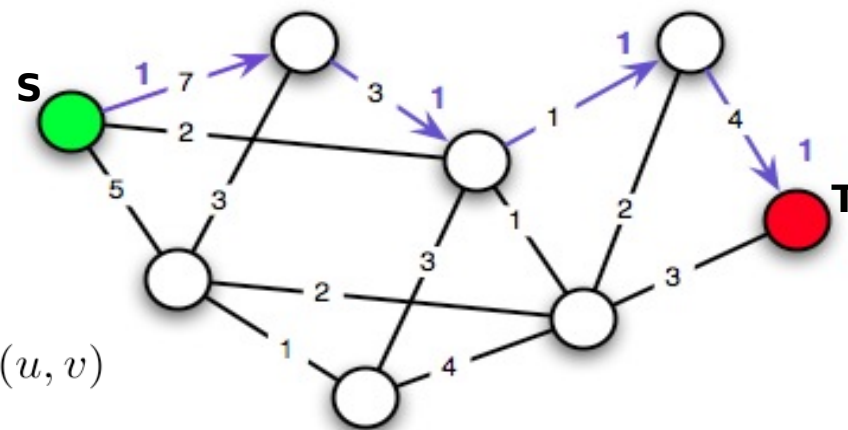
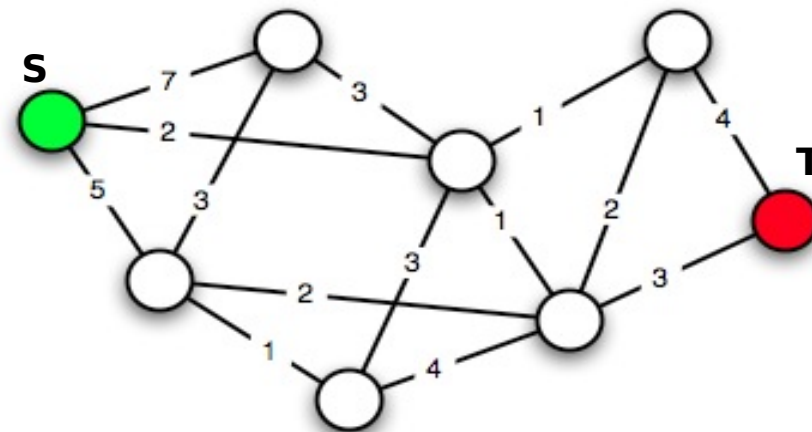
$$\Pr[n\text{-ből } k \text{ esik } B\text{-be}] = \binom{n}{k} p^k (1 - p)^{n-k}$$



Adatfolyamok hálózatokban

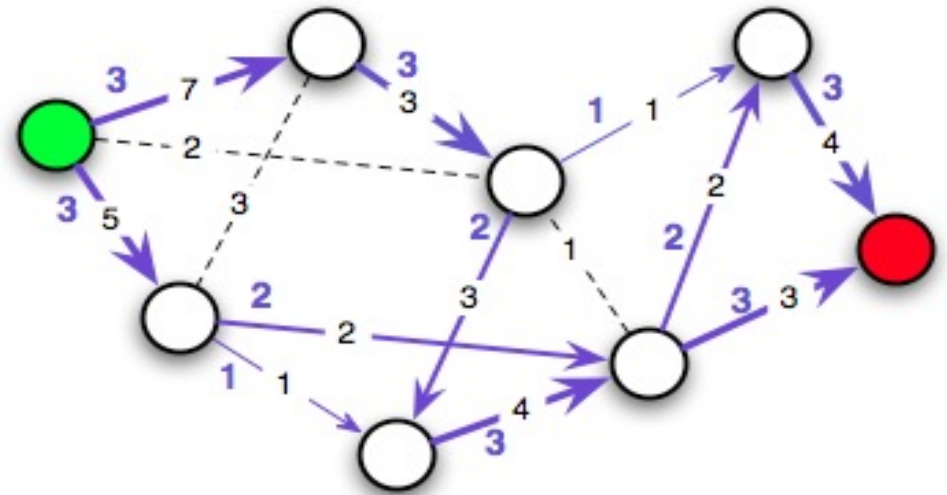
- Motiváció:
 - Optimizáljuk az adatfolyamot a forrástól a célig
- Definíció:
 - **max. folyam probléma**
 - Adott
 - gráf $G=(V,E)$
 - kapacitás függvény $w: E \rightarrow \mathbf{R}^+_0$,
 - forrás halmaz S és cél halmaz T
 - Keressünk egy maximum folyamot S -től T -ig
 - Egy folyam függvény $f: E \rightarrow \mathbf{R}^+_0$, ahol
 - minden $e \in E: f(e) \leq w(e)$
 - minden $u,v \in V: f(u,v) \geq 0$
 - $\forall u \in V \setminus (S \cup T) \quad \sum_{v \in V} f(v,u) = \sum_{v \in V} f(u,v)$
 - A folyam mérete:

$$\sum_{u \in S} \sum_{v \in V} f(u,v)$$



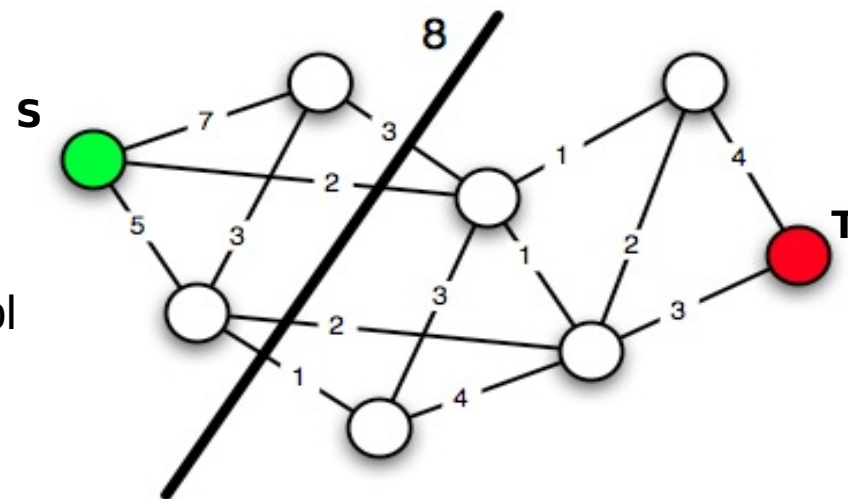
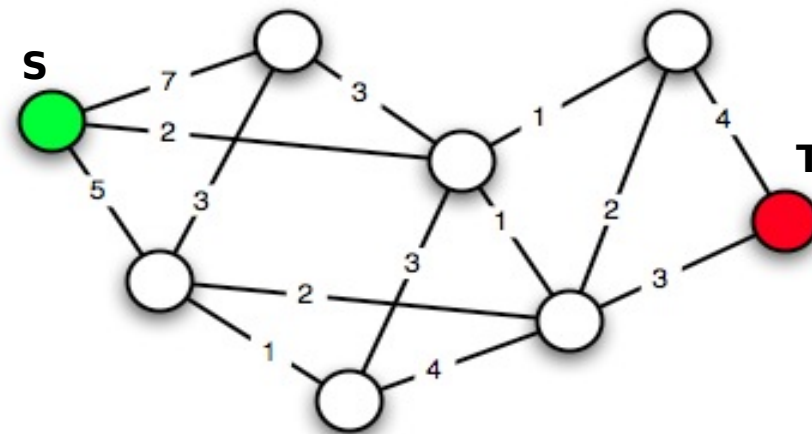
A max. folyam kiszámítása

- Minden természetes csővezeték rendszerben a max. folyam a természet által kiszámítódik
- Számítógép algoritmusok a max. folyam kiszámítására:
 - Lineáris programozás
 - A folyam egyenletek a lineáris programozási probléma egyenletei
 - Simplex (vagy ellipsoid módszer, vagy belső pont módszer)
 - Ford-Fulkerson
 - Amíg van egy nyitott út (olyan, ami megjavítja a folyamat) növeljük a folyamat ezen az úton
 - Edmonds-Karp
 - Ford-Fulkerson speciális esete
 - Használjunk szélességi keresést ennek az útnak a megtalálásához



Min. vágás hálózatokban

- Motiváció:
 - Találjuk meg a szűk keresztmetszetet a hálózatban
- Definíció:
 - **Min. vágás probléma**
 - adott
 - gráf $G=(V,E)$
 - kapacitás függvény $w: E \rightarrow \mathbf{R}_0^+$,
 - forrás halmaz S és cél halmaz T
 - Találjunk egy minimum vágást S és T között
- Egy C vágás élek olyan halmaza, hogy
 - $G \setminus C$ -ben nincs út S egyetlen csomópontjából T egyetlen csomópontjába se
- C vágás mérete: $\sum_{e \in C} w(e)$

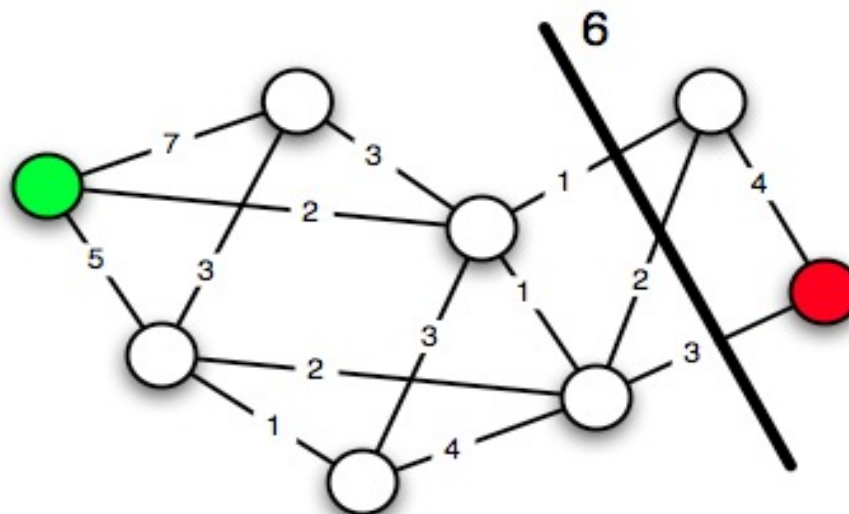
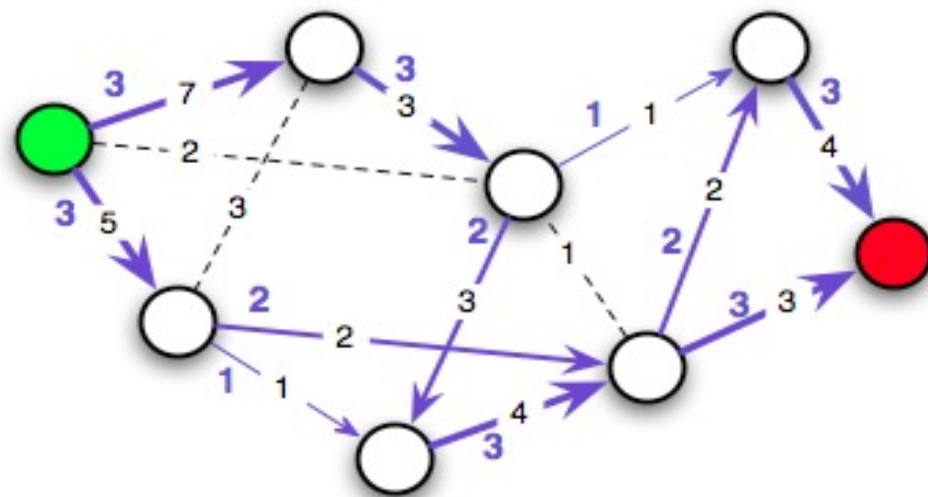


Min-vágás-max-folyam tétel

- Tétel

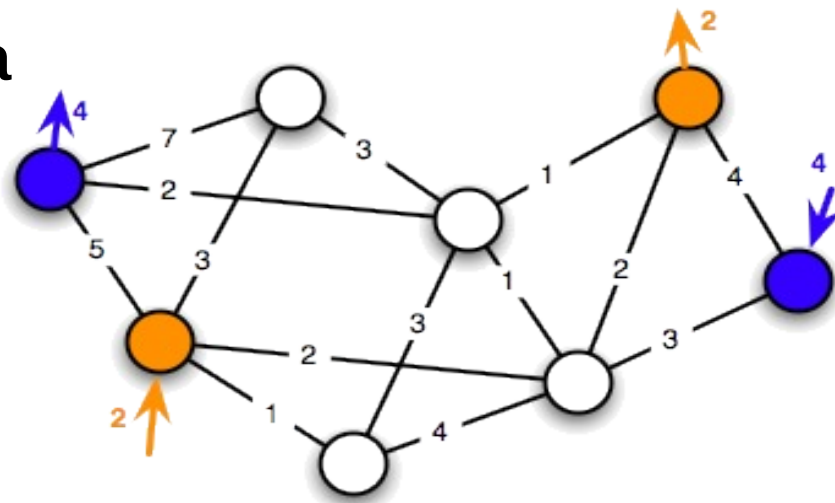
Minden gráfra, minden kapacitás függvényre, minden forrás és cél csomópont halmazra

a minimum vágás értéke egyenlő a maximum folyamatéval



Multi-commodity folyam probléma

- Motiváció:
 - Elméleti modell minden pont-pont kommunikáció optimális elhelyezésére a hálózatban kapacitásokkal
- Definíció
 - **Multi-commodity folyam probléma**
 - adott
 - gráf $G=(V,E)$
 - kapacitás függvény $w: E \rightarrow \mathbf{R}^+_0$,
 - commodities $K_1, \dots, K_k$:
 - $K_i=(s_i, t_i, d_i)$, ahol
 - s_i az forrás csomópont
 - t_i a cél csomópont
 - d_i a szükséglet



- Találjunk folyamokat $f_1, f_2, \dots, f_k$ minden commodity-nak, úgy hogy
 - kapacitás:

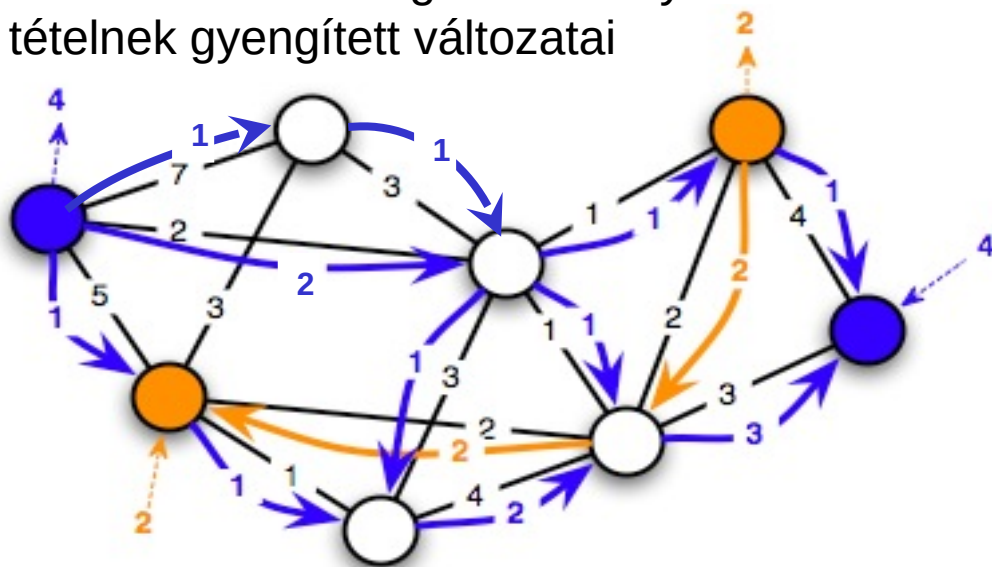
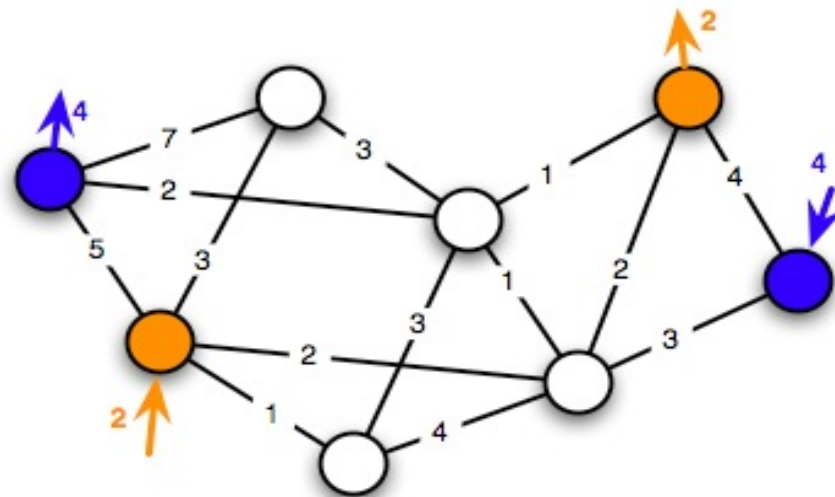
$$\sum_{i=1}^k f_i(u, v) \leq w(u, v)$$
 - folyam tulajdonság:

$$\forall v \notin \{s_i, t_i\} : \sum_{u \in V} f_i(u, v) = \sum_{u \in V} f_i(v, u)$$
 - szükséglet:

$$\sum_{v \in V} f_i(s_i, v) = \sum_{u \in V} f_i(u, t_i) = d_i$$

Multi-commodity folyam problémák megoldása

- A Multi-Commodity folyam probléma megoldható lineáris programozással
 - Az egyenlőségek a lineáris program egyenletei
 - Simplex vagy ellipsoid vagy belső pont algoritmus
- Léteznek a min-vágás-max-folyam tételnek gyengített változatai



Minimum sűrűség összefüggőséghez

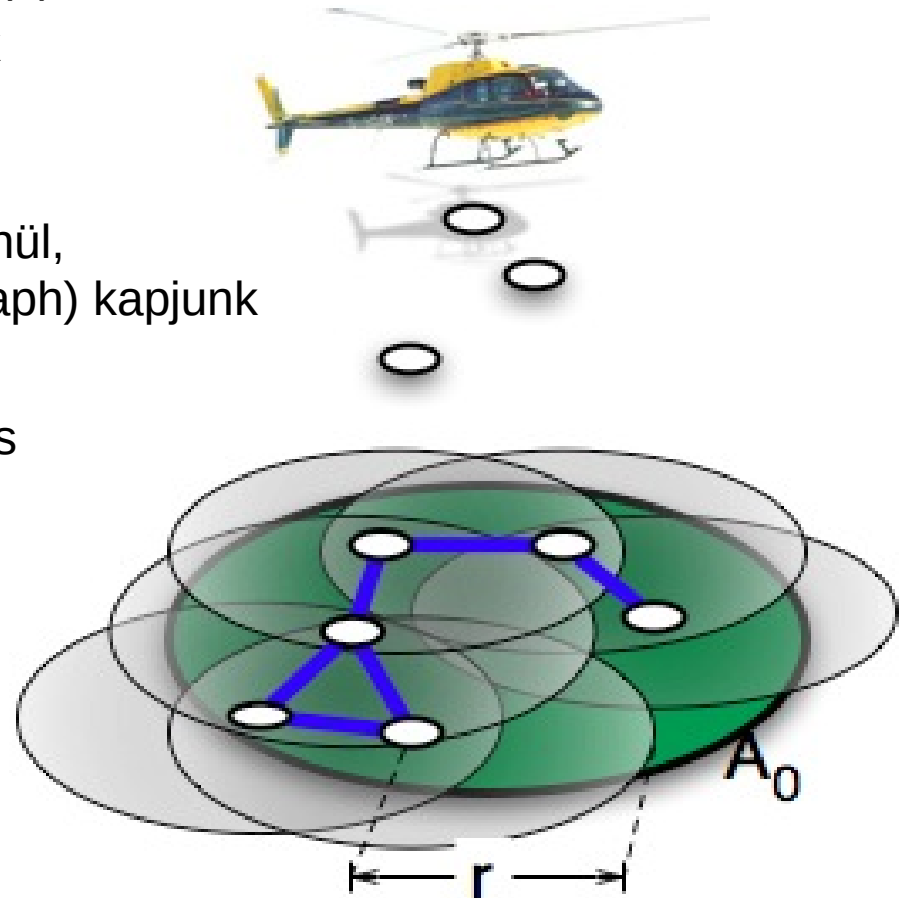
- Gupta, Kumar: **The Capacity of Wireless Networks**, *IEEE Transactions on Information Theory*, 46(2), 388-404, 2000.

- Kritikus energia vezetékek nélküli hálózatok összefüggéséhez
- Motiváció:
 - Hány csomópontot kell elhelyezni véletlenül, hogy egy összefüggő UDG-t (unit-disk graph) kapjunk

- Tétel
 - Egy négyzet alakú A_0 területen szükséges és elégséges véletlenül elhelyezni n csomópontot, hogy összefüggő UDG-t kapjunk, ahol

$$c \cdot \pi r^2 \cdot n = |A_0| \log n$$

- valamilyen konstans c -re.
- Ekvivalens: $\Theta\left(\frac{n}{\log n}\right) = \frac{|A_0|}{r^2}$



Miért kell ennyi csomópont?

- Elégséges feltétel, hogy a hálózat ne legyen összefüggő, hogy
 - legalább egy csomópont egy r oldalhosszú négyzetben
 - 8 szomszédos négyzet üres
- A valószínűség, hogy n csomópont közül egy se esik a szomszédos négyzetekbe:

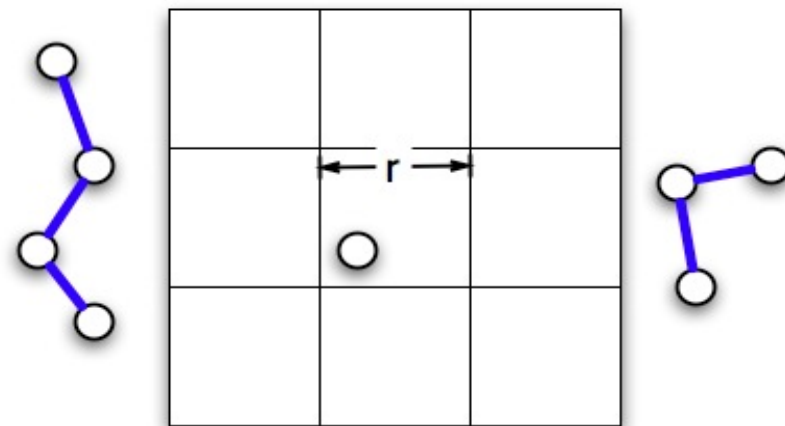
$$\left(1 - \frac{8r^2}{|A_0|}\right)^n$$

- $x \in [0, 0.75]$ esetén:

$$e^{-2x} \leq (1 - x) \leq e^{-x}$$

- Ezért (elég nagy A_0 -ra)

$$\left(1 - \frac{8r^2}{|A_0|}\right)^n \geq e^{-\frac{16r^2n}{|A_0|}}$$



- Ilyen izolált csomópontok számának várható értéke legalább

$$n \cdot e^{-\frac{16r^2n}{|A_0|}}$$

- Ha $r^2 = o\left(\frac{|A_0| \ln n}{n}\right)$, akkor

az izolált csomópontok számának várható értéke legalább 1

Elég ennyi csomópont?

- Elégséges feltétel az összefüggéshez
 - szomszédos $r/3$ oldalhosszúságú négyzetekben legalább egy csomópont legyen
- Annak a valószínűsége, hogy legalább egy csomópont egy ilyen négyzetben van:

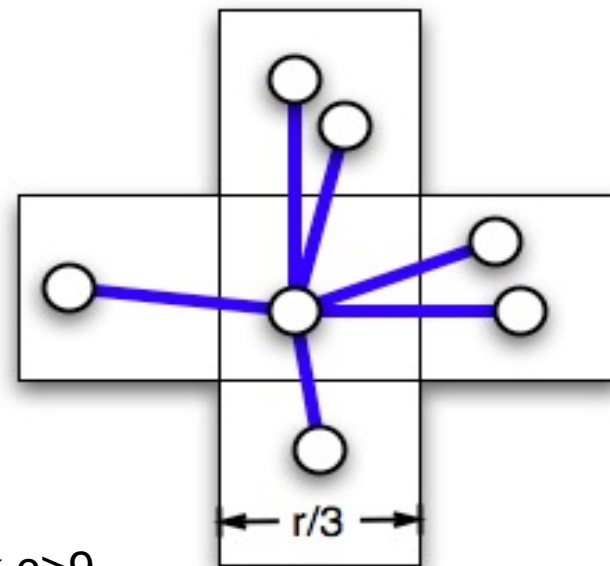
$$1 - \left(1 - \frac{r^2}{9|A_0|}\right)^n$$

- Válasszuk

$$r^2 = c \cdot \frac{|A_0| \ln n}{n}$$

- Akkor a fenti valószínűség:

$$1 - \left(1 - \frac{c \ln n}{9n}\right)^n \geq 1 - e^{-\frac{c}{9} \ln n} = 1 - n^{-\frac{c}{9}}$$



- Válasszuk $c > 9$
 - Akkor egy üresen maradt szomszédos négyzet valószínűsége $o(n^{-1})$
 - Ezt a valsz.-t $4n$ -nel megszorozva (minden szomszédos négyzethez), egy felső korlátot ad arra a valsz.-re, hogy van olyan csomópont, amelyiknek valamelyik oldalon nincs szomszédja
- Akkor a hiba valószínűsége $o(1)$

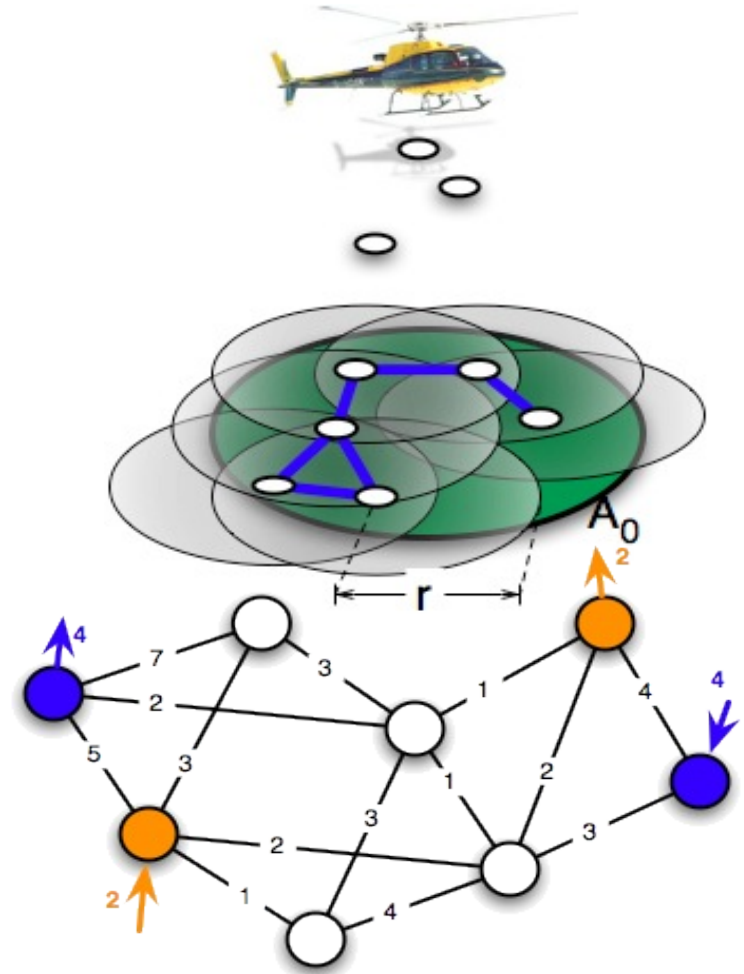
Hálózati folyam véletlen UDG-ban

- Motiváció:
 - Mekkora átvitelt enged a hálózat
- Tétel
 - Tegyük fel, hogy egy négyzet alakú A_0 területen n csomópont véletlenül van elhelyezve egyenletes eloszlás szerint, úgy hogy

$$\Theta\left(\frac{n}{\log n}\right) = \frac{|A_0|}{r^2}$$

- Tegyük fel, hogy minden csomópont képes W bit/s rátával adatot átvinni egy szomszédnak az UDG-ban. Tegyük fel, hogy minden csomópont véletlenül választ egy csomópontot és annak küld adatot. A ráta, ami elérhető minden csomópontban ehhez az átvitelhez:

$$\Theta\left(\frac{W}{\sqrt{n \log n}}\right)$$



Bizonyítás vázlat

- 1. megfigyelés:

- ha $\Theta\left(\frac{n}{\log n}\right) = \frac{|A_0|}{r^2}$

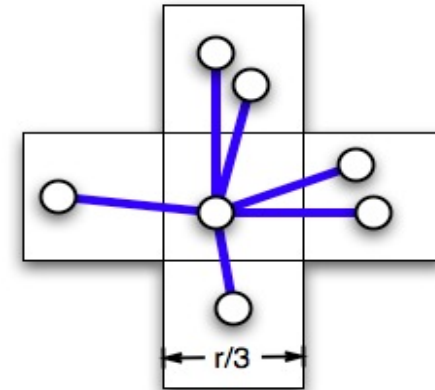
- a véletlen elhelyezés egy rács-szerű struktúrához vezet, melyben a cellák élhossza $r/3$

- 2. megfigyelés:

- A hálózat lényegében egy rács $m \times m$ cellával, ahol

$$m = \Theta\left(\sqrt{\frac{n}{\log n}}\right)$$

- Átlagosan minden cella $\log n$ csomópontot tartalmaz és ennyi egy csomóponthoz szomszédos élek száma egy szomszédos cellába



- Egy rácsban egy ekkora igény routolható n^2/m kapacitással (horizontális vagy vertikális vágás a szűk keresztmetszet)
- Ebben a hálózatban a minimális vágás $m \log n = (n \log n)^{1/2}$
- Ezért a multi-commodity folyam: $W/(n \log n)^{1/2}$

Diszkuszió

- Véletlenül elhelyezett csomópontok esetén egy A négyzeten $\Omega(n \log n)$ csomópontra van szükség
 - hogy az UDG összfüggő legyen
 - ahol $n = |A|/r^2$
- Ekkor a hálózat egy rácshoz hasonlóan viselkedik
 - egy polylogaritmikus faktortól eltekintve
- A rács szűk keresztmetszete a szélessége
 - optimális esetben (egy négyzet esetén) ez $n^{1/2}$.
- Ha a $\Omega(\log n)$ -szorzós overhead-et nem érjük el,
 - akkor a véletlenül elhelyezett UDG nem összefüggő
- Ez egy esete a „coupon-collector” problémának
 - Mennyi kártyát kell begyűjteni, amíg mind az n különböző kupont megtaláljuk