Modern irányzatok a bonyolultságelméletben: éles korlátok és dichotómia tételek

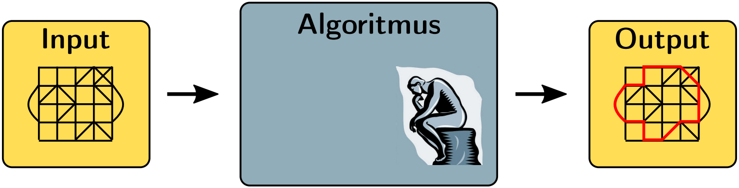
Marx Dániel

[1] Paraméteres Algoritmusok és Bonyolultság Kutatócsoport
Informatikai Kutatólaboratórium
SZTAKI

2015. június 15.

Kombinatorikus keresési problémák

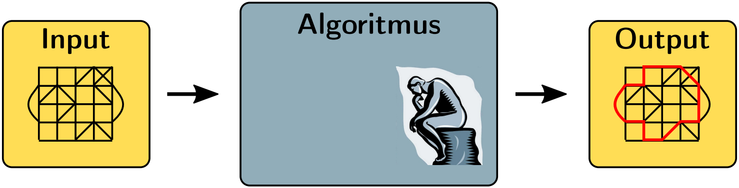
A feladat az adott bemenethez tartozó exponenciálisan nagy megoldástérből a legjobb megoldás kiválasztása.



- Legrövidebb út
- Utazóügynök-probléma
- Logikai formulák kielégíthetősége
- Korlátkielégítési problémák
- Ütemezési problémák
- Mintaillesztési problémák
- Hálózattervezés
- ...

Kombinatorikus keresési problémák

A feladat az adott bemenethez tartozó exponenciálisan nagy megoldástérből a legjobb megoldás kiválasztása.



Ez a terület nem foglalkozik olyan problémákkal ahol pl.

- folyamatosan érkező bemenetekre folyamatosan válaszolni kell,
- a bemenethez való hozzáférés költséges,
- a bemenet nem teljesen ismert, véletlen változókat tartalmaz.

Hatékony algoritmusok

Legrosszabbeset-analízis:
a maximális futási idő meghatározása n hosszú bemeneten.

- Szeretjük: lineáris idő, $O(n)$.

- Nem szeretjük: exponenciális idő, $2^{O(n)}$.

Elméleti szempontból rendkívül fontosak a **polinom időben** $n^{O(1)}$ megoldható problémák.

(Legrosszabbeset-analízis helyett lehetne vizsgálni

- átlagos futási időt véletlenszerű bemeneteken,
- futási időt gyakorlatban előforduló bemeneteken.)

$P \stackrel{?}{=} NP$

Az **NP-teljesség** elmélete erős bizonyítékot nyújt arra, hogy bizonyos problémákra nem létezik polinom idejű algoritmus.

[NP = Nemdeterminisztikus Polinomidő]

Több ezer problémáról mutatták meg, hogy NP-teljes. Pl.

- leghosszabb út keresése,
- utazóügynök-probléma
- gráfszínezés
- logikai formulák kielégíthetősége

$P \stackrel{?}{=} NP$

Az **NP-teljesség** elmélete erős bizonyítékot nyújt arra, hogy bizonyos problémákra nem létezik polinom idejű algoritmus.

[NP = Nemdeterminisztikus Polinomidő]

Több ezer problémáról mutatták meg, hogy NP-teljes. Pl.

- leghosszabb út keresése,
- utazóügynök-probléma
- gráfszínezés
- logikai formulák kielégíthetősége

Ha **bármelyik** NP-teljes problémára létezik polinom idejű algoritmus, akkor az **összesre** is létezik!

- $P = NP$: összes ilyen problémára létezik polinom idejű algoritmus és sok másra is!
- $P \neq NP$: NP-teljes problémákra nem létezik polinom idejű algoritmus.

Az NP-teljesség csak azt mondja, hogy (valószínűleg) nem létezik polinom idejű algoritmus a problémára!

Nem derül ki, hogy

- Létezik-e a $2^{O(n)}$ idejű nyers erő módszer helyett pl. $2^{O(\sqrt{n})}$ idejű algoritmus?
- Esetleg $n^{O(\log n)}$ vagy $n^{O(\log \log n)}$ idejű?
- Létezik-e olyan polinom idejű algoritmus, amely az optimálisnál 0,1%-al rosszabb eredményt ad?
- Létezik-e olyan algoritmus a KLIKK problémára, amelynek a futási ideje polinomiális a gráf méretében, de exponenciális a keresett klikk méretében?

Ezen kérdések megválaszolására a P $\neq$ NP sejtésnél erősebb hipotézisek szükségesek!

Exponential-Time Hypothesis (ETH)

3SAT **probléma:** adott egy n változós 3CNF Boole formula pl.

$$(x_1 \vee \bar{x}_3 \vee x_4) \wedge (\bar{x}_1 \vee \bar{x}_4 \vee x_5) \wedge (x_1 \vee x_3 \vee \bar{x}_5),$$

adjunk változóknak értékeket úgy, hogy a formula igaz legyen.

- Triviális algoritmus: 2^n próbálkozás
- Legjobb felső korlát: $O(1{,}30704^n)$ [Hertli 2011].

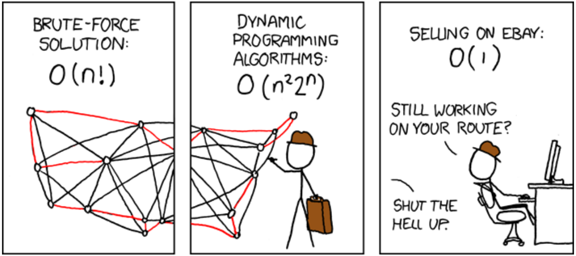
[Impagliazzo, Paturi, Zane 2001] fogalmazta meg a következő sejtést és vizsgálta következményeit:

Exponential-Time Hypothesis (ETH): az n változós 3SAT problémára nem létezik $2^{o(n)}$ idejű algoritmus.

Utazóügynök-probléma

Utazóügynök-probléma (TSP): adottak a távolságok n város között, keressük az összes várost meglátogató legrövidebb körutat.

- Nyers erő: $O(n!)$ lépés.
- Dinamikus programozás: $O(n^2 \cdot 2^n)$ lépés [Bellman 1962].
- Ha ETH igaz, nincs $2^{o(n)}$ idejű algoritmus.



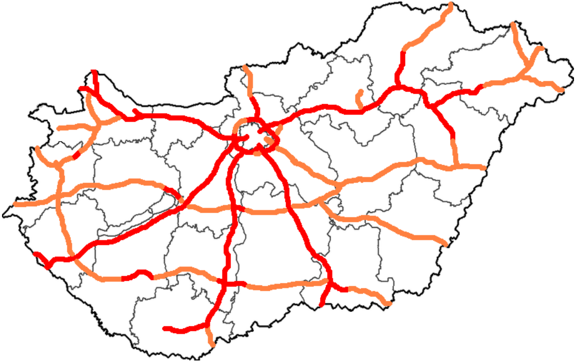
http://xkcd.com/399/

Utazóügynök-probléma síkgráfokon

Úthálózatok többé-kevésbé tekinthetők síkbarajzolható gráfoknak.

- Síkgráfokon a probléma megoldható $2^{O(\sqrt{n})}$ időben.
- Ha ETH igaz, nincs $2^{o(\sqrt{n})}$ idejű algoritmus.

Az algoritmus jelentős mértékben kihasználja a síkgráfok szerkezetét.



Utazóügynök-probléma síkgráfokon

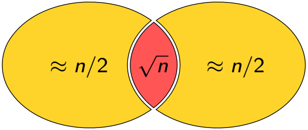
Úthálózatok többé-kevésbé tekinthetők síkbarajzolható gráfoknak.

- Síkgráfokon a probléma megoldható $2^{O(\sqrt{n})}$ időben.
- Ha ETH igaz, nincs $2^{o(\sqrt{n})}$ idejű algoritmus.

Az algoritmus jelentős mértékben kihasználja a síkgráfok szerkezetét.

Klasszikus tény:

Minden n csúcsú síkgráfnak van $O(\sqrt{n})$ méretű kiegyensúlyozott szeparátora.



Utazóügynök-probléma síkgráfokon

Úthálózatok többé-kevésbé tekinthetők síkbarajzolható gráfoknak.

- Síkgráfokon a probléma megoldható $2^{O(\sqrt{n})}$ időben.
- Ha ETH igaz, nincs $2^{o(\sqrt{n})}$ idejű algoritmus.

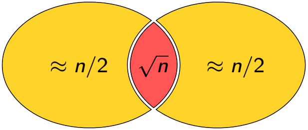
Az algoritmus jelentős mértékben kihasználja a síkgráfok szerkezetét.

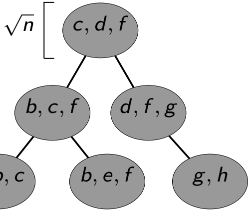
Klasszikus tény:

Minden n csúcsú síkgráfnak van $O(\sqrt{n})$ méretű kiegyensúlyozott szeparátora.

Modern nyelven:

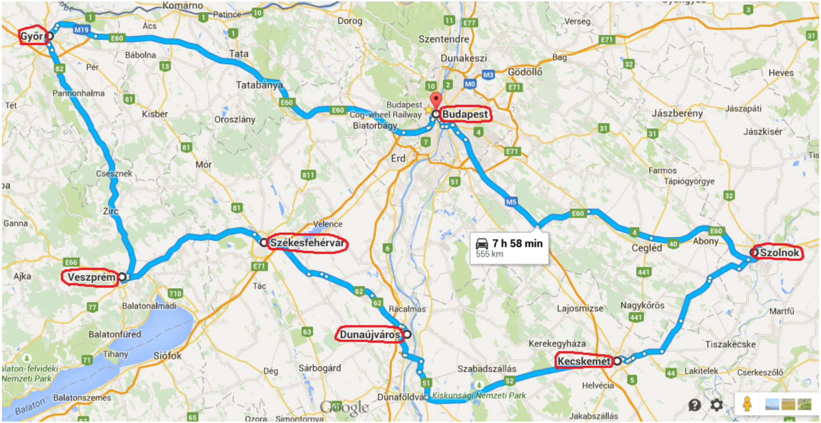
Minden n csúcsú síkgráfnak $O(\sqrt{n})$ a favastagsága (treewidth).





Utazóügynök-probléma síkgráfokon

Ha az úthálózat síkgráf lenne, valójában akkor sem akarnánk az összes kereszteződést meglátogatni.



Pontosabb kérdés: egy n csúcsú síkgráfban k kijelölt csúcsot kell meglátogatni a lehető legrövidebb körúton ($k \ll n$)

Paraméteres bonyolultság

Alapgondolat: A futási időt az input mérete n és az input valamilyen fontos k paraméterének a függvényében elemezzük. A cél a kombinatorikus robbanást k-ra korlátozni.

Példák a paraméterre:

- a kívánt megoldás mérete
- bizonyos objektumok száma
- a tér dimenziója
- ...

Definíció

Egy probléma **fixed-parameter tractable (FPT),** ha megoldható $f(k) \cdot n^{O(1)}$ időben, ahol f egy tetszőleges (csak k-tól függő) függvény.

Paraméteres bonyolultság

Az alábbi összes probléma megoldható nyers erővel $n^{O(k)}$ időben, de valójában FPT és megoldható $2^{O(k)} \cdot n^{O(1)}$ időben is:

- k hosszú út keresése
- k hosszú kör keresése
- összes él lefogása k csúccsal
- k diszjunkt 3 elemű halmaz kiválasztása
- legrövidebb Steiner fa k pont összekötésére
- ...

... és ETH mellett ezekre a problémákra nem létezik $2^{o(k)} \cdot n^{O(1)}$ idejű algoritmus!

Paraméteres bonyolultság

Az alábbi összes probléma megoldható nyers erővel $n^{O(k)}$ időben, de valójában FPT és megoldható $2^{O(k)} \cdot n^{O(1)}$ időben is:

- k hosszú út keresése
- k hosszú kör keresése
- összes él lefogása k csúccsal
- k diszjunkt 3 elemű halmaz kiválasztása
- legrövidebb Steiner fa k pont összekötésére
- ...

... és ETH mellett ezekre a problémákra nem létezik $2^{o(k)} \cdot n^{O(1)}$ idejű algoritmus!

- n^k idő: reménytelen nagy n és pl. $k = 60$ esetén.
- $1{,}3^k \cdot n$: akár lehet hatékony nagy n és $k = 60$ esetén is ($1{,}3^{60} \approx 7.000.000$).

Paraméteres bonyolultság

Az alábbi összes probléma megoldható nyers erővel $n^{O(k)}$ időben és ha ETH igaz, akkor ezekre a problémákra nem létezik $n^{o(k)}$ idejű algoritmus:

- k méretű klikk keresése
- k diszjunkt halmaz keresése egy halmazrendszerben
- k darab egymástól $\leq d$ távolságra lévő csúcs kiválasztása
- k központ kiválasztása, amitől a gráf összes csúcsa $\leq d$ távolságra van
- ...

Technikák


Gráfminorok
Kernelizálás
Színkódolás
Algebrai módszerek
Favastagság
Iterált tömörítés

Színkódolás (Color Coding)

k-ÚT

Bemenet: G gráf, k egész.
Kimenet: egy k hosszú út a gráfban.

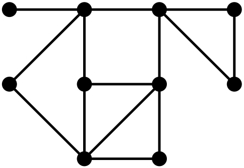
A probléma NP-nehéz, mivel tartalmazza HAMILTON ÚT problémát speciális esetként.

Tétel [Alon, Yuster, Zwick 1994]

A k-ÚT probléma megoldható $2^{O(k)} \cdot n^{O(1)}$ idõben.

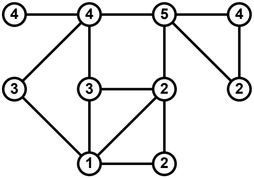
Színkódolás (Color Coding)

- Tekintsük a gráf csúcsainak egy véletlenszerű színezését k színnel.



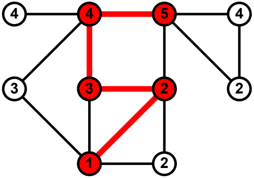
Színkódolás (Color Coding)

- Tekintsük a gráf csúcsainak egy véletlenszerű színezését k színnel.



Színkódolás (Color Coding)

- Tekintsük a gráf csúcsainak egy véletlenszerű színezését k színnel.



- Keressünk $1 - 2 - \cdots - k$ színezésű utat; a válasz „IGEN” vagy „NEM”.
  - Ha nincs k hosszú út: nincs $1 - 2 - \cdots - k$ színezésű út $\Rightarrow$ „NEM”.
  - Van legalább egy k hosszú út: k^{-k} valószínűséggel $1 - 2 - \cdots - k$ lesz a színezése $\Rightarrow$ „IGEN”.

Hibavalószínűség

Hasznos tény

Ha a siker valószínűsége legalább p, akkor annak a valószínűsége, hogy $1/p$ ismétlés után sem lesz „IGEN” a válasz legfeljebb

$$(1-p)^{1/p} < \left(e^{-p}\right)^{1/p} = 1/e \approx 0{,}38$$

Hibavalószínűség

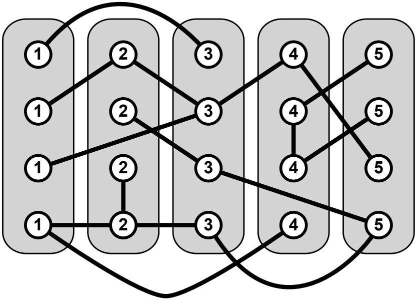
Hasznos tény

Ha a siker valószínűsége legalább p, akkor annak a valószínűsége, hogy $1/p$ ismétlés után sem lesz „IGEN" a válasz legfeljebb

$$(1-p)^{1/p} < \left(e^{-p}\right)^{1/p} = 1/e \approx 0{,}38$$

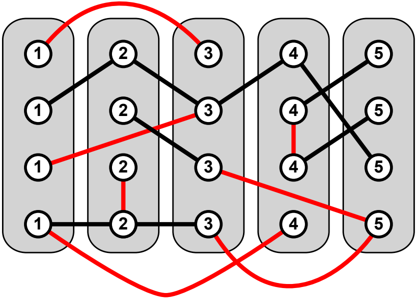
- Vagyis ha $p > k^{-k}$, akkor a hiba valószínűsége k^k ismétlés után legfeljebb $1/e$.
- Az egész algoritmust konstans sokszor megismételve a hiba valószínűsége levihető tetszőlegesen kicsi konstansra.
- Pl. $100 \cdot k^k$ véletlen színezés után a hibás válasz valószínűsége legfeljebb $1/e^{100}$.

$1 - 2 - \cdots - k$ színezésű út keresése



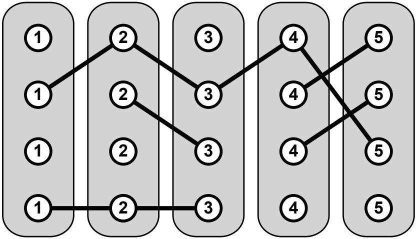
- Nemszomszédos színeket összekötő élek feleslegesek.
- A maradék éleket irányítsuk a nagyobb szín felé.
- A feladat annak ellenőrzése, hogy van-e irányított út az 1-es színosztálytól a k-as színosztályig.

$1 - 2 - \cdots - k$ színezésű út keresése



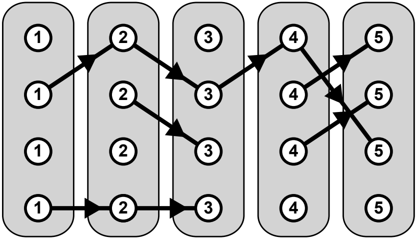
- Nemszomszédos színeket összekötő élek feleslegesek.
- A maradék éleket irányítsuk a nagyobb szín felé.
- A feladat annak ellenőrzése, hogy van-e irányított út az 1-es színosztálytól a k-as színosztályig.

$1 - 2 - \cdots - k$ színezésű út keresése



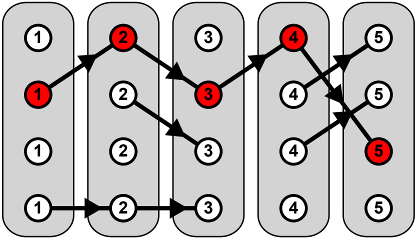
- Nemszomszédos színeket összekötő élek feleslegesek.
- A maradék éleket irányítsuk a nagyobb szín felé.
- A feladat annak ellenőrzése, hogy van-e irányított út az 1-es színosztálytól a k-as színosztályig.

$1 - 2 - \cdots - k$ színezésű út keresése



- Nemszomszédos színeket összekötő élek feleslegesek.
- A maradék éleket irányítsuk a nagyobb szín felé.
- A feladat annak ellenőrzése, hogy van-e irányított út az 1-es színosztálytól a k-as színosztályig.

$1 - 2 - \cdots - k$ színezésű út keresése

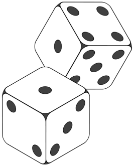


- Nemszomszédos színeket összekötő élek feleslegesek.
- A maradék éleket irányítsuk a nagyobb szín felé.
- A feladat annak ellenőrzése, hogy van-e irányított út az 1-es színosztálytól a k-as színosztályig.

Color Coding

k-**Út**

Színkódolás
siker valószínűsége: k^{-k}



$1 - 2 - \cdots - k$
színezésű út keresése

polinom időben megoldható

Négyzetgyök-jelenség síkgráfokon

Síkgráfokon gyakran sokkal hatékonyabb algoritmusokat tudunk találni és legtöbbször $\sqrt{k}$ szerepel a futási időben.

$2^{O(\sqrt{k})} \cdot n^{O(1)}$ idő

- k hosszú út
- k független csúcs
- k független háromszög
- összes él lefogása k csúccsal
- ...

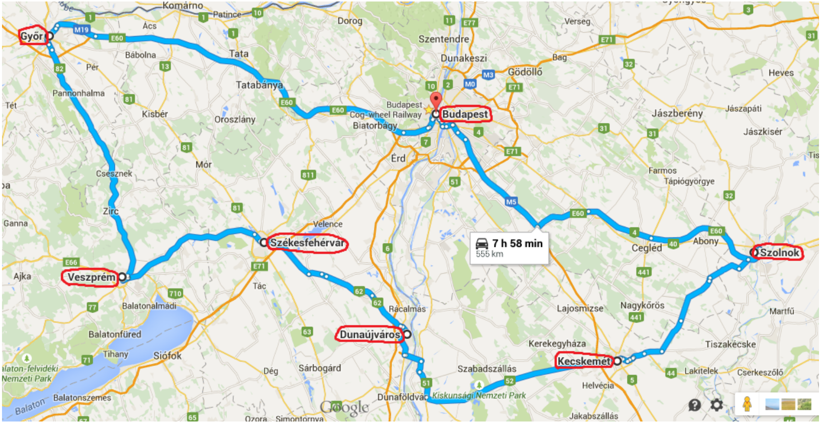
$n^{O(\sqrt{k})}$ idő

- k csúcs egymástól $\geq d$ távolságra
- k csúcs akiktől minden csúcs $\leq d$ távolságra van
- k terminál elválasztása minimális számú él törlésével

... és ETH mellett ezekre a problémákra nem létezik $2^{o(\sqrt{k})} \cdot n^{O(1)}$ ill. $n^{o(\sqrt{k})}$ algoritmus!

Utazóügynök-probléma síkgráfokon

- Nyers erő: $k! \cdot n^{O(1)}$ lépés.
- Dinamikus programozás: $2^k \cdot n^{O(1)}$ lépés [Bellman 1962].
- Síkgráfokra: $2^{O(\sqrt{k})} \cdot n^{O(1)}$ [Klein M. 2014]
- Ha ETH igaz, nincs $2^{o(\sqrt{k})} \cdot n^{O(1)}$ idejű algoritmus síkgráfokon.



Dichotómia tételek

- Minden problémának vannak könnyű és nehéz speciális esetei.
- Cél: a speciális esetek szisztematikus vizsgálata, az összes könnyű és az összes nehéz speciális eset felderítése.
- Dichotómia tétel: egy végtelen problémacsalád összes tagjáról megállapítjuk, hogy könnyű (pl. polinom időben megoldható) vagy nehéz (pl. NP-teljes).
- Két fő terület, ahol ilyen vizsgálatok történtek:
  - Logikai (Satisfiability) és korlátkielégítési (CSP) problémák: milyen típusú feltételek teszik nehézzé a problémát?
  - Gráfproblémák: milyen gráfok/gráfosztályok teszik nehézzé a problémát?

Logikai formulák

Feladat: 0-1 változókon adottak logikai feltételek, úgy kell értékeket adni a változókat, hogy az összes feltétel teljesüljön.

Ez lehet könnyű vagy nehéz, attól függően, hogy milyen formájúak a feltételek.

- $x_1 \neq x_2$: polinom időben megoldható (két színnel színezés)
- $x_1 \vee x_2$, $\bar{x}_1 \vee x_2$, $\bar{x}_1 \vee \bar{x}_2$: polinom időben megoldható.
- $(x_1 \wedge x_2 \wedge x_3) \to x_4$: polinom időben megoldható (Horn Sat).
- $x_1 \vee x_2 \vee x_3$, $\bar{x}_1 \vee x_2 \vee x_3$: NP-teljes (3Sat)
- $x_1 \oplus x_2 \oplus x_3 = 1$: polinom időben megoldható (lineáris egyenletrendszer).

Logikai formulák

Feladat: 0-1 változókon adottak logikai feltételek, úgy kell értékeket adni a változókat, hogy az összes feltétel teljesüljön.

Ez lehet könnyű vagy nehéz, attól függően, hogy milyen formájúak a feltételek.

- $x_1 \neq x_2$: polinom időben megoldható (két színnel színezés)
- $x_1 \vee x_2$, $\bar{x}_1 \vee x_2$, $\bar{x}_1 \vee \bar{x}_2$: polinom időben megoldható.
- $(x_1 \wedge x_2 \wedge x_3) \rightarrow x_4$: polinom időben megoldható (Horn Sat).
- $x_1 \vee x_2 \vee x_3$, $\bar{x}_1 \vee x_2 \vee x_3$: NP-teljes (3Sat)
- $x_1 \oplus x_2 \oplus x_3 = 1$: polinom időben megoldható (lineáris egyenletrendszer).

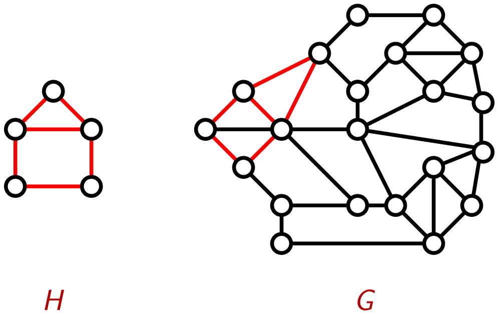
Schaefer Dichotómia Tétele [Schaefer 1978]

Pontosan karakterizálja azokat a logikai feltételeket, amelyek mellett a probléma polinom időben megoldható és azokat, amelyek mellett a probléma NP-teljes.

Részgráf keresés

Részgráf keresés

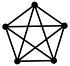
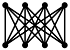
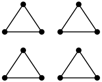
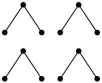
Bemenet: H, G gráfok.
Kimenet: G-nek egy H-val izomorf részgráfja.



Ha $\mathcal{H}$ valamilyen gráfosztály, $\mathcal{H}$-Részgráf Keresés probléma az a speciális eset, ahol a H mintagráf a $\mathcal{H}$ osztályba tartozik.

Részgráf Keresés speciális esetei

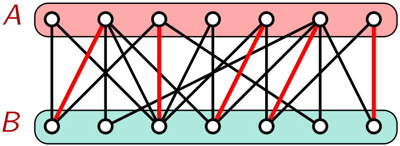
Számos speciális esetről tudjuk, hogy NP-teljes:

	Klikk NP-teljes
	Teljes Páros Gráf NP-teljes
	*k*-Út NP-teljes
	Háromszög Pakolás NP-teljes.
	P_2-Pakolás NP-teljes.

Párosítás probléma

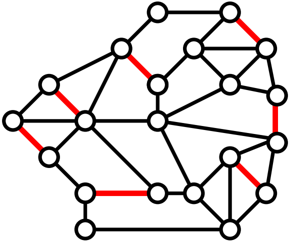
Párosítás keresése páros gráfban:

polinom időben megoldható („magyar módszer”) [Kőnig, Egerváry]



Párosítás keresése általános gráfban:

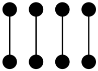
polinom időben megoldható [Edmonds 1965]



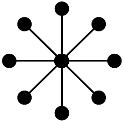
Polinom időben megoldható esetek

Néhány gráfosztály, amelyre a $\mathcal{H}$-Részgráf Keresés polinom időben megoldható:

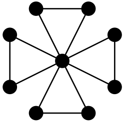
- $\mathcal{H}$ csak független éleket tartalmaz (párosítás)
- $\mathcal{H}$ csak csillagokat tartalmaz
- $\mathcal{H}$ csak „szélmalmokat” tartalmaz
- $\mathcal{H}$ csak „hosszú csillagokat” tartalmaz



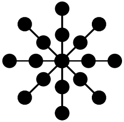
független élek



csillag



szélmalom

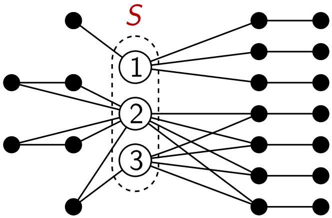


hosszú csillag

Részgráf keresés

Definíció

Az $\mathcal{H}$ gráf osztály **matching splittable** ha létezik olyan $c \geq 0$ konstans, hogy minden $H \in \mathcal{H}$ gráfból lehet törölni c csúcsot úgy, hogy utána minden összefüggő komponensnek ≤ 2 csúcsa legyen.



Tétel [Jansen, M. 2015]

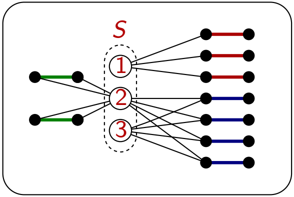
Legyen $\mathcal{H}$ egy csúcstörlésre nézve zárt gráfosztály.

- Ha $\mathcal{H}$ matching splittable, akkor a $\mathcal{H}$-Részgráf Keresés probléma randomizált polinom időben megoldható,
- és minden más esetben NP-teljes.

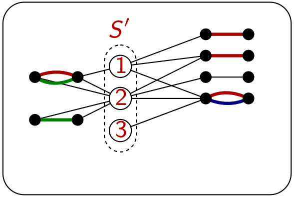
Algoritmus

Tétel [Jansen, M. 2015]

Legyen $\mathcal{H}$ egy csúcstörlésre nézve zárt gráfosztály. Ha $\mathcal{H}$ matching splittable, akkor a $\mathcal{H}$-Részgráf Keresés probléma randomizált polinom időben megoldható.



H

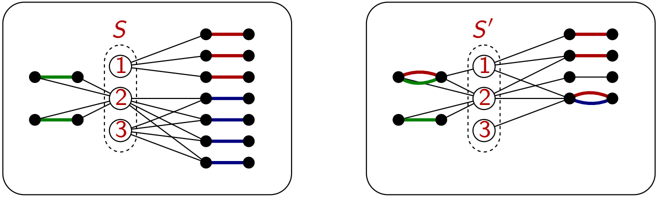


G

Algoritmus

Tétel [Jansen, M. 2015]

Legyen $\mathcal{H}$ egy csúcstörlésre nézve zárt gráfosztály. Ha $\mathcal{H}$ matching splittable, akkor a $\mathcal{H}$-Részgráf Keresés probléma randomizált polinom időben megoldható.



H G

Theorem [Mulmuley, Vazirani, Vazirani 1987]

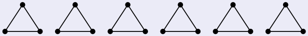
Legyen G egy gráf ahol az élek c színnel vannak színezve legyenek $k_1, \ldots, k_c$ egészek. Létezik egy $n^{O(c)}$ idejű randomizált algoritmus annak eldöntésére, hogy van-e olyan párosítás, amely pontosan k_i darab i színű élet tartalmaz.

NP-teljesség

Lemma

Legyen $\mathcal{H}$ egy csúcstörlésre nézve zárt gráfosztály amely nem matching splittable. Az alábbiak közül legalább az egyik igaz:

- $\mathcal{H}$ tartalmazza az összes klikket.
- $\mathcal{H}$ tartalmazza az összes teljes páros gráfot.
- $\mathcal{H}$ tartalmazza az n háromszögből álló gráfot minden $n \geq 1$-re.



- $\mathcal{H}$ tartalmazza az n kételű útból álló gráfot minden $n \geq 1$-re.

$\mathcal{H}$-Részgráf Keresés mind a négy esetben NP-teljes!

Bizonyítás a Ramsey-tételen alapszik (minden kellően nagy gráf tartalmaz vagy nagy klikket vagy nagy független halmazt).

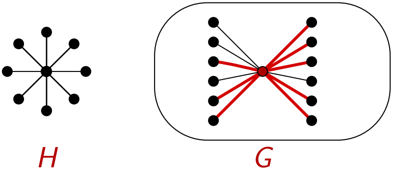
Részgráfok számlálása

Részgráf Számlálás

Bemenet: H, G gráfok.
Kimenet: G-nek a H-val izomorf részgráfjainak a száma.

Párosítások keresése könnyű, de az adott méretű párosítások megszámlálása NP-nehéz! [Valiant 1979]

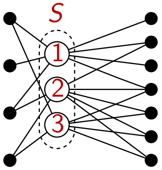
Ha a $\mathcal{H}$ osztály csak csillagokat tartalmaz, akkor a $\mathcal{H}$-Részgráf Számlálás polinom időben megoldható: végigpróbáljuk a középpont összes lehetséges elhelyezését (egy d fokú csúcson egy s fokú csillag pontosan $\binom{d}{s}$-szer jelenik meg).



Részgráfok számlálása

Definíció

Az $\mathcal{H}$ gráf osztálynak korlátos a csúcslefogási száma, ha létezik olyan $c \geq 0$ konstans, hogy minden $H \in \mathcal{H}$ gráfban az összes élet le lehet fogni c csúcs törlésével.



Tétel

Legyen $\mathcal{H}$ egy csúcstörlésre nézve zárt gráfosztály.

- Ha $\mathcal{H}$-nek korlátos a csúcslefogási száma, akkor a $\mathcal{H}$-Részgráf Számlálás probléma polinom időben megoldható [korábbi munkák],
- és minden más esetben NP-teljes [Curticapean, M. 2014].

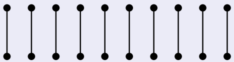
Részgráfok számlálása

Ramsey-tétel újabb alkalmazása:

Lemma

Ha $\mathcal{H}$ egy csúcstörlésre zárt gráfosztály és nem korlátos a csúcslefogási száma, akkor legalább az egyik teljesül:

- $\mathcal{H}$ tartalmazza az összes klikket.
- $\mathcal{H}$ tartalmazza az összes teljes páros gráfot.
- $\mathcal{H}$ tartalmazza a független éleket tartalmazó gráfokat.



$\mathcal{H}$-Részgráf Számlálás NP-nehéz mindhárom esetben!

Megjegyzés: az eredmény kiterjeszthető olyan osztályokra, amelyek nem zártak a csúcstörlésre nézve, de ez sokkal bonyolultabb eszközöket igényel.

Összefoglaló

- A polinom időben megoldható/NP-teljes felosztásnál finomabb, kvantitatívabb kérdéseket is meg tudunk válaszolni.
- A $\mathrm{P} \neq \mathrm{NP}$ sejtés helyett erősebb hipotézisre van szükségünk: Exponential-Time Hypothesis (ETH)
- Paraméteres bonyolultság azt vizsgálja, hogy a bemenetnek pontosan melyik paramétere idézi elő az exponenciális robbanást.
- Dichotómia tételek pontosan feltérképeznek egy problémacsaládot, megtalálják az összes könnyű speciális esetet és az összes algoritmikus ötletet a területen.

Összefoglaló

- A polinom időben megoldható/NP-teljes felosztásnál finomabb, kvantitatívabb kérdéseket is meg tudunk válaszolni.
- A $\mathsf{P} \neq \mathsf{NP}$ sejtés helyett erősebb hipotézisre van szükségünk: Exponential-Time Hypothesis (ETH)
- Paraméteres bonyolultság azt vizsgálja, hogy a bemenetnek pontosan melyik paramétere idézi elő az exponenciális robbanást.
- Dichotómia tételek pontosan feltérképeznek egy problémacsaládot, megtalálják az összes könnyű speciális esetet és az összes algoritmikus ötletet a területen.

Köszönöm a figyelmet!