Szegedi Tudományegyetem

Természettudományi és Informatikai Kar

SZTE Informatika Doktori Iskola

Doktori értekezés tézisei

A k-klikk probléma

Modellezési kifejezőereje, soros és párhuzamos algoritmusai

Zaválnij Bogdán

Szeged, 2020.

Témavezetők:
Dr. Krész Miklós
Dr. Szabó Sándor

Disszertációnk témaköre a diszkrét optimalizálás, és azon belül is kifejezetten a gráfokkal jellemzett problémák területe. Ezen problémák a legkülönféle területről származnak, és a matematikai programozás egy igen érdekes részterülete. Munkánk ezen terület egy speciális és kitüntetett feladatával foglalkozik, a k-klikk problémával. Ezen kívül a maximum klikk problémát is érintjük bizonyos esetekben. Mivel az említett két probléma NP-teljen illetve NP-nehéz, így még közepes méretű feladat esetén is igen komoly nehézséget jelent a megoldásuk. Ezért fontos, hogy hatékony algoritmusokat és nagy számítási teljesítményt alkalmazzunk, például szuperszámítógépet. Ez a cél vezet majd el minket olyan más feladatokhoz, mint a problémák részproblémákra osztása, hatékony ütemezés, illetve a részeredmények végeredménnyé integrálása.

A problémák megoldására különböző modelleket használhatunk, melyek között gyakorlatilag szabadon választhatunk. Így választás legtöbbször azon múlik, milyen megoldó szoftver áll rendelkezésünkre. A leggyakoribb választás így valamilyen lineáris programozási (LP) eszköz, azon belül vegyes egész (MILP), égészértékű (ILP) vagy nulla-egy (0–1 LP) programozási modell alkalmazása. Kombinatorikus optimalizálási vagy döntési feladatokhoz leginkább ILP vagy 0–1 LP használatos, de természetesen más modellek is számításba jönnek, mint SAT vagy MaxSAT modellek vagy korlátprogramozás (CP). Bár általában mindegyik alkalmas a probléma modellezésére mégis lehetnek hatékonyságbeli különbségek, méghozzá két okból kifolyólag. Egyfelől egyes problémákhoz bizonyos modellek jobban illeszkednek, és így a megoldószoftver is hatékonyabb lehet. Másfelől egyes megoldószoftverek fejlettebbek másoknál, mivel sokkal több energiát fektettek a fejlesztésükbe. Így tehát az elsődleges választás szinte mindig az ILP megközelítés, ami kellően sokoldalú, könnyen modellezhető és több nagyon fejlett szoftver adott hozzá. Vegyük észre azonban, hogy ez nem feltétlenül a leghatékonyabb megközelítés, és egyes nehéz problémák esetén kudarcot okoz. Egy másik problémára is fel kell hívjuk a figyelmet, méghozzá a számítások megbízhatósága terén. Az ILP megoldók LP számításokat végeznek, amely viszont kerekítési hibákat halmozhat fel [Aki2016]. Ha a megbízhatóság fontos, akkor másik megoldót kell választani, amely mentes ezen hibáktól, és a klikk-keresők ilyenek, mivel csakis egész és bit értékekkel számolnak. A jelen munka arra is rá kíván mutatni, hogy a klikk keresés jó és hatékony modell lehet sok különféle probléma esetén.

Ugyancsak kiemeljük majd, hogy a gráf reprezentáció alkalmasabb arra, hogy nagyléptékű párhuzamos algoritmusokat fejlesszünk ki, és szuperszámítógépek segítségével nehéz feladatokat oldjunk meg. A párhuzamosítás kérdésköre ugyanis különösen problematikus a kombinatorikus optimalizálás témakörében.

A probléma definíciója

Legyen $G = (V, E)$ egy egyszerű, véges gráf. V a gráf csúcsainak halmaza, E a $V \times V$ Descartes-szorzat részhalmaza, az élek halmaza. V és E véges, és a gráfban nincsenek dupla élek vagy hurkok. Az élek irányítatlanok, és sem a csúcsok, sem az élek nem súlyozottak.

Vegyük a G gráf $\Delta = (U, F)$ részgráfját. Azt mondjuk, hogy Δ a G egy klikkje, ha $F = U \times U$. Más szavakkal, Δ G egy klikkje, ha Δ minden csúcsa páronként szomszédos G-ben. Δ csúcsainak számosságát, azaz az U halmaz méretét a Δ klikk méretének is nevezzük. Egy k méretű Δ klikket sokszor a G gráf k-klikkjének nevezünk.

1. Probléma. *Adott a G véges egyszerű gráf, és a k pozitív egész szám. Eldöntendő, hogy G tartalmaz-e k-klikket.*

A k-klikk probléma egy nevezetes NP-teljes probléma, amely harmadik volt Karp eredeti 21 NP-teljes problémája között [Karp1972].

Tézisek

Első tézis

Bemutattunk problémákat, melyek hatékonyan fogalmazhatók meg mint klikk-keresési feladat. Többek között a latin négyzetek és szúdoku, a sok királynő feladat, Costas táblázatok és kódelmélet témaköréből. Bemutattuk, hogy a részgráf izomorfia is visszavezethető klikk feladatra, és rámutattunk, hogy a kémia és alakfelismerés témakörének feladatai így megoldhatóak. Bemutattuk, hogy ütemezési feladatokat is meg tudunk fogalmazni klikk feladatként, és hogy ezek hatékonyan is számolhatóak.

Egyes esetekben méréseket is mellékeltünk, melyek célja nem az volt, hogy hatékonyságban versenyezzenek, hanem az hogy megmutassa, hogy az átfogalmazások nem triviális méretű problémák esetén is értelmes, és összehasonlítható más megoldásokkal.

Itt egy problémát emelünk ki, hogy eredményeinket bemutassuk, a Costas táblázat problémáját. A probléma a radar és szonár eszközök csatornahatékonyságának mérnöki problémájából ered [Cost1965]. Az átfogalmazás – tudomásunk szerint – nem ismert a szakirodalomban. A probléma formálisan felírva a következő. Adott egy $n \times n$ táblázat, melynek celláiba n pontot kell rakjunk, a következő feltételek szerint. Két pont nem helyezkedhet el ugyanazon sorban vagy oszlopban, illetve páronként a pontok közötti irányvektor minden esetben különböző kell, hogy legyen a többi irányvektortól.

Mivel a feltételek pontpárok közötti irányvektorokra írnak elő feltételt (ez egy pontpár esetén összesen 4 koordináta), a segédgráf $G(V, E)$ csúcsait négyesekkel fogunk jellemezni: (x_1, y_1, x_2, y_2), ahol a két csúcs koordinátáit írjuk fel, a (x_1, y_1) és a (x_2, y_2) koordinátákat. Ezek után felsorolhatnánk az összes lehetséges négyest, de egyértelműen elég csak a valid párokat felsorolni, azaz amik nem esnek egy sorba vagy oszlopba. Azaz nem soroljuk fel a $(1, 1, 1, 3)$ négyest, mert a két pont egy oszlopban van. A segédgráfunknak így $n^2(n-1)^2/2$ csúcsa lesz. A segédgráf élei az összekötött pontpárok közötti megengedő viszonyt reprezentálja. Külön figyelnünk kell arra, hogy a segédgráf élei két pontpárt reprezentáló csúcsot kötnek össze, de ebből a 4 pontból kettő egybeeshet, így egy él 3 vagy 4 pontot reprezentál.

Formálisan *nem* húzzuk be az élt két, (x_1, y_1, x_2, y_2) és (a_1, b_1, a_2, b_2), csúcs között, akkor, és csak akkor, ha:

1. Ha két pontpárnak azonos sora vagy oszlopa van;
2. Ha egy pontpárnak azonos sora vagy oszlopa van, és 4 különböző pont van;
3. Ha 3 pont van, ezek között egy párnak azonos a sora vagy oszlopa, és az irányvektorok között van egyezőség;
4. Ha 4 pont van, nincs azonos sor és oszlop, de az irányvektorok között egyezés van.

Minden más esetben behúzzuk a csúcsok közötti élt. Egy $k = n(n-1)/2$ méretű k-klikk n darab pont összes párját reprezentálja, azaz megoldása a problémának.

Ezen egyszerű átfogalmazás segítségével – még kernelizálást sem használva –, a nem triviális 14×14 táblázat egy megoldását néhány másodperc alatt meg lehet találni, és az összes megoldást fél óra alatt fel lehet sorolni.

Második tézis

Kiválasztottunk a problémáknak egy osztályát, a gráfok és hipergráfok színezésének problémáját, és részletesen bemutattuk, hogy miképp lehet ezeket ugyancsak viszavezetni k-klikk-keresésre. Az átfogalmazás segítségével meg tudtunk támogatni egy nyitott kérdést, melyet Volosin tett fel a hipergráfok színezhetősége kapcsán.

Példaként a gráfok csúcsainak háromszögmentes színezésének [Szab2012] k-klikk átfogalmazását mutatjuk be. A kérdéses színezés definíciója a következő:

1. G minden csúcsához pontosan egy színt rendelünk.
2. A G gráf bármely 3-klikkje (háromszöge) csúcsai nem kaphatják mind ugyanazt a színt.

A fenti problémát visszavezethetjük a 1. Problémára. Adott a $G = (V, E)$ gráf és a k pozitív egész. Egy $\Gamma = (W, F)$ segédgráfot szerkesztünk. A Γ csúcsaihoz hármasokat rendelünk:

$$(\{u, v\}, a, b), \quad \text{ahol} \quad \{u, v\} \in E,\ 1 \leq a, b, \leq k.$$

Legyen m a G gráf csúcsainak számossága, azaz $m = |E|$. A hármasok száma így mk^2 lesz.

Egy $(\{u, v\}, a, b)$ hármas azt az információt kódolja, hogy az u és v csúcsokhoz, azaz az $\{u, v\}$ él végpontjaihoz az a és b színt rendeljük. (Feltételezzük, hogy G-ben nincs izolált csúcs.)

Vegyük a Γ következő két különböző csúcsát

$$w_1 = (\{u_1, v_1\}, a_1, b_1), \qquad w_2 = (\{u_2, v_2\}, a_2, b_2)$$

Legyen

$$X = \{u_1, v_1\} \cup \{u_2, v_2\} = \{u_1, v_1, u_2, v_2\}.$$

Egyértelmű, hogy $|X| \leq 4$, és mivel $u_1 \neq v_1$, így $|X| \geq 2$, azaz $2 \leq |X| \leq 4$. Legyen H_X a G részgráfja, melyet X feszít ki. A u_1, v_1, u_2, v_2 csúcsok rendre a a_1, b_1, a_2, b_2 színeket kapják a H_X gráfban.

Az $|X| \leq 3$ esetben ezen csúcsok között egyesek egybeesnek, így megtörténhet, hogy ugyanahhoz a csúcshoz két különböző színt akarnánk rendelni. Ebben az esetben az H_X gráf nem kvalifikáló.

Ugyancsak megtörténhet, hogy a H_X egy 3-klikket tartalmaz, és a klikk mindhárom csúcsához ugyanazt a színt rendelnénk. Ebben az esetben a H_X gráf ismét csak nem kvalifikáló.

Minden más esetben a H_X gráfot kvalifikálónak nevezünk.

Amikor megszerkesztjük a Γ segédgráfot, a gráf w_1, w_2 csúcsait összekötjük Γ-ban abban az esetben, ha H_X kvalifikáló.

1. Megfigyelés. *Ha G gráfnak van háromszögmentes k-színezése, akkor a Γ segédgráfban van m-klikk.*

Bizonyítás. Tegyük fel, hogy a G gráf csúcsainak van háromszög mentes színezése k színnel. Legyen $f : V \to \{1, \ldots, k\}$ az a függvény, ami ezt a színezést kódolja. Legyen

$$D = \{(\{u, v\}, f(u), f(v)) : \ \{u, v\} \in E\},$$

és legyen Δ a Γ egy részgráfja, melyet D feszít ki. Egyértelmű, hogy $|D| = m$. Azt állítjuk, hogy Δ a Γ gráf egy klikkje.

A bizonyításhoz válasszunk két különböző csúcsot D-bol, a w_1 és w_2 csúcsokat. Vizsgáljuk meg ezen két csúcs által meghatározott H_X gráfot. Az f függvény a H_X gráf minden egyes csúcsához pontosan egy színt rendel hozzá. Mivel f a G gráf egy háromszögmentes színezése, így annak részgráfja is háromszögmentes színezést kap. Azaz H_X egy kvalifikáló gráf, így a w_1, w_2 csúcsok össze lesznek kötve a Γ gráfban. □

2. Megfigyelés. *Ha a Γ segédgráfban van m-klikk. akkor a G gráf csúcsainak van háromszögmentes színezése k színnel.*

Bizonyítás. Tegyük fel, hogy a Γ segédgráfban van m-klikk, Δ. Legyen D a Δ csúcsainak halmaza, és $|D| = m$.

Legyen

$$I_{\{u,v\}} = \{(\{u, v\}, a, b) : \ 1 \leq a, b, \leq k\}$$

minden $\{u, v\} \in E$ esetre. Egyértelmű, hogy $|I_{\{u,v\}}| = k^2$. Vegyük észre, hogy a $I_{\{u,v\}}$, $\{u, v\} \in E$ halmazok páronként függetlenek, és a Γ gráf független halmazai.

Valóban, ha

$$w_1 = (\{u, v\}, a_1, b_1), \qquad w_2 = (\{u, v\}, a_2, b_2)$$

a $I_{\{u,v\}}$ halmaz különböző elemei, akkor a w_1, w_2 csúcsokhoz tartozó H_X gráfnak 2 csúcsa van. Mivel $w_1 \neq w_2$, így nem lehetséges, hogy $a_1 = a_2$, $b_1 = b_2$, azaz H_X nem kvalifikáló. Azaz a Γ gráfban a w_1, w_2 csúcsok nincsenek összekötve.

A Γ segédgráfnak létezik egy legális csúcsszínezése m színnel. Ebben a színezésben a $I_{\{u,v\}}$, $\{u, v\} \in E$ halmazok alkotják a színosztályokat.

Mivel Δ a Γ gráf klikkje, így minden egyes színosztály legfeljebb egy csúcsot tartalmazhat D-ből. Az azonos számosságok miatt levonhatjuk azt a következtetést, hogy a D egy teljes reprezentatív halmaza a színosztályoknak.

Legyen

$$T = \{\{u, v\} : \ (\{u, v\}, a, b) \in D\},$$

amiből az következik, hogy $E = T$. Azaz G minden egyes csúcsa ami egy élen fekszik legalább egy színt kap. Az állításunk az, hogy pontosan egy színt rendelünk hozzá.

Indirekt bizonyítást alkalmazunk. Feltesszük, hogy a G egyik csúcsa több, mint egy színt kap. Ebben az esetben létezik a Δ klikknek két különböző csúcsa, w_1, w_2, ahol a kérdéses csúcs több színt kap a H_X részgráfban, azaz H_X nem kvalifikáló. Másfelől viszont a Γ szerkesztésekor a w_1, w_2 csúcsokat azért kötöttük össze, mert a H_X gráf kvalifikáló volt.

Összefoglalva, úgy szerkesztjük meg az $f : V \to \{1, \ldots, k\}$ függvényt, hogy $f(u) = b$ minden esetben, ahol $(\{u, v\}, a, b)$ a Δ egy csúcsa. Azt kell már csak bizonyítanunk, hogy az f függvény által meghatározott G gráf színezése háromszögmentes.

Tegyük fel, hogy a G gráfban van egy 3-klikk, Ω, melynek a csúcsai ugyanazt a színt kapták. A Δ klikknek van két különböző csúcsa, w_1, w_2, úgy, hogy Ω egy 3-klikk a w_1, w_2 által meghatározott H_X részgráfban. Ebből az következik, hogy H_X nem kvalifikáló. Másfelől viszont a Γ szerkesztésekor a w_1, w_2 csúcsokat azért kötöttük össze, mert a H_X gráf kvalifikáló volt. □

1. Tétel. *Adott G egy gráf, melynek m éle van és a k pozitív egész. Szerkesszük meg a fentiek szerint a Γ segédgráfot. A G gráfnak pontosan akkor és csak akkor van k színnel háromszögmentes színezése, ha a Γ gráfban van egy m-klikk.*

Bizonyítás. következik a 1 és 2 megfigyelésekből. □

3$^{\text{rd}}$ thesis

Miután bemutattuk, hogy milyen klikk-keresők vannak, ezek segédalgoritmusait is bemutattuk. Külön részleteztük, hogy milyen felső becslők vannak a klikkszámra, és ezek eredményeit – mind jóság, mind az algoritmus tapasztalati hatékonysága tekintetében – mérések segítségével összehasonlítottuk. Ugyancsak bemutattunk lehetséges kernelizációs technikákat.

A következő segédalgoritmusokat részleteztük és hasonlítottuk össze az általuk kiszámítható felső becsléseket:

- **A gráf csúcsainak színezése.** Mivel a probléma maga is NP-nehéz, így heurisztikus megközelítéseket alkalmaztunk, a DSatur színezést Brélaztól [Brél1979] és az iterált színezési elvet Culbersontól [Culb1992]. A becslést a $\chi(G) \geq \omega(G)$ képlet adja.

- **b-fold színezés.** b különböző szín hozzárendelése minden egyes csúcshoz úgy, hogy a szomszédos csúcsok színhalmazai függetlenek legyenek.

A korlát $\omega(G) \leq \frac{\chi_b(G)}{b}$. A számításhoz egy segédátfogalmazást használtunk, amely egy $\Gamma = (W, F)$ segédgráf csúcsainak legális színezése volt. A segédgráfot az adott $G = (V, E)$ gráfból szerkesztjük, ahol Γ csúcsai $(v_i, k) \in W, v_i \in V, 1 \leq k \leq b$ rendezett párok. Az élek definíciója:

$$F = \begin{cases} \{(v_i, k), (v_i, l)\} & \text{if } k \neq l & 1 \leq k, l \leq b \\ \{(v_i, k), (v_j, l)\} & \text{if } \{v_i, v_j\} \in E & 1 \leq k, l \leq b \end{cases} \tag{1}$$

- **Élszínezés.** A G gráf éleihez színeket rendelünk [Szab2014c] a következők szerint:

  1. G minden éle pontosan egy színt kap.
  2. Ha x, y, z egy 3-klikk különböző csúcsai G-ben, akkor az $\{x, y\}$, $\{y, z\}$, $\{x, z\}$ élek kötelezően 3 különböző színt kell, hogy kapjanak.
  3. Ha x, y, u, v egy 4-klikk különböző csúcsai G-ben, akkor az $\{x, y\}$, $\{x, u\}$, $\{x, v\}$, $\{y, u\}$, $\{y, v\}$, $\{u, v\}$ élek kötelezően 6 különböző színt kell, hogy kapjanak.

  Legyen Δ egy l-klikk G gráfban, és tegyük fel G élszínezhető k színnel. Ebben az esetben az $l(l-1)/2 \leq k$ egyenlőtlenség fennáll. A G gráf éleit úgy színezhetjük, hogy megszerkesztjük a $\Gamma = (W, F)$ segédgráfot, ahol a G éleit egy-egy csúcs reprezentálja. A Γ csúcsait a fenti szabályt figyelembe véve kötjük össze, azaz Γ két csúcsát összekötjük, ha a nekik megfelelő 3 vagy 4 csúcs G-ben klikket alkot. Könnyű belátni, hogy Γ csúcsainak színezése a G egy élszínezését adja.

- **Lovász theta szám.** Egy valós érték, amely egy gráf Shannon kapacitásának felső becslését adja [Lov1976].

- **Részleges MaxSAT becslés,** A [Li2010a] szerint.

Továbbá a következő kernelizációs technikákat részleteztük:

- **Struction,** A [Ebe1984] szerint.

- **Színindexek.** Amikor egy csúcsot (vagy élt) tudunk törölni, ha az adott csúcs (vagy él) szomszédai által kifeszített részgráf nem színezhető $(k-1)$ (vagy $(k-2)$) színnel.

- **Csúcs dominancia.** Legyen a, b egy G gráf két nem összekötött csúcsa. Azt mondjuk, hogy b csúcs dominálja a csúcsot, ha $N(a) \subseteq N(b)$. Ez esetben az a csúcs törölhető G-ből, ha G-ben k-klikket keresünk.

- **Él dominancia.** Az $\{u, v\}$ él dominálja az $\{x, y\}$ élt, ha $\{u, x\}$ vagy $\{u, y\}$ nem éle G-nek, és $\{v, x\}$ vagy $\{v, y\}$ nem éle G-nek, és $N(x) \cap N(y) \subseteq N(u) \cap N(v)$. Ez esetben az $\{x, y\}$ él törölhető G-ből, ha G-ben k-klikket keresünk.

Negyedik tézis

Megterveztünk és implementáltunk a saját k-klikk kereső programunkat, illetve részleteztük annak hátterét. A program segítségével maximum klikk keresőt is implementáltunk, melyet mérésekkel összehasonlítottunk más, modern klikkeresőkkel, melynek alapján kijelenthetjük, hogy a klikkeresőnk a legjobbak között van. Az algoritmusunk építőelemei a következők:

Elágazás és korlátozás

Jól ismert, hogy a csúcsok színezése a klikkméret felső becslését adja. Így tehát, ha k méretű klikket keresünk, és adott a gráf csúcsainak egy c színnel ($c \geq k$) való színezése, akkor kiválaszthatjuk a legkisebb $c - (k - 1)$ színosztályt, és a bennük lévő csúcsok mentén ágaztathatjuk el a keresést – ez lesz az *elágazási szabály*. Ezen csúcsok halmazát k-klikk fedő csúcshalmaznak nevezzük. Ennek jelentősége az elágazás és korlátozás (Branch and Bound) algoritmusok működéséből adódik. Ezen algoritmusok törlik a már vizsgált elemeket, azaz a már megvizsgált csúcsokat a későbbiek során mér nem veszik figyelembe. Azaz ha az összes fenti csúcs törlésre kerül, akkor a maradék csúcsoknak adott egy $(k - 1)$-színezése, ami azt jelenti, hogy nem lehet a gráfban k-klikk. Vegyük észre, hogy a k értéke nélkül nem alkalmazhatjuk ezt az elágazási szabályt, és az összes csúcs mentén kéne elágaznunk. A k-klikk fedő csúcshalmaz mérete az elágazási faktor, és a lehető legkisebb ilyen halmaz keresése erőteljesen csökkenteni fogja a keresőfa méretét.

Hatékony színezés

A klikk-kereső algoritmusok szakirodalmában közismert, hogy egy jó színezés nagyban gyorsíthatja a klikk-keresési eljárást. Ha a DSatur algoritmust alkalmazzuk a keresőfa minden szintjén, az erőteljesen csökkenti a fa méretét, de ennek túlzottan nagy a költsége. Egy hatékony algoritmust kapunk, ha a költséges DSatur [Brél1979] algoritmust csak a keresőfa tetején alkalmazzuk, és a lentebbi szinteken mér csak az olcsóbb soros színezőt alkalmazzuk. A Dsatur eredményét tovább élesíthetjük a Culberson által ajánlott iterált színezővel [Culb1992]. Ez a technika a színosztályok átrendezése után használja a szekvenciális színezőt. A végeredmény soha nem lehet rosszabb, mint a kiinduló állapot, viszont a gyakorlatban sokszor nagyban javít rajta. Így tehát a keresőfa tetején először egy DSatur színezést végeztünk a gráfon, majd többszörös iterált színezést. A megállási feltétel az volt, ha 1000 iteráció során sem csökken a színosztályok száma.

A csúcsok újraszínezése

A Branch and Bound eljárás során, ahogy minden szinten egyre kevesebb csúcs van a elágazási halmazban, használni tudjuk a fenti szintek színezését, méghozzá a soros színező újrapakolási elve miatt. Minden szinten átrendezzük a színosztályokat nagyság szerint, és a nagyobbal kezdve egy soros színezést hajtunk végre. Mivel a k-klikk fedő csúcshalmaz a legkisebb színosztályokból áll, így éppen ezekből a színosztályokból mozgatunk át egyes csúcsokat a nagyobb színosztályokba – ezzel csökkentve az elágazási faktort. Mivel ezt a szekvenciális színezőt a fa minden csúcsában el kell végezzük, így fontos, hogy ez az algoritmus hatékonyan legyen implementálva. Ehhez bitseteket használtunk, így kapva egy rendkívül gyors eljárást.

Az elágazási csúcsok átrendezése

Korábbi eredményeink [Zav2014a, Zav2014b, Zav2015], ahol a klikk-keresés párhuzamosítását tárgyaltuk, rámutattak arra, hogy az elágazás vizsgálata még inkább fontos az algoritmustervezésben. Az derült ki, hogy a csúcsok sorrendje, ahogy az elágazásban haladunk, rendkívül fontos, és nagy hatással van a keresőfa méretére. Egy nagyon egyszerű átrendezést alkalmaztunk végül. Mindig a legkisebb fokszámú csúcsot vesszük előre, azaz a csúcsok fokszáma szerint növekvő sorrendet alkalmazunk. Így először a feltételezetten legkönnyebb feladatot oldjuk meg, és a csúcs törlése miatt könnyítünk a későbbi feladatokon. Bár az eljárás nagyon olcsó, mégis sokszor rendkívüli mértékben csökkenti a keresőfa méretét.

Ötödik tézis

Miután bemutattuk a párhuzamosítás problémakörét bevezettük a zavaró struktúrák elvét. Ezen elv segítségével elméleti szinten tudtuk megalapozni a k-klikk kereső párhuzamos algoritmusainkat.

Legyen $G = (V, E)$ egy egyszerű, véges gráf, és k egy pozitív egész, továbbá legyen $W \subseteq V$. Ha a G gráf minden k-klikkjének legalább egy csúcsa benne van W-ben, akkor a W halmazt a G gráf k-klikk fedő csúcshalmazának nevezzük. Legyen $W = \{w_1, w_2, \ldots, w_n\}$ egy k-klikk fedő csúcshalmaza a G gráfnak. Tekintsük a G gráf H_i részgráfját, melyet a $N(w_i)$ csúcshalmaz feszít ki minden i, $1 \leq i \leq n$ esetre. Legyen Δ a G gráf egy k-klikkje. A W definíciója szerint a w_i csúcsa kell legyen Δ-nak valamely i, $1 \leq i \leq n$ esetén. Következésképp a H_i pontosan $k-1$ csúcsát tartalmazza a Δ klikknek. Ezen megfigyelésnek egyértelmű a következménye: a k-klikk keresés feladata visszavezethető egy sor $(k-1)$-klikk kereső feladatra a H_i gráfokban minden i, $1 \leq i \leq n$ esetre.

A fenti eredményt ki lehet bővíteni. Legyen $G = (V, E)$ egy egyszerű, véges gráf, és k egy pozitív egész, továbbá legyen F a G gráf összes s-klikk halmazának részhalmaza. Ha a G gráf minden k-klikkjének legalább egy s-klikkje benne van F-ben, akkor az F halmazt a G gráf *k-klikk fedő s-klikk halmaz*ának nevezzük. Nevezetesen az $s = 2$ esetben F egy *k-klikk fedő élhalmaz* lesz. Legyen F a G gráf k-klikk fedő s-klikk halmaza, és legyen

$$c_i = \{u_{i,1}, u_{i,2}, \ldots, u_{i,s}\}, 1 \leq i \leq |F|$$

az összes F béli s-klikk. Legyen H_i olyan részgráf, melyet a következő csúcsok feszítenek ki:

$$H_i = \bigcap_j N(u_{i,j}) \quad 1 \leq j \leq s, \tag{2}$$

és legyen Δ a G gráf egy k-klikkje. Az F definíciója szerint kell, hogy legyen egy c_i csúcshalmaz, ami egy s-klikk Δ-ban valamely i, $1 \leq i \leq |F|$ esetre. Következésképp a H_i gráf pontosan $k-s$ csúcsát tartalmazza Δ-nak. Ezen megfigyelésnek egyértelmű a következménye: a k-klikk keresés feladata visszavezethető egy sor $(k-s)$-klikk kereső feladatra a H_i gráfokban minden i, $1 \leq i \leq n$ esetre.

Hatodik tézis

A fenti elvek alapján implementáltunk egy nagyléptékben párhuzamos k-klikk kereső algoritmust. Megvizsgáltuk a részproblémák sorrendjének jelentőségét. Ezen alapulva bevezettük a Las Vegas párhuzamosítási elvet, amely jelentősen csökkentette a keresőteret. Az eredeti sorrend szerint 4 nagyságrendben különböztek a részproblémák nehézségei, mely különbséget nagyban sikerült csökkenteni.

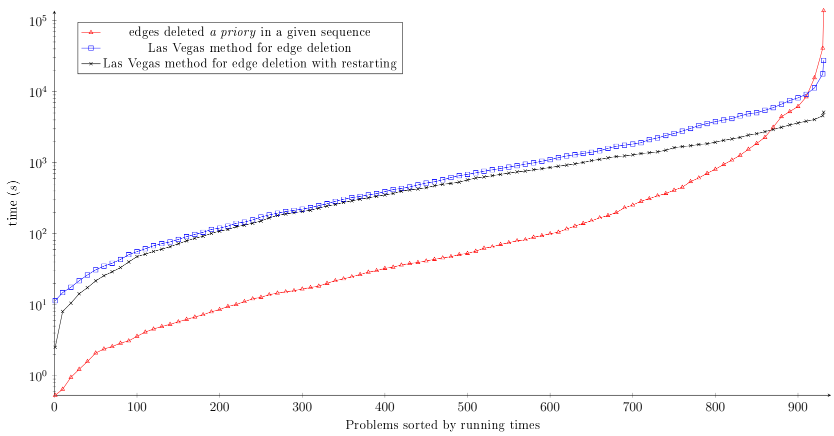
A következő három módszert hasonlítottuk össze:

1. Az eredeti *a priory* sorrend szerinti törlés, ahol az élek egy előre fixált sorrendjében töröljük a (sorrend szerinti) korábbi éleket a (sorrend szerinti) későbbi részfeladatokból.

2. A Las Vegas módszer, ahol nem törlünk éleket a kezdeti részfeladatokból. A részfeladatokat párhuzamosan kezdjük el megoldani, és ha valamely részfeladat lefutott, akkor ezt a részfeladatot reprezentáló élt töröljük az összes többi részfeladatból – mind a még várakozókból, mind a már futók esetén.

3. A Las Vegas módszer újraindítással, amely ugyanúgy indul, mint az előző, és csak a legvégén különbözik, amikor már majdnem minden részproblémát megoldottunk, és már kevesebb részprobléma fut, mint ahány processzor van. Ebben az esetben újraindítunk egy részproblémát, méghozzá futásban tartva az eredeti indulását is, azaz ugyanazt a részproblémát két program is egyszerre old meg. Az egyiknek az az előnye, hogy már félig bejárta a keresőfát, miközben a másik előnye az, hogy jobb prekondicionálást tud elérni az újraindítás miatt.

Közelről megvizsgáltunk egy, a `monoton-9` problémát. Az 1. ábrán átrendeztük a részproblémákat a megoldási idejük szerint. Az időskála logaritmikus, így a különbségek több nagyságrendet ölelnek át. Ezen az ábrán látszik, hogy mind a három megközelítésben a leghosszabban futó probléma dominál, de a Las Vegas módszer újraindítással tudja ezt a legjobban kiegyenlíteni, és a futásidők különbségét 2 nagyságrend méretűre csökkenteni.



1. ábra. A `monoton-9` részproblémái idő szerint sorrendbe állítva.

A szerző saját eredményei

1. A Costas táblázat átfogalmazása az első tézisben és 2.1.4. fejezetben saját eredmény. Még nem publikált.

2. Az ütemezési feladatok átfogalmazása az első tézisben és 2.3. fejezetben közös eredmény Szabó Sándorral. Még nem publikált.

3. A 3-free színezés k-klikk megközelítése a második tézisben és 3.2. fejezetben saját eredmény. A [Szab2016b] lett publikálva.

4. A hipergráfok színezés k-klikk megközelítése, illetve Voloshin által felvetett problémának a számolása a második tézisben és 3.3. fejezetben saját eredmény. A [Szab2019b] lett publikálva.

5. A klikk méretének felső becsléseinek összehasonlítása a harmadik tézisben és 4.2. fejezetben, beleértve a soros és párhuzamos DSatur és ismételt színező algoritmusok implementálását, illetve az él és b-fold színezésekhez való használata saját eredmény. A [Szab2017, Marg2019] lett publikálva.

6. A negyedik tézisben és 5. fejezetben bemutatott soros k-klikk keresőt először Szabó Sándor fejlesztette ki, de később a szerző jelentősen kibővítette az ismételt színező, csúcsátrendezés illetve bitsetekkel való párhuzamosítás fejlesztésével. Továbbá a szerző szerkesztette ennek segítségével a maximum klikk kereső algoritmust és programot. A [Szab2018a] lett publikálva.

7. A „zavaró struktúrák” elméleti háttere és az ezen alapuló párhuzamosítási elvek az ötödik tézisben és 7. fejezetben főképp a szerző eredménye. A [Szab2018b] lett publikálva.

8. A hatodik tézisben és 8. fejezetben részletezett párhuzamos program Szabó Sándor ötletén alapulva a szerző fejlesztése. A Las Vegas párhuzamosítási elv, illetve a mérések, melyek a megközelítést megalapozták saját eredmény. A [Zav2014a, Zav2014b, Zav2015] lett publikálva.

Hivatkozások

[Aki2016] Akiba, T. and Iwata, Y. "Branch-and-reduce exponential/fpt algorithms in practice: A case study of vertex cover." *Theoretical Computer Science.* 609:211–225, 2016.

[Brél1979] Brélaz, D. "New Methods to Color the Vertices of a Graph." *Communications of the ACM.* 1979. 22. 4. pp. 251–257.

[Cost1965] Costas, J.P. *Medium constraints on sonar design and performance.* Class 1 Report R65EMH33, G.E. Corporation. 1965.

[Culb1992] Culberson, J.C. *Iterated Greedy Graph Coloring and the Difficulty Landscape.* Technical Rep. University of Alberta. 1992.

[Ebe1984] Ebenegger, Ch. Hammer, P.L. and de Werra, D. "Pseudo-boolean functions and stability of graphs." *North-Holland mathematics studies.* volume 95, pp. 83–97. 1984.

[Karp1972] Karp, R.M. *Reducibility Among Combinatorial Problems.* In: Complexity of Computer Computations. Eds: R. E. Miller and J. W. Thatcher. New York. Plenum. pp. 85–103.

[Li2010a] Li, C.-M. and Quan, Z. *An Efficient Branch-and-Bound Algorithm Based on MaxSAT for the Maximum Clique Problem.* In: Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence. (AAAI-10), pp. 128–133.

[Lov1976] Lovasz L. "On the Shannon Capacity of a Graph," *IEEE Transactions on Information Theory.* 25, 1, 1979. pp. 1–7.

[Marg2019] Margenov, S. et al. *Applications for ultrascale systems.* In: Ultrascale Computing Systems. Institution of Engineering and Technology (IET), London, (2019) pp. 189–244.

[Szab2012] Szabó S. and Zaválnij B. "Greedy Algorithms for Triangle Free Coloring." *AKCE Int. J. Graphs Comb.,* 9, No. 2 (2012), pp. 169–186.

[Szab2014a] Szabó S. and Zaválnij B. "Coloring the edges of a directed graph." *Indian Journal of Pure and Applied Mathematics.* April 2014, Volume 45, Issue 2, pp 239–260.

[Szab2014b] Szabó S. and Zaválnij B. "Coloring the nodes of a directed graph." *Acta Univ. Sapientiae, Informatica.* 6, 1 (2014) 117—131.

[Szab2014c] Szabó S. and Zaválnij B. "Estimating clique size via coloring edges in graphs." *AKCE Int. J. Graphs Comb.* (submitted.)

[Szab2016b] Szabó S. and Zaválnij B. "Reducing Graph Coloring to Clique Search," *Asia Pacific Journal of Mathematics*, 3 (2016), pp. 64–85.

[Szab2017] San Segundo, P., Szabó S. and Zaválnij B. "Parallelization of the clique search problem using sub-chromatic bounds." NESUS. Technical Report. January 17, 2017.

[Szab2018a] Szabo S. and Zavalnij B. *A different approach to maximum clique search.* In: 20th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing. (SYNASC2018) IEEE Proceedings, (2018) pp. 310–316.

[Szab2018b] Szabo S. and Zavalnij B. "Decomposing clique search problems into smaller instances based on node and edge colorings." *Discrete Applied Mathematics.* 242 (2018) pp. 118–129.

[Szab2019a] Szabo S. and Zavalnij B. "Benchmark Problems for Exhaustive Exact Maximum Clique Search Algorithms." *Informatica (Ljubljana).* 43 : 2 (2019) pp. 177–186.

[Szab2019b] Szabo S. and Zavalnij B. "Reducing hypergraph coloring to clique search." *Discrete Applied Mathematics.* 264. (2019) pp. 196–207.

[Zav2014a] Zaválnij B. "Three Versions of Clique Search Parallelization." *Journal of Computer Science and Information Technology.* Vol. 2:(No. 2) pp. 9–20. (2014)

[Zav2014b] Zavalnij, B. *The Las Vegas method of parallelization.* In: Information Society 2014 – IS 2014: Volume A; Intelligent Systems. pp. 105–108. 2014.

[Zav2015] Zavalnij, B. *Speeding up Parallel Combinatorial Optimization Algorithms with Las Vegas Method.* In: 10th International Conference on Large-Scale Scientific Computations. Lecture Notes in Computer Science (LNCS) 2015.