GRÁFOK FAFELBONTÁSAI ÉS HATÁSUK AZ ALGORITMIKUS BONYOLULTSÁGRA

MTA DOKTORI ÉRTEKEZÉS TÉZISEI

Marx Dániel

MTA Számítástechnikai és Automatizálási Kutatóintézet
Budapest

2019

1. Bevezetés

Számos alkalmazási terület igényel hatékony algoritmusokat kombinatorikus feladatok megoldására. A gyakorlatban felmerülő optimalizálási problémák jelentős része NP-teljes, ezért nem várhatjuk, hogy polinomidejű algoritmusokat találhatunk ezek megoldására. Nem remélhetjük, hogy ezeket a problémákat teljes általánoságban hatékonyan tudjuk kezelni, de lehetséges, hogy léteznek olyan elméleti vagy gyakorlati jelentőségű speciális esetek, amelyeket bizonyíthatóan hatékonyan meg lehet oldani. Az algoritmuselméleti kutatások jelentős része olyan speciális eseteket vagy algoritmikus szempontból kedvező tulajdonságokat vizsgál, amelyek hatékony algoritmusokat tesznek lehetővé.

Talán a legszélesebb körben vizsgált kedvező tulajdonság a probléma *felbonthatósága:* a feladat bemenete rekurzív módon felbontható kisebb részproblémákra olyan módon, hogy a részproblémák megoldásából összeállítható az eredeti probléma egy megoldása. A felbonthatóság pontos definíciója függ a konkrét problémától, de tipikusan valami olyasmit jelent, hogy találhatunk kisméretű szeparátorokat, ami a problémát egymástól nagyjából független részekre bontja és bármilyen kapcsolat ezen részek között csak a szeparátoron keresztül történhet. A gráfokon értelmezett algoritmikus problémák esetén (illetve olyan problémákon, amiket természetes módon lehet gráfokkal modellezni) rendkívül hasznos a *favastagság* fogalma a felbonthatóság fogalmának a modellezésére. Durván fogalmazva, egy gráf favastagsága azt méri, hogy a gráf mennyire hasonlít a fa struktúrára: alacsony favastagságú gráfok nagyjából úgy néznek ki, mintha egy fa gráfban minden csúcsot kicserélnénk valamilyen kis gráfra. A favastagság formális definíciója (lásd alább) elsőre nem tűnik túl szemléletesnek, de nagyon pontosan kifejezi azokat a feltételeket, amelyek a kis szeparátorokon történő rekurzív felbontáson alapuló algoritmusok igényelnek. A favastagság fogalmának a természetességét az is mutatja, hogy ez a definíció (ekvivalens megfogalmazásban) egymástól függetlenül háromszor is megjelent az irodalomban [6, 45, 69].

Az értekezés átfogó témája annak megértése, hogy a favastagság és hozzá hasonló strukturális paraméterek hogyan befolyásolják bizonyítható módon a problémák bonyolultságát. A bemutatott eredmények között vannak algoritmusok (bizonyítható felső korláttal a futási időre) és bonyolultsági eredmények (bizonyítható alsó korlátok a probléma megoldásához szükséges futási időre). Az értekezés arra az alapvető kérdésre próbál válasz találni, hogy a probléma bonyolultsága mennyire pontosan kapcsolódik a favastagsághoz. Vizsgáljuk, hogy önmagában az alacsony favastagság ténylegesen csökkenti-e a probléma bonyolultságát, illetve hogy pontosan számszerűsíthető-e, hogy adott favastagságnál milyen futási idő érhető el. Továbbá vizsgáljuk azt a kérdést, hogy vajon a favastagság (esetleg annak valamilyen variánsa) az egyetlen gráfelméleti tulajdonság, ami a probléma bonyolultságát csökkenti. Noha ez egy nagyon általános és nehezen megválaszolható kérdésnek tűnik, az irodalomban vannak példák arra, hogy lehetséges formális választ adni ilyen típusú általános kérdésekre is [20,21,28,40,43]. Az értekezés bemutat ilyen eredményeket, amelyek egyrészt pontosabb korlátokat adnak a korábbi eredményeknél, másrészt kiterjesztik őket pl. hipergráfokra is, ahol teljesen más technikák és eredmények jelennek meg.

A favastagság formális definíciójához szükségünk van a gráfok egy speciális felbontásának a definíciójára. Egy G gráf *fafelbontása* egy $(\mathcal{B}, T)$ pár, ahol T egy fa (körmentes összefüggő gráf) és $\mathcal{B} = \{B_t \mid t \in V(T)\}$ a $V(G)$ csúcshalmaz részhalmazainak egy családja, a következő tulajdonságokkal:

- $\bigcup_{t\in V(T)} B_t = V(G)$,
- minden $xy \in E(G)$ élre $\{x, y\} \subseteq B_t$ teljesül valamely $t \in V(T)$ csúcs esetén;
- minden $x \in V(G)$ csúcsra a $\{t \mid x \in B_t\}$ halmaz a T fa egy összefüggő részfáját feszíti.

A fafelbontás *vastagsága* $\max_{t\in V(T)}\{|B_t| - 1\}$ és a G gráf *favastagsága* $\mathrm{tw}(G)$ a G gráf összes lehetséges fafelbontásának a vastagságának a minimuma. A fenti definícióban a -1 pusztán normalizációs célokat szolgál: ezáltal $\mathrm{tw}(G) \leq 1$ akkor és csak akkor ha a G gráf minden komponense fa. A favastagság pontos értékének meghatározása NP-nehéz [3], de minden rögzített $k \geq 1$ értékre létezik lineárisidejű algoritmus, amely a G gráfnak egy k vastagságú felbontást találja meg ha $\mathrm{tw}(G) \leq k$ teljesül [8]. Léteznek algoritmusok az optimálishoz közeli felbontások keresésére [2, 10].

2. Optimális algoritmusok fafelbontásokon

Az 2. fejezet azzal a kérdéssel foglalkozik, hogy a favastagság illetve egy adott vastagságú fafelbontás hogy befolyásolja különböző klasszikus gráfelméleti problémák bonyolultságát. Ismeretes, hogy ha a bementi gráf favastagsága k, akkor számos algoritmikus probléma megoldható $f(k)\cdot n^c$ időben, ahol $f(k)$ egy csak k-tól függő függvény és a c kitevő egy k-tól független konstans. A paraméteres bonyolultság nyelvén az ilyen futási idő esetén azt mondjuk, hogy a probléma FPT (fixed-parameter tractable) a k (favastagság) értékével parametrizálva [25, 29, 30, 35]. A legtöbb esetben, például a Független Halmaz, 3-Szín, Domináns Halmaz, vagy Hamilton Kör problémáknál a dinamikus programozás módszere kézenfekvő módon ad ilyen algoritmusokat. Annak ellenére, hogy ezek a dinamikus programozást használó algoritmusok tervezése egy jól ismert sémát követ, sok esetben körülményes a használata és a részproblémák pontos definiálása hosszú és bonyolult bizonyításokat eredményez. Sok esetben használható Courcelle eredménye [19], amely kimondja, hogy *minden* olyan algoritmikus probléma, ami a kiterjesztett monadikus másodrendű logikában kifejezhető egy ϕ formulával, megoldható $f(k, \phi) \cdot n$ időben k favastagságú gráfokon.

Courcelle eredményével könnyen kaphatunk $f(k)\cdot n$ idejű algoritmust például a 3-Szín problémára, de ezzel a módszerrel nem kapunk jó felső korlátott az $f(k)$ függvény nagyságrendjére. Számos esetben a problémára szabott egyedi módszerekkel sokkal jobb korlátot kaphatunk az $f(k)$ függvényre. A Független Halmaz problémára még elég egyszerű $2^k \cdot n^{O(1)}$ idejű algoritmust adni, de a Domináns Halmaz problémára a kézenfekvő módszerek csak egy $4^k \cdot n^{O(1)}$ idejű algoritmust adnak és a $3^k \cdot n^{O(1)}$ idő eléréshez egy új nemtriviális módszert (a gyors részhalmaz konvolúciót) kellett felhasználni [7, 70]. Hasonló módon, a Hamilton Kör problémára a kézenfekvő dinamikus programozást használó algoritmus $2^{O(k \log k)} \cdot n^{O(1)}$ időt igényel; a gyorsabb $c^k \cdot n^{O(1)}$ idejű algoritmusokhoz új technikákat kellett bevezetni [9, 26, 27].

Látva, hogy bizonyos algoritmikus problémáknál a futási idő favastagságtól való függését csak új technikák bevezetésével lehetett javítani, míg más problémáknál semmilyen javulást nem sikerült elérni, adódik a kérdés: mi a lehetséges legjobb $f(k)$ függvény amit elérhetünk egy adott problémánál? Az 2. fejezet ezt a kérdést vizsgálja és számos esetben éles alsó korlátot ad az elérhető legjobb $f(k)$ függvényre.

Az alsó korlátok természetesen feltételesek: ha P = NP (amit formálisan nem zárhatunk ki), akkor a vizsgált problémák mind megoldhatók polinomidőben. Ezért az összes ilyen jellegű negatív eredmény valamilyen bonyolultságelméleti hipotézisen alapszik. Kézenfekvő lenne a P $\neq$ NP feltételezést alapul venni, de úgy tűnik, ez önmagában nem elég erős éles alsó korlátok bizonyításához. Az elmúlt évtizedekben számos új bonyolultságelméleti hipotézis megfogalmaztak és az ezekre alapozott feltételes bonyolultsági eredmények nagymértékben hozzájárultak az algoritmikus problémák bonyolultságának pontos megértéséhez. Impagliazzo, Paturi és Zane [47, 48] vezették be az Erős Exponenciális

Idő Hipotézist (Strong Exponential-Time Hypothesis, SETH), amely lényegében azt állítja, hogy a konjunktív normálformájú Boole formulák kielégíthetőségét vizsgáló SAT probléma n változó és m tag esetén nem oldható meg $(2-\epsilon)^n \cdot m^{O(1)}$ időben semmilyen $\epsilon > 0$ konstansra. Ez azt jelenti, hogy az n változó összes lehetséges 2^n behelyettesítésének a kipróbálásánál nincs lényegesen jobb algoritmus. Az 2. fejezet fő eredménye ezen a hipotézisen alapszik:

1. tétel (Theorem 2.1 az értekezésben). Ha SETH igaz, akkor minden $\epsilon > 0$ konstansra teljesül, hogy

- Független Halmaz nem oldható meg $(2-\epsilon)^{\mathrm{tw}(G)} \cdot n^{O(1)}$ időben,
- Domináns Halmaz nem oldható meg $(3-\epsilon)^{\mathrm{tw}(G)} \cdot n^{O(1)}$ időben,
- Maximális Vágás nem oldható meg $(2-\epsilon)^{\mathrm{tw}(G)} \cdot n^{O(1)}$ időben,
- Páratlan Körök Lefedése nem oldható meg $(3-\epsilon)^{\mathrm{tw}(G)} \cdot n^{O(1)}$ időben,
- q-Színezés nem oldható meg $(q-\epsilon)^{\mathrm{tw}(G)} \cdot n^{O(1)}$ időben,
- Háromszög Partíció nem oldható meg $(2-\epsilon)^{\mathrm{tw}(G)} \cdot n^{O(1)}$ időben.

Ezek az eredmények azt mutatják, hogy (a SETH feltételezéssel) az ismert algoritmusok optimálisak és nem várható további javulás. Vagyis nem érdemes ilyen irányú javításba munkát fektetni, pontosabban bármilyen erre törekvő munka a SETH megcáfolásával lenne egyenértékű. A fenti eredmények adták az első éles alsó korlátokat a favastagsággal való parametrizálásra, amelyeket további hasonló eredmények követtek [14, 22, 24, 27, 31, 49, 50].

A fejezetben publikált eredmények egy *SIAM Journal on Computing* folyóiratban megjelent cikken alapszik és Daniel Lokshtanovval és Saket Saurabh-al közös munka [57]. A cikk előzetes változata a SODA 2011 konferencia kötetében jelent meg [56].

3. Favastagság hatása a korlátkielégítési problémák bonyolultságára

A 3. fejezet azt vizsgálja, hogy a korlátkielégítési problémák (Constraint Satisfaction Problem, CSP) esetén a feltételek által alkotott gráf struktúrája hogyan hat a probléma algoritmikus bonyolultságára. Ismeretes, hogy alacsony favastagság esetén a probléma hatékonyan megoldható, a kérdés, hogy léteznek-e más hasonló gráfelméleti tulajdonságok, amelyek csökkentik a probléma bonyolultságát.

A korlátkielégítési feladatok egy nagyon általános problémaosztályt alkotnak, speciális esetként tartalmazva számos klasszikus algoritmikus problémát pl. a formula-kielégíthetőség, gráfszínezések, vagy az adatbázis lekérdezések területéről. Formálisan, a CSP probléma egy $I = (V, D, C)$ példánya leírható a változók egy V halmazával, a változók értékkészletével és a korlátok C halmazával. Minden $c_i \in C$ korlát egy $\langle s_i, R_i \rangle$ párral írtható le, ahol s_i a változók egy rendezett m_i hosszú listája és R_i egy m_i aritású reláció R felett. Az $\langle s_i, R_i \rangle$ korlátban az R_i reláció sorolja fel azokat az értékkombinációkat amelyek az s_i listában szereplő változókon egyszerre engedélyezett. A CSP probléma megoldása egy olyan f függvény, amely minden V-beli változóhoz egy D-beli értéket rendel és minden $\langle s_i, R_i \rangle$ korlátra teljesül, hogy ha $s_i = (v_{i_1}, v_{i_2}, \ldots, v_{i_m})$ akkor $(f(v_{i_1}), f(v_{i_2}), \ldots, f(v_{i_m}))$ kielégíti az R_i relációt. A CSP eldöntési változatában egy adott $I = (V, D, C)$ bemenet esetén kell eldönteni, hogy létezik-e megoldás. Például a 3SAT probléma értelmezhető egy olyan CSP feladatnak, ahol a változók értékkészlete $D = \{0, 1\}$ és a korlátok a 3CNF formula tagjainak felelnek meg (vagyis minden korlát ternáris). A gráfok csúcsszínezési problémája is értelmezhető CSP feladatként: a változók a csúcsoknak felelnek meg, az értékkészlet a felhasználható színek halmaza és minden élnek egy bináris „nem egyenlő” korlát felel meg.

A (V, D, C) feladat G Gaifmann gráfját a következőképpen definiáljuk: a csúcsok $V(G)$ halmaza a változók V halmaza és két $x, y \in V$ változó között akkor van él, ha létezik egy olyan korlát, amely mindkettőt tartalmazza, vagyik létezik egy $\langle s_i, R_i \rangle \in C$ melyre $x, y \in s_i$ teljesül. Binárisnak nevezzük azokatat a CSP feladatokat ahol minden korlát egy bináris reláció, vagyis minden s_i lista pontosan két változót tartalmaz. Bináris CSP esetén minden korlátnak egy él felel meg a Gaifmann gráfban. Általánosabban, ha nagyobb aritású korlátok is vannak, akkor egy k változót tartalmazó korlátnak egy k méretű klikk fog megfelelni a Gaifmann gráfban.

Freduer [36] megmutatta, hogy ha egy CSP feladat Gaifmann gráfjának a favastagsága k, akkor a probléma megoldható $n^{O(k)}$ időben. Vagyis ha a CSP feladatoknak egy olyan osztályát nézzük, ahol a Gaifmann gráf favastagsága korlátos, akkor Freuder eredménye alapján ez az osztály polinom időben megoldható. Felmerül a kérdés, hogy a korlátos favastagságon kívül létezik-e más olyan gráfelméleti tulajdonság, amely polinomidejű megoldhatóságot garantál. Formálisan, legyen $\mathcal{G}$ gráfoknak egy tetszőleges osztálya; a CSP($\mathcal{G}$) feladat az általános CSP feladatnak azon megszorítása, ahol a bemenet Gaifmann gráfja egy $\mathcal{G}$

osztályba tartozó gráf. Freuder eredményéből következik, hogy ha $\mathcal{G}$ korlátos favastagságú (vagyis létezik olyan c konstans, hogy $\mathrm{tw}(G) \leq c$ minden $G \in \mathcal{G}$ gráf esetén), akkor $\mathrm{CSP}(\mathcal{G})$ polinom időben megoldható.

2. tétel (Freuder [36]). Ha $\mathcal{G}$ korlátos favastagságú, akkor $\mathrm{CSP}(\mathcal{G})$ polinom időben megoldható.

Egy meglepően általános eredmény kimondja, hogy korlátos favastagság az egyetlen olyan tulajdonság, ami $\mathrm{CSP}(\mathcal{G})$ polinomidejű megoldhatóságot garantálja.

3. tétel (Grohe, Schwentick, Segoufin [40, 43]). Legyen $\mathcal{G}$ egy tetszőleges gráfosztály és tételezzük fel az $\mathrm{FPT} \neq \mathrm{W}[1]$ bonyolultságelméleti hipotézist. A $\mathrm{CSP}(\mathcal{G})$ probléma polinomidőben megoldható akkor és csak akkor ha $\mathcal{G}$ korlátos favastagságú.

A $\mathrm{FPT} \neq \mathrm{W}[1]$ hipotézis a paraméteres bonyolultság standard feltételezése [25, 29, 30, 35]. A fenti eredmény teljesmértékben megválaszolja a polinomidőben való megoldhatóság kérdését: ha $\mathcal{G}$ egy *tetszőleges* gráfosztály nemkorlátos favastagsággal, akkor nem létezik polinomidejű algoritmus a $\mathrm{CSP}(\mathcal{G})$ problémára. Bármilyen általános is ez az eredmény, nem ad viszont választ arra a kérdésre, hogy létezik-e olyan gráfosztály, ahol Freuder $n^{O(k)}$ algoritmusánál lényegesen hatékonyabb módszer létezik. Elképzelhető, hogy van olyan $\mathcal{G}$ gráfosztály, ahol $\mathrm{CSP}(\mathcal{G})$ megoldható $n^{O(\sqrt{k})}$ vagy akár $n^{O(\log k)}$ időben, ahol k a Gaifmann gráf favastagsága. A második fejezet fő eredménye erre a kérdésre ad negatív választ: akármilyen $\mathcal{G}$ gráfosztályra szorítjuk meg a $\mathrm{CSP}(\mathcal{G})$ problémát, Freuder $n^{O(k)}$ idejű algoritmusa egy logaritmikus faktortól eltekintve optimális. Az eredményt az Exponenciális Idő Hipotézis (Exponential-Time Hypothesis, ETH) feltételezésével bizonyítjuk: ez a hipotézis lényegében azt állítja, hogy az n-változós 3SAT probléma nem oldható meg $2^{o(n)}$ időben.

4. tétel (Theorem 3.2 az értekezésben). Legyen $\mathcal{G}$ egy tetszőleges gráfosztály és tételezzük fel az ETH bonyolultságelméleti hipotézist. Nem létezik a $\mathrm{CSP}(\mathcal{G})$ problémára olyan algoritmus, aminek a futási ideje $f(G)n^{o(\mathrm{tw}(G)/\log \mathrm{tw}(G))}$ ahol G a bemenet Gaifmann gráfja és f egy tetszőleges függvény.

A fenti tétel kimondja, hogy Freuder $n^{O(\mathrm{tw}(G))}$ idejű algoritmusa lényegében optimális, eltekintve egy legfeljebb logaritmikus tényezőtől a kitevőben. Továbbá levezethető ebből az eredményből egy önmagában is érdekes és fontos következmény. Ha egy paraméteres probléma nem FPT (vagyis nem oldható meg $f(k)n^c$ időben valamilyen f függvény és c konstans mellett), akkor a k paraméternek valamilyen formában szerepelnie kell a futási idő kitevőjében valamilyen formában. A szokásos W[1]-nehézségi eredmények csak arra nyújtanak bizonyítékot, hogy a probléma nem FPT, de nem mondanak semmit arról, hogy a paraméternek milyen formában kell szerepelnie a kitevőben. Számos vizsgált W[1]-nehéz paraméteres probléma egyszerűen megoldható $n^{O(k)}$ időben (esetleg pusztán nyers erő alkalmazásával); felmerül a kérdés, hogy létezhet-e ezekre lényegesen jobb, pl. $n^{O(\sqrt{k})}$ vagy akár $n^{O(\log\log k)}$ idejű algoritmus. Az első ilyen irányű kvantitatív eredményt Chen et al. bizonyította a Klikk problával kapcsolatban:

5. tétel (Chen et al. [16, 17]). Ha ETH igaz, akkor a Klikk problémára nem létezik $f(k)n^{o(k)}$ idejű algoritmus semmilyen f függvényre.

Paraméteres visszavezetésekkel az 5. tételhez hasonló alsó korlátokat kaphatunk más problémákra is. A kapott alsó korlát erőssége azon múlik, hogy a visszavezetés során a paraméter értékre hogyan változik. Ha a visszavezetés csak linárisan növeli meg a paraméter értékét, akkor a célproblémánál is kizárhatjuk az $f(k)n^{o(k)}$ idejű algoritmusokat (ha ETH igaz). De az is tipikus, hogy a k méretű klikk kereséséről történő visszavezetés esetén a célproblémába valamilyen módon reprezentáljuk a megtalálandó k csúcsot és a köztük lévő $\binom{k}{2}$ élet, ezáltal az új paraméter értéke $O(k^2)$ nagyságrendű. Ez esetben a visszavezetés 5. tétellel együtt csak azt a gyengébb eredményt adja, hogy nem létezhet $f(k)n^{o(\sqrt{k})}$ idejű algoritmus.

Sok esetben az ilyen alsó korlát nem éles: a legjobb ismert algoritmus $n^{O(k)}$ idejű. A Klikk problémáról történő visszavezetés általában kis munkával módosítható egy Részgráf Izomorfia problémáról történő visszavezetésre: ebben a problémában a bemenet két gráf, G és H, az eldöntendő kérdés, hogy G-nek van-e H-val izomorf részgráfja. Vagyis a Klikk probléma a Részgráf Izomorfia azon speciális esete, ahol H egy teljes gráf. Technikai okokból sokszor érdemesebb a Particionált Részgráf Izomorfia problémát vizsgálni, ahol a G gráf csúcshalmaza $|V(H)|$ osztályra van partícionálva és G-nek egy olyan részgráfját keressük, amely minden osztályból pontosan egy csúcsot tartalmaz. A 4. tételből levezethető a következő alsó korlát:

6. tétel (Corollary 3.4 az értekezésben). Ha ETH igaz, akkor nem létezik a Particionált Részgráf Izomorfia problémára $f(k)n^{o(k/\log k)}$ idejű algoritmus, ahol k a H gráf *élszáma* és f egy tetszőleges függvény.

A legtöbb Particionált Részgráf Izomorfia problémáról történő visszavezetésnél az célprobléma paraméterének értéke a H gráf éleinek számával arányos. Vagyis egy ilyen visszavezetés 6. tétellel együtt kizárja a $f(k)n^{o(k/\log k)}$ idejű algoritmus létezését a célproblémára (ha ETH igaz). Ez a technika számos $n^{O(k)}$ időben megoldható problémára alkalmazható hogy kizárjuk az $f(k)n^{o(k/\log k)}$ idejű algoritmusokat és ezzel majdnem éles alsó korlátot kapjunk [11–13, 15, 20, 23, 32, 33, 44, 51, 53, 58, 65, 68]. A legtöbb esetben nem ismert más módszer, amivel ennyire éles alsó korlátot kaphatunk ezekre a problémákra.

A fejezetben közölt eredmények egy *Theory of Computing* folyóiratban megjelent egyszerzős cikkben lettek közölve [62]. A cikk előzetes változata a FOCS 2007 konferencia kötetében jelent meg [59].

4. Frakcionális élfedés

Az előző fejezet pontos választ ad arra, hogy a Gaifmann gráf favastagsága hogyan befolyásolja a CSP feladatok bonyolultságát. Ha bináris CSP feladatokról beszélünk (vagyis minden korlát két változót érint), akkor a Gaifmann gráf elég pontosan leírja a korlátok struktúráját. Ez bizonyos értelemben igaz akkor is, ha minden korlát konstans sok (pl. 3) változót tartalmaz. De ha egy korlátban tetszőleges számú változó szerepelhet, akkor a Gaifmann gráf nem tartalmaz minden információt a korlátok struktúrájáról. Egyszerű példaként tekintsük azt az esetet, amikor Gaifmann gráf egy n csúcsú teljes gráf. Ez előállhat úgy is, hogy a feladat egyetlen n-áris korlátot tartalmaz az összes változón, vagy úgy is, hogy a feladat $\binom{n}{2}$ bináris relációt tartalmaz páronként az összes reláción. Pontosabb információt kapunk a korlátok struktúrájáról, ha egy $I = (V, D, C)$ CSP feladat H *hipergráfját* vizsgáljuk: a csúcsok $V(H)$ halmaza megfelel a változók V halmazának és minden $\langle s_i, R_i \rangle \in C$ korlátnak megfelel egy hiperél, amely az s_i-beli változókat tartalmazza. Hasonlóan a CSP($\mathcal{G}$) definíciójához, ha $\mathcal{H}$ hipergráfoknak egy tetszőleges osztálya, akkor CSP($\mathcal{H}$) a CSP probléma megszorítása olyan feladatokra, ahol a korlátok hipergráfja a $\mathcal{H}$ osztályba esik.

Ha egy korlát tetszőleges számú változót érinthet, akkor nem kerülhetjük el azt a kérdést, hogy milyen módon vannak reprezentálva a korlátok a bemenetben. Mivel egy ilyen korlát exponenciális sok érték-kombinációt enged meg/tilt, a korlát leírására szolgáló reprezentáció választása jelentősen tudja növelni a bemenet n méretét. Mivel a futási időben a bemenet mérete is megjelenik, a reprezentációtól függően változhat a feladat bonyolultsága.

Több természetes módja van egy reláció reprezentálásának: igazságtábla, konjunktív vagy diszjunktív normálforma, vagy egyszerűen a relációban szereplő listák felsorolása. Az értekezés ezek közül a legutolsó reprezentációt tételezi fel: ha $\langle s_i, R_i\rangle \in C$ egy olyan korlát ahol $R_i \subseteq D^{m_i}$ egy m_i-áris reláció, akkor feltételezzük, hogy a bemenetben fel van sorolva az összes olyan m_i hosszúságú rendezett m_i lista, ami kielégíti ezt a korlátot. Ezen rendezett listák száma legfeljebb D^{m_i}, de ennél lényegesen kevesebb is lehet olyan korlátoknál, ahol csak kevés kielégítő kombináció van. A továbbiakban $\|I\|$ fogja jelölni az $I = (V, D, C)$ példány ilyen formájú reprezentációjának a hosszúságát. Ez a reprezentáció különösen természetes relációs adatbázisok elméleti vizsgálatánál, ahol egy n-es akkor elégít ki egy korlátot, ha szerepel egy relációs tábla soraként. A relációs táblák fizikailag soronként vannak tárolva, ezért a vizsgálandó adatbázis (bemenet) mérete a relációt kielégítő n-esek számától függ.

Az előző bekezdésben tárgyalt reprezentáció mellett léteznek olyan hipergráf struktúrák, ahol a hatékony megoldhatóság nem a korlátos favastagság következménye. A H hipergráf *élfedési száma* $\rho(H)$ a legkisebb élhalmaz mérete amely lefedi az összes csúcsot. Ismeretes, hogy ha egy CSP feladat H hipergráfjára igaz, hogy $\rho(H) \leq k$, vagyis az összes csúcs k éllel lefedhető, és minden relációt legfeljebb N különböző kombináció elégít ki, akkor legfeljebb N^k különböző megoldás létezik és ezek felsorolhatók $\|I\|^{k+O(1)}$ időben. Ebből következik, hogy ha $\mathcal{H}$ egy korlátos élfedési számú hipergráfosztály, akkor CSP($\mathcal{H}$) polinom időben megoldható.

7. tétel (Gottlob et al. [39]). Ha egy CSP feladat hipergráfja H és minden feltétel reláció legfeljebb N kombinációt enged meg, akkor legfeljebb $N^{\rho(H)}$ megoldás létezik és ezek felsorolhatók $\|I\|^{\rho(H)+O(1)}$ időben.

A negyedik fejezet egyik legfontosabb észrevétele, hogy a frakcionális élfedési számmal is hasonló korlátokat és eredményeket kaphatunk. A frakcionális élfedési szám az élfedési szám lineáris relaxációja, a következőképpen definiálható. Egy H hipergráf frakcionális élfedése egy olyan $x : E(H) \to [0, \infty)$ függvény melyre $\sum_{e \ni v} x(e) \geq 1$ teljesül minden $v \in V(H)$ csúcs esetén. A frakcionális élfedési szám $\rho^*(H)$

a $\sum_{e\in E(H)} x(e)$ összeg minimuma, ahol x a H hipergráf frakcionális élfedése. Könnyen látható, hogy $\rho^*(H) \leq \rho(H)$: egy k méretű élhalmaz, amely lefedi az összes csúcsot, tekinthető úgy is, mint egy frakcionális élfedés, amely pontosan k élen vesz fel 1 értéket és az összes többi élen 0 értékű. Viszont ismeretes, hogy $\rho(H)$ tetszőlegesen nagyobb lehet, mint $\rho^*(H)$.

A 4. fejezet megmutatja, hogy ha CSP feladat hipergráfjának a frakcionális élfedési száma alacsony, akkor ez (az élfedési számhoz hasonló módon) korlátot ad a lehetséges megoldások számára és lehetővé teszi a megoldások hatékony megtalálását.

8. tétel (Theorem 4.6 az értekezésben). Ha egy CSP feladat hipergráfja H és minden feltétel reláció legfeljebb N kombinációt enged meg, akkor legfeljebb $N^{\rho^*(H)}$ megoldás létezik és ezek felsorolhatók $\|I\|^{\rho^*(H)+O(1)}$ időben.

Ebből következően ha csak olyan H hipergráfokat tekintünk, ahol $\rho^*(H) \leq c$ valamilyen c konstansra, akkor a probléma polinomidőben megoldható.

9. következmény (Corollary 4.7 az értekezésben). Ha $\mathcal{H}$ egy korlátos frakcionális élfedési számú hipergráf osztály, akkora CSP($\mathcal{H}$) probléma polinom időben megoldható.

A következő tétel megmutatja, hogy az $N^{\rho^*(H)}$ felső korlát a 8. tételben éles, vagyis a megoldások lehetséges száma tényleg szorosan összefügg a frakcionális élfedési számmal.

10. tétel (Theorem 4.8 az értekezésben). Legyen H egy tetszőleges hipergráf. Minden $N_0 \geq 1$ értékre létezik egy olyan CSP feladat, amelynek H a hipergráfja, minden feltétel reláció legfeljebb $N \geq N_0$ kombinációt enged meg és a megoldások száma legalább $N^{\rho^*(H)}$.

Vegyük észre, hogy a 8. tételben az összes megoldás felsorolásához szükséges futási idő kitevője tartalmaz egy $+O(1)$ tagot a megoldások lehetséges számához képest. Munkánk során nem volt cél ennek a futási időnek további optimalizálása, de ezt felismerve később más szerzők hatékonyabb algoritmusokat adtak, ahol kitevőben csak $\rho^*(H)$ áll [66, 71].

A fenti eredmények elsősorban az adatbáziselméleti területen találtak visszhangra, ahol kiindulási alapul szolgáltak számos további cikkhez

[38, 46, 52, 54, 55, 66, 67, 71]. Ismert, hogy relációs adatbázisokban a lekérdezések megfogalmazhatók egy CSP feladatként. Például ha adott három reláció $R(a, b)$, $S(b, c)$ és $T(a, c)$ az a, b, c attribútumokon, akkor a

$$Q(a, b, c) := R(a, b) \bowtie S(b, c) \bowtie T(c, a).$$

lekérdezést lehet úgy tekinteni, hogy a $Q(a, b, c)$ eredmény reláció egy CSP feladat összes megoldását tartalmazza, ahol ez a CSP feladat a három változón (a, b, c) értelmezett és három bináris korlátot (R, S, T) tartalmaz. Vagyis a 8.–10. tételek alsó és felső korlátokat adnak a lekérdezések megoldásszámára és a megoldások felsorolásához szükséges időre. Az általunk adott eredmények megmutatják a frakcionális élfedési szám szerepét és éles korlátot adnak, választ adva egy alapvető kérdésre az adatbázisok elméletében. Noha ezek az eredmények technikailag nem túl bonyolultak (felhasználva Shearer [18] egy kombinatorikus tételét és a lineáris programozás dualitását), meglepő módon nem szerepelt ilyen eredmény az irodalomban. Munkánkra építve több új cikk született at adatbáziselmélet legfontosabb konferenciáin és folyóirataiban [38, 46, 52, 54, 55, 66, 67, 71], mutatva ezen eredmények alapvető mivoltát.

A 4. fejezet második fele átfogalmazza a CSP eredményeket az adatbáziselmélet nyelvén és tovább finomítja őket. Pontosabb korlátot adunk abban az estben, ha nem egy uniform N felső korlát adott minden egyes relációnál a megengedett kombinációk számára, hanem minden R_i relációra külön-külön adott egy N_i felső korlát. Vizsgáljuk, hogy ha bemenetként adottak ezek az N_i korlátok, akkor mennyire pontos korlátot lehet ebből kiszámolni polimomidőben. Tovább megmutatjuk, hogy a 8. tételben adott algoritmus kifejezhető ún. Join-Project Plan formában a megadott futási idő mellett, de vannak olyan esetek amikor az egyszerűbb Join Plan formában csak a futási idő szignifikáns növekedése mellett lehet a lekérdezést kifejezni. Ezen eredmények pontos kifejtése és a relációs adatbázisok elméleti hátterének rövid összefoglalása az értekezés 1.5. illetve 4.3.-4.5. fejezeteiben található.

A 4.1.-4.2. fejezetben közölt eredmények az *ACM Transactions on Algorithms* folyóíratban jelentek meg egy Martin Grohe-val közös cikkben [42]. A cikk előzetes változata a SODA 2006 konferencia kiadványában jelent meg [41]. A 4.3-4.5 fejezetben közült eredmények az *SIAM Journal on Computing* folyóíratban jelentek meg egy Albert Atserias-szal és Martin Grohe-val közös cikkben [5]. A cikk előzetes változata a FOCS 2008 konferencia kiadványában jelent meg [4].

5. Frakcionális hiperfavastagság

Ismert, hogy ha a $\mathcal{H}$ hipergráfosztály favastagsága korlátos, akkor CSP($\mathcal{H}$) polinomidőben megoldható (2. tétel). Ezt általánosítva Gottlob et al. [37,39] bevezette az (általánosított) hiperfa vastagság fogalmát és megmutatta, hogy ha a $\mathcal{H}$ hipergráfosztály hiperfa vastagsága korlátos, akkor CSP($\mathcal{H}$) polinomidőben megoldható. Másrészről a 9. következményben kimondtuk, hogy CSP($\mathcal{H}$) akkor is megoldható polinom időben, ha $\mathcal{H}$ korlátos frakcionális élfedési számú. Ez a két eredmény egymással összemérhetetlen: a korlátos hiperfa vastagság és a korlátos frakcionális szám tulajdonságok közül egyik sem implikálja a másikat (lásd az 1. ábrát).

Az 5. fejezet ennek a két eredménynek a közös általánosítását adja a frakcionális hiperfavastagság bevezetésével. Egy H hipergráf $(\mathcal{B}, T)$ fafelbontásának a frakcionális hiperfa vastagsága az a legkisebb c érték, amire teljesül hogy minden $t \in V(T)$ esetén a B_t csúcshalmaz lefedhető egy legfeljebb c értékű frakcionális fedéssel. A H hipergráf frakcionális hiperfa vastagsága a lehető legjobb (legkisebb értékű) fafelbontásának a frakcionális hiperfa vastagsága. Ez a definíció általánosítja Gottlob et al. [37, 39] (általánosított) hiperfa vastagság fogalmát: a lényeges különbség, hogy itt egy B_t-t lefedő c értékű frakcionális fedést követelünk meg, nem egy B_t-t lefedő c hiperélből álló élhalmazt. Továbbá általánosítja frakcionális élfedési szám fogalmát is: ha $\mathcal{H}$ egy korlátos frakcionális élfedési számú hipergráf osztály, akkor nyilvánvalóan $\mathcal{H}$ korlátos frakcionális hiperfa vastagságú.

A 8. tételt standard dinamikus programozási technikákkal kombinálva kapjuk a következő eredményt:

11. tétel (Theorem 5.6 az értekezésben). Legyen adott egy CSP feladat H hipergráffal és H egy $(\mathcal{B}, T)$ fafelbontását, melynek frakcionális hiperfa vastagsága legfeljebb r. Ekkor a feladat $\|I\|^{r+O(1)}$ időben megoldható.

Vagyis a 11. tétel megmutatja, hogy az alacsony hiperfa vastagság hatékony megoldhatóságot tesz lehetővé, azzal a technikai megkötéssel, hogy nem elég, ha egy ilyen fafelbontás létezik, hanem feltételezzük, hogy ténylegesen adott is egy ilyen felbontás a bemenetben. Sajnos már annak az eldöntése is NP-teljes, hogy a frakcionális hiperfa vastagság legfeljebb 2 [34], vagyis nem remélhetjük, hogy hatékony algoritmust találhatunk egy adott hipergráf optimális frakcionális hiperfa vastagságú felbontására. Viszont adunk egy közelítő algoritmust, ami egy olyan felbontást ad, aminek a frakcionális hiperfa vastagsága az optimumnak lefeljebb a köbe.

12. tétel (Theorem 5.4 az értekezésben). Minden $r \geq 1$ értékre létezik egy $n^{O(r^3)}$ idejű algoritmus, a következő specifikációval: ha az input egy H hipergráf melynek a frakcionális hiperfa vastagsága legfeljebb r, akkor a kiment H egy fafelbontása melynek a frakcionális hiperfa vastagsága $O(r^3)$.

Vagyis ha olyan CSP feladatokat tekintünk, ahol tudjuk, hogy a H hipergráf frakcionális hiperfa vastagsága legfeljebb r, akkor a 12. tétel egy $O(r^3)$ frakcionális hiperfa vastagságú felbontást ad és a 11. tétel algoritmusával a probléma megoldható $\|I\|^{O(r^3)}$ időben. Vagyis minden fix r értékre létezik polinomidejű algoritmus, így a 9. következményt általánosíthatjuk:

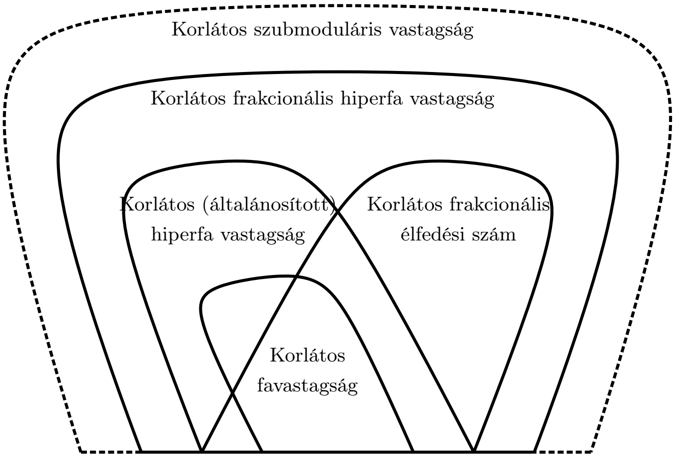
13. következmény (Corollary 5.8 az értekezésben). Ha $\mathcal{H}$ egy korlátos frakcionális hiperfa vastagságú hipergráf osztály, akkor a CSP($\mathcal{H}$) probléma polinom időben megoldható.

Az 1. ábrán láthatjuk azokat az ismert hipergráf tulajdonságokat, amelyek a CSP feladat hatékony megoldhatóságát garantálják. Mint az ábra mutatja, a korlátos frakcionális hiperfa vastagság a legbővebb ilyen tulajdonság.

Az 5.1. fejezetben közölt eredmények az *ACM Transactions on Algorithms* folyóíratban jelentek meg egy Martin Grohe-val közös cikkben [42]. A cikk előzetes változata a SODA 2006 konferencia kiadványában jelent meg [41]. Az 5.2. fejezetben közölt eredmények az *ACM Transactions on Algorithms* folyóíratban jelentek meg egy egyszerzős cikkben [61]. A cikk előzetes változata a SODA 2009 konferencia kiadványában jelent meg [60].

6. CSP hatékony megoldhatóságát lehetővé tevő hipergráf tulajdonságok teljes klasszifikációja.

A 3. tétel teljesen klasszifikálja (megfelelő bonyolultságelméleti hipotézis esetén) azokat a $\mathcal{G}$ gráfosztályokat, amelyek esetén CSP($\mathcal{G}$) polinom időben megoldható. Természetesen merül fel a kérdés, hogy lehet-e adni



1. ábra. Hipergráf tulajdonságok, amelyek a CSP feladat hatékony megoldhatóságát garantálják. A korlátos szubmoduláris vastagságú hipergráf osztályokra a probléma FPT, míg a többi tulajdonság esetén a probléma polinom időben megoldható.

hasonló klasszifikációt azon $\mathcal{H}$ hipergráf osztályokra, amelyek esetén CSP($\mathcal{H}$) polinom időben megoldható. A 13. következmény kimondja, hogy a korlátos frakcionális hiperfa vastagságú osztályok ilyenek, vajon léteznek-e ezen túl is ilyen osztályok? Vagy megmutatható esetleg, hogy ha a $\mathcal{H}$ osztálynak nem korlátos a frakcionális hiperfa vastagsága, akkor CSP($\mathcal{H}$) nem oldható meg polinom időben?

Ezekre a kérdésekre jelenleg nem ismert pontos válasz. A 6. fejezet viszont választ ad a kérdéseknek egy szintén természetes, de technikailag robusztusabb változatára. Relaxáljuk azt a követelményt, hogy a feladatot polinom időben kell megoldani és engedjünk meg a futási időben egy $f(H)$ tényezőt, ami csak a feladat hipergráfjától függ: a keresett futási idő $f(H)\|I\|^{O(1)}$, ahol $f(H)$ egy tetszőleges függvény ami csak a H hipergráftól függ. (Ez ekvivalens azzal, hogy $f(k)\|I\|^{O(1)}$ időt követelünk meg valamilyen $f(k)$ függvényre ami csak a változók k számától függ.) Vagyis a paraméteres bonyolultság nyelvén fogalmazva azokat az eseteket szeretnénk karakterizálni, ahol a probléma fixed-parameter tractable (FPT) a hipergráffal (illetve annak méretével) parametrizálva. Az ilyen formájú futási idő megengedése akkor motivált, ha feltételezzük, hogy a változók számánál a bemenet lényegesen nagyobb, ezért megengedhető, ha a futási időben szerepel pl. egy a változók számában exponenciális tényező. Ez a feltételezés helytálló lehet adatbáziselméleti alkalmazásokban, ahol tipikusan a lekérdezés mérete (változók száma) elenyésző az adatbázisban szereplő relációk méretéhez képest. Vagyis esetleg megengedhető egy akár exponenciális idejű előfeldolgozás is a lekérdezés optimalizálására (például megfelelő felbontások keresésére), ha ez az exponenciális idő csak a lekérdezés hosszától függ és nem függ az adatbázis méretétől. Ez a szemlélet nagyon gyakori az adatbáziselméleti irodalomban (a „data complexity" és a „query complexiy" szétválasztása) és egy sokkal robusztusabb képet ad, ahol például az optimális felbontások megkeresésének a bonyolultsága irreleváns és jobban lehet koncentrálni magának a lekérdezésnek az optimális módon történő végrehajtásának a bonyolultságára.

A 6. fejezet fő eredménye pontosan karakterizálja azokat a hipergráf osztályokat amelyekre a CSP($\mathcal{H}$) probléma FPT a hipergráffal parametrizálva. Az eredmény kimondásához egy új definíciót, a szubmoduláris vastagságot kell bevezetni. Egy U univerzum részhalmazain értelmezett f függvény *szubmoduláris*, ha $f(X) + f(Y) \geq f(X \cap Y) \cup f(X \cup Y)$ minden $X, Y \subseteq U$ részhalmazra és *monoton* ha $f(X) \leq f(Y)$ teljesül minden $X \subseteq Y \subseteq U$ eset;n. Legyen $(\mathcal{B}, T)$ egy H hipergráf fafelbontása és legyen f egy U részhalmazain értelmezett függvény. A felbontás *f-vastagsága* az $f(B_t)$ érték maximuma minden $t \in V(T)$ esetén. A H hipergráf szubmoduláris vastagsága az f-vastagság legnagyobb lehetséges értéke minden olyan monoton szubmoduláris f függvényre, amely teljesíti, hogy $f(\emptyset) = 0$ és $f(e) \leq 1$ minden $e \in E(H)$ hiperélre. Másképpen fogalmazva, ha H szubmoduláris vastagsága c, akkor minden fenti feltételeket teljesítő f függvényre létezik olyan fafelbontás, amelynek az f-vastagsága legfeljebb c. A definíció érdekessége, hogy a különböző f függvényekhez különböző fafelbontások tartozhatnak: nem feltétlenül létezik olyan fafelbontás aminek alacsony az f-vastagsága minden megengedett f függvényre. Megmutatható, hogy egy H hipergráf szubmoduláris vastagsága nem nagyobb mint a frakcionális hiperfa vastagság, vagyis a korlátos szubmoduláris vastagság egy általánosabb tulajdonság (lásd az 1. ábrát).

A 6. fejezet fő eredménye kimondja, hogy ennek a paraméternek a korlátossága esetén lesz a CSP($\mathcal{H}$) feladat FPT.

14. tétel (Theorem 6.1 az értekezésben). Ha ETH igaz, akkor a CSP($\mathcal{H}$) problémára akkor és csak akkor létezik $f(H)\|I\|^{O(1)}$ idejű algoritmus valamilyen f függvényre ha $\mathcal{H}$ egy korlátos szubmoduláris vastagságú hipergráf osztály.

Ennek az eredménynek a bizonyítása két részből áll: egy algoritmikus részből (algoritmus a CSP($\mathcal{H}$) problémára, ha $\mathcal{H}$ egy korlátos szubmoduláris vastagságú hipergráf osztály) és egy bonyolultságelméleti részből (nem létezik ilyen algoritmus, ha $\mathcal{H}$ nem egy ilyen osztály). Mindkét rész lényeges új technikai ötleteket követel. Ezen algoritmikus első lépését Alon et al. [1] egy kombinatorikus eredménye inspirálja. Az algoritmus először elágazásokkal több alesetre bontja a feladatot úgy, hogy minden alesetben a megoldás tér bizonyos értelemben uniform: minden részmegoldás nagyjából azonos számú nagyobb megoldásra terjeszthető ki. Ez a tulajdonság lehetővé teszi, hogy a részmegoldások számát egy szubmoduláris függvénnyel közelítsük. A korlátos szubmoduláris vastagság biztosítja, hogy létezik egy olyan fafelbontás, ahol minden B_t csúcshalmazon ennek a szubmoduláris függvénynek az értéke alacsony, vagyis minden ilyen csúcshalmazon korlátos számú részmegoldás létezik. Ebben az esetben viszont a probléma standard módszerekkel megoldható. A bonyolultságelméleti részben a fő kihívás, hogy megértsük mit is jelent a magas szubmoduláris vastagság. Kombinatorikai módszerekkel megmutatjuk, hogy magas szubmoduláris vastagság esetén a hipergráf tartalmaz olyan struktúrákat, amelyek lehetőve teszik egy 3SAT probléma hatékony beágyazását. Ebből következik, hogy ha a CSP($\mathcal{H}$) problémára létezne $f(H)\|I\|^{O(1)}$ idejű algoritmus, akkor ez a 3SAT problémára adna egy olyan hatékony algoritmust, ami ellentmondana az ETH feltételezésnek.

A 6. fejezetben közölt eredmények a *Journal of the ACM* folyóíratban jelentek meg egy egyszerzős cikkben [64]. A cikk előzetes változata a STOC 2010 konferencia kötetében jelent meg [63].

Hivatkozások

[1] N. Alon, I. Newman, A. Shen, G. Tardos, and N. Vereshchagin. Partitioning multi-dimensional sets in a small number of „uniform" parts. *European J. Combin.*, 28(1):134–144, 2007.

[2] E. Amir. Approximation algorithms for treewidth. *Algorithmica*, 56(4):448–479, 2010.

[3] S. Arnborg, D. G. Corneil, and A. Proskurowski. Complexity of finding embeddings in a k-tree. *SIAM J. Algebraic Discrete Methods*, 8(2):277–284, 1987.

[4] A. Atserias, M. Grohe, and D. Marx. Size bounds and query plans for relational joins. In *49th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2008)*, pages 739–748, 2008.

[5] A. Atserias, M. Grohe, and D. Marx. Size bounds and query plans for relational joins. *SIAM J. Comput.*, 42(4):1737–1767, 2013.

[6] U. Bertelè and F. Brioschi. On non-serial dynamic programming. *J. Comb. Theory, Ser. A*, 14(2):137–148, 1973.

[7] A. Björklund, T. Husfeldt, P. Kaski, and M. Koivisto. Fourier meets Möbius: fast subset convolution. In *Proceedings of the 39th Annual ACM on Symposium on Theory of Computing (STOC 2007)*, pages 67–74, 2007.

[8] H. L. Bodlaender. A linear-time algorithm for finding tree-decompositions of small treewidth. *SIAM J. Comput.*, 25(6):1305–1317, 1996.

[9] H. L. Bodlaender, M. Cygan, S. Kratsch, and J. Nederlof. Deterministic single exponential time algorithms for connectivity problems parameterized by treewidth. *Inf. Comput.*, 243:86–111, 2015.

[10] H. L. Bodlaender, P. G. Drange, M. S. Dregi, F. V. Fomin, D. Lokshtanov, and M. Pilipczuk. A $c^k \cdot n$ 5-approximation algorithm for treewidth. *SIAM J. Comput.*, 45(2):317–378, 2016.

[11] É. Bonnet, P. Giannopoulos, and M. Lampis. On the parameterized complexity of red-blue points separation. In D. Lokshtanov and N. Nishimura, editors, *12th International Symposium on Parameterized and Exact Computation (IPEC 2017)*, volume 89 of *LIPIcs*, pages 8:1–8:13. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2017.

[12] É. Bonnet and T. Miltzow. Parameterized hardness of art gallery problems. In P. Sankowski and C. D. Zaroliagis, editors, *24th Annual European Symposium on Algorithms (ESA 2016)*, volume 57 of *LIPIcs*, pages 19:1–19:17. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2016.

[13] É. Bonnet and F. Sikora. The graph motif problem parameterized by the structure of the input graph. *Discrete Applied Mathematics*, 231:78–94, 2017.

[14] G. Borradaile and H. Le. Optimal dynamic program for r-domination problems over tree decompositions. In J. Guo and D. Hermelin, editors, *11th International Symposium on Parameterized and Exact Computation (IPEC 2016)*, volume 63 of *LIPIcs*, pages 8:1–8:23. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2016.

[15] K. Bringmann, L. Kozma, S. Moran, and N. S. Narayanaswamy. Hitting set for hypergraphs of low vc-dimension. In P. Sankowski and C. D. Zaroliagis, editors, *24th Annual European Symposium on Algorithms (ESA 2016)*, volume 57 of *LIPIcs*, pages 23:1–23:18. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2016.

[16] J. Chen, B. Chor, M. Fellows, X. Huang, D. W. Juedes, I. A. Kanj, and G. Xia. Tight lower bounds for certain parameterized NP-hard problems. In *Proceedings of 19th Annual IEEE Conference on Computational Complexity (CCC 2004)*, pages 150–160, 2004.

[17] J. Chen, X. Huang, I. A. Kanj, and G. Xia. Linear FPT reductions and computational lower bounds. In *Proceedings of the 36th Annual ACM Symposium on Theory of Computing (STOC 2004)*, pages 212–221, New York, 2004. ACM.

[18] F. R. K. Chung, R. L. Graham, P. Frankl, and J. B. Shearer. Some intersection theorems for ordered sets and graphs. *J. Combin. Theory Ser. A*, 43(1):23–37, 1986.

[19] B. Courcelle. The monadic second-order logic of graphs III: Tree-decompositions, minor and complexity issues. *ITA*, 26:257–286, 1992.

[20] R. Curticapean, H. Dell, and D. Marx. Homomorphisms are a good basis for counting small subgraphs. In H. Hatami, P. McKenzie, and V. King, editors, *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing (STOC 2017), Montreal, QC, Canada, June 19-23, 2017*, pages 210–223. ACM, 2017.

[21] R. Curticapean and D. Marx. Complexity of counting subgraphs: Only the boundedness of the vertex-cover number counts. In *55th*

IEEE Annual Symposium on Foundations of Computer Science (FOCS 2014), pages 130–139. IEEE Computer Society, 2014.

[22] R. Curticapean and D. Marx. Tight conditional lower bounds for counting perfect matchings on graphs of bounded treewidth, cliquewidth, and genus. In *Proceedings of the 27th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2016)*, pages 1650–1669, 2016.

[23] R. Curticapean and M. Xia. Parameterizing the permanent: Genus, apices, minors, evaluation mod 2k. In V. Guruswami, editor, *IEEE 56th Annual Symposium on Foundations of Computer Science (FOCS 2015), Berkeley, CA, USA, 17-20 October, 2015*, pages 994–1009. IEEE Computer Society, 2015.

[24] M. Cygan, H. Dell, D. Lokshtanov, D. Marx, J. Nederlof, Y. Okamoto, R. Paturi, S. Saurabh, and M. Wahlström. On problems as hard as CNF-SAT. *ACM Trans. Algorithms*, 12(3):41:1–41:24, 2016.

[25] M. Cygan, F. V. Fomin, L. Kowalik, D. Lokshtanov, D. Marx, M. Pilipczuk, M. Pilipczuk, and S. Saurabh. *Parameterized Algorithms*. Springer, 2015.

[26] M. Cygan, S. Kratsch, and J. Nederlof. Fast hamiltonicity checking via bases of perfect matchings. *J. ACM*, 65(3):12:1–12:46, 2018.

[27] M. Cygan, J. Nederlof, M. Pilipczuk, M. Pilipczuk, J. M. M. van Rooij, and J. O. Wojtaszczyk. Solving connectivity problems parameterized by treewidth in single exponential time. In *Proceedings of the 52nd Annual IEEE Symposium on Foundations of Computer Science (FOCS 2011)*, pages 150–159, 2011.

[28] V. Dalmau and P. Jonsson. The complexity of counting homomorphisms seen from the other side. *Theor. Comput. Sci.*, 329(1-3):315–323, 2004.

[29] R. G. Downey and M. R. Fellows. *Parameterized Complexity*. Monographs in Computer Science. Springer, New York, 1999.

[30] R. G. Downey and M. R. Fellows. *Fundamentals of Parameterized Complexity*. Texts in Computer Science. Springer, 2013.

[31] L. Egri, D. Marx, and P. Rzążewski. Finding list homomorphisms from bounded-treewidth graphs to reflexive graphs: a complete complexity characterization. In *35th Symposium on Theoretical Aspects of Computer Science (STACS 2018)*, pages 27:1–27:15, 2018.

[32] E. Eiben, D. Knop, F. Panolan, and O. Suchý. Complexity of the Steiner network problem with respect to the number of terminals. *CoRR*, abs/1802.08189, 2018.

[33] D. Eppstein and D. Lokshtanov. The parameterized complexity of finding point sets with hereditary properties. *CoRR*, abs/1808.02162, 2018.

[34] W. Fischl, G. Gottlob, and R. Pichler. General and fractional hypertree decompositions: Hard and easy cases. In *Proceedings of the 37th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems (PODS 2018)*, pages 17–32, 2018.

[35] J. Flum and M. Grohe. *Parameterized Complexity Theory*. Texts in Theoretical Computer Science. An EATCS Series. Springer, Berlin, 2006.

[36] E. C. Freuder. Complexity of k-tree structured constraint satisfaction problems. In *Proc. of AAAI-90*, pages 4–9, Boston, MA, 1990.

[37] G. Gottlob, M. Grohe, N. Musliu, M. Samer, and F. Scarcello. Hypertree decompositions: Structure, algorithms, and applications. In D. Kratsch, editor, *Graph-Theoretic Concepts in Computer Science (WG 2005)*, volume 3787 of *Lecture Notes in Computer Science*, pages 1–15. Springer Berlin / Heidelberg, 2005.

[38] G. Gottlob, S. T. Lee, G. Valiant, and P. Valiant. Size and treewidth bounds for conjunctive queries. *J. ACM*, 59(3):16:1–16:35, June 2012.

[39] G. Gottlob, N. Leone, and F. Scarcello. Hypertree decompositions and tractable queries. *Journal of Computer and System Sciences*, 64:579–627, 2002.

[40] M. Grohe. The complexity of homomorphism and constraint satisfaction problems seen from the other side. *J. ACM*, 54(1):1, 2007.

[41] M. Grohe and D. Marx. Constraint solving via fractional edge covers. In *Proceedings of the of the 17th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2006)*, pages 289–298, 2006.

[42] M. Grohe and D. Marx. Constraint solving via fractional edge covers. *ACM Trans. Algorithms*, 11(1):4:1–4:20, 2014.

[43] M. Grohe, T. Schwentick, and L. Segoufin. When is the evaluation of conjunctive queries tractable? In *Proceedings of the thirty-third annual ACM symposium on Theory of computing (STOC 2001)*, pages 657–666, New York, NY, USA, 2001. ACM Press.

[44] J. Guo, S. Hartung, R. Niedermeier, and O. Suchý. The parameterized complexity of local search for TSP, more refined. *Algorithmica*, 67(1):89–110, 2013.

[45] R. Halin. S-functions for graphs. *Journal of Geometry*, 8(1-2):171–186, 1976.

[46] X. Hu and K. Yi. Towards a worst-case I/O-optimal algorithm for acyclic joins. In *Proceedings of the 35th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems (PODS 2016)*, pages 135–150, 2016.

[47] R. Impagliazzo and R. Paturi. On the complexity of k-SAT. *J. Comput. Syst. Sci.*, 62(2):367–375, 2001.

[48] R. Impagliazzo, R. Paturi, and F. Zane. Which problems have strongly exponential complexity? *J. Comput. System Sci.*, 63(4):512–530, 2001.

[49] Y. Iwata and Y. Yoshida. On the equivalence among problems of bounded width. In N. Bansal and I. Finocchi, editors, *23rd Annual European Symposium (ESA 2015)*, volume 9294 of *Lecture Notes in Computer Science*, pages 754–765. Springer, 2015.

[50] L. Jaffke and B. M. P. Jansen. Fine-grained parameterized complexity analysis of graph coloring problems. In *Proceedings of the 10th International Conference on Algorithms and Complexity (CIAC 2017)*, volume 10236 of *Lecture Notes in Computer Science*, pages 345–356, 2017.

[51] K. Jansen, S. Kratsch, D. Marx, and I. Schlotter. Bin packing with fixed number of bins revisited. *J. Comput. Syst. Sci.*, 79(1):39–49, 2013.

[52] M. Joglekar and C. Ré. It's all a matter of degree - using degree information to optimize multiway joins. *Theory Comput. Syst.*, 62(4):810–853, 2018.

[53] M. Jones, D. Lokshtanov, M. S. Ramanujan, S. Saurabh, and O. Suchý. Parameterized complexity of directed steiner tree on sparse graphs. *SIAM J. Discrete Math.*, 31(2):1294–1327, 2017.

[54] M. A. Khamis, H. Q. Ngo, C. Ré, and A. Rudra. Joins via geometric resolutions: Worst case and beyond. *ACM Trans. Database Syst.*, 41(4):22:1–22:45, 2016.

[55] M. A. Khamis, H. Q. Ngo, and A. Rudra. FAQ: questions asked frequently. In T. Milo and W. Tan, editors, *Proceedings of the 35th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems (PODS 2016)*, pages 13–28. ACM, 2016.

[56] D. Lokshtanov, D. Marx, and S. Saurabh. Known algorithms on graphs on bounded treewidth are probably optimal. In *Proceedings of the 22nd Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2011)*, pages 777–789, 2011.

[57] D. Lokshtanov, D. Marx, and S. Saurabh. Known algorithms on graphs of bounded treewidth are probably optimal. *ACM Trans. Algorithms*, 14(2):13:1–13:30, 2018.

[58] D. Lokshtanov, M. S. Ramanujan, S. Saurabh, and M. Zehavi. Parameterized complexity and approximability of directed odd cycle transversal. *CoRR*, abs/1704.04249, 2017.

[59] D. Marx. Can you beat treewidth? In *48th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2007), October 20-23, 2007, Providence, RI, USA, Proceedings*, pages 169–179. IEEE Computer Society, 2007.

[60] D. Marx. Approximating fractional hypertree width. In C. Mathieu, editor, *Proceedings of the Twentieth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2009)*, pages 902–911. SIAM, 2009.

[61] D. Marx. Approximating fractional hypertree width. *ACM Trans. Algorithms*, 6(2):1–17, 2010.

[62] D. Marx. Can you beat treewidth? *Theory of Computing*, 6(1):85–112, 2010.

[63] D. Marx. Tractable hypergraph properties for constraint satisfaction and conjunctive queries. In *Proceedings of the 42nd ACM Symposium on Theory of Computing (STOC 2010)*, pages 735–744, 2010.

[64] D. Marx. Tractable hypergraph properties for constraint satisfaction and conjunctive queries. *J. ACM*, 60(6):42:1–42:51, 2013.

[65] D. Marx and M. Pilipczuk. Optimal parameterized algorithms for planar facility location problems using voronoi diagrams. In N. Bansal and I. Finocchi, editors, *23rd Annual European Symposium (ESA 2015)*, volume 9294 of *Lecture Notes in Computer Science*, pages 865–877. Springer, 2015.

[66] H. Q. Ngo, E. Porat, C. Ré, and A. Rudra. Worst-case optimal join algorithms. *J. ACM*, 65(3):16:1–16:40, 2018.

[67] H. Q. Ngo, C. Ré, and A. Rudra. Skew strikes back: new developments in the theory of join algorithms. *SIGMOD Record*, 42(4):5–16, 2013.

[68] M. Pilipczuk and M. Wahlström. Directed multicut is W[1]-hard, even for four terminal pairs. *TOCT*, 10(3):13:1–13:18, 2018.

[69] N. Robertson and P. D. Seymour. Graph minors. III. Planar tree-width. *J. Comb. Theory, Ser. B*, 36(1):49–64, 1984.

[70] J. M. M. van Rooij, H. L. Bodlaender, and P. Rossmanith. Dynamic programming on tree decompositions using generalised fast subset convolution. In *Proceedings of the 17th Annual European Symposium on Algorithms (ESA 2009)*, volume 5757 of *Lecture Notes in Computer Science*, pages 566–577. Springer, 2009.

[71] T. L. Veldhuizen. Triejoin: A simple, worst-case optimal join algorithm. In *Proc. 17th International Conference on Database Theory ((ICDT) 2014)*, pages 96–106, 2014.