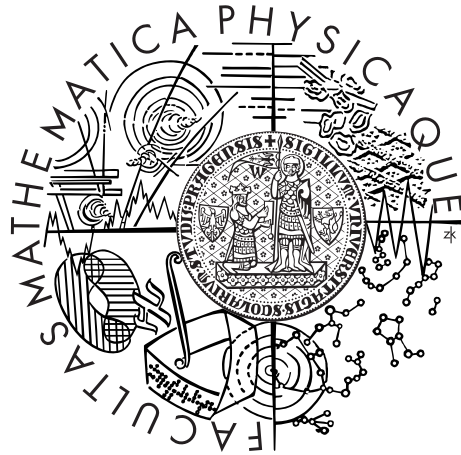


Univerzita Karlova v Praze
Matematicko-fyzikální fakulta

DIPLOMOVÁ PRÁCE



Miloš Chromý

Rozšíření matched formulí

Katedra teoretické informatiky a matematické logiky

Vedoucí diplomové práce: RNDr. Petr Kučera Ph.D.

Studijní program: Informatika

Studijní obor: Teoretická informatika

Praha 2015

Rád bych poděkoval svým rodičům za podporu v celém průběhu studií i v životě, vedoucímu diplomé práce RNDr. Petru Kučerovi Ph.D. za odborné vedení, připomínky, odborné rady, čas a trpělivost, kterou měl při vedení této práce, doc. RNDr. Ondřeji Čepkovi, Ph.D. za odborné rady, čas a připomínky, které mi poskytl v průběhu zpracovávání diplomové práce a také kamarádům, kteří mě podporují a pomáhají mi v mnoha směrech.

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně a výhradně s použitím citovaných pramenů, literatury a dalších odborných zdrojů.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona v platném znění, zejména skutečnost, že Univerzita Karlova v Praze má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle §60 odst. 1 autorského zákona.

V dne

Podpis autora

Název práce: Rozšíření matched formulí

Autor: Miloš Chromý

Katedra: Katedra teoretické informatiky a matematické logiky

Vedoucí diplomové práce: RNDr. Petr Kučera Ph.D., Katedra teoretické informatiky a matematické logiky

Abstrakt: S KNF formulí můžeme asociovat incidenční graf. Tento graf je bipartitní a jedna partita zastupuje proměnné a druhá klauzule. Díky tomu můžeme definovat nové třídy KNF formulí, jimiž jsou matched formule a biklikově splnitelné formule. Obě tyto třídy mají tu vlastnost, že formule, které do nich patří jsou splnitelné i po změně polarity libovolného literálu. Takovým formulím říkáme var-splnitelné.

V této práci uvažujeme práci Stefana Szeidera, která popisuje parametrizovaný algoritmus s pevným parametrem řešící problém matched formulí s malou deficiencí, což je rozdíl počtu klauzulí a počtu proměnných. Ukázali jsme, proč tento přístup nejde přímo zobecnit pro biklikově splnitelné formule.

Vzhledem k tomu, že testování toho, je-li formule biklikově splnitelná, je NP-úplné, popsali jsme heuristiku, která hledá biklikové pokrytí v čase $\mathcal{O}(n^2e)$, kde n je počet proměnných ve formuli a e je délka formule. Provedli jsme experimenty na náhodných formulích. Z výsledků těchto experimentů lze usuzovat, že existuje fázový přechod ve výsledcích heuristiky. Dále jsme provedli experimenty, které ověřují existenci fázového přechodu matched formulí. Tyto výsledky jsme porovnali s výsledky experimentů provedených s heuristikou. Výsledky experimentu provedeném na matched formulí jsme též porovnali s teoretickou hranicí fázového přechodu, kterou dokázali John Franco a Allen Van Gelder.

Klíčová slova: Matched formule, splnitelnost, maximální deficiencie formule.

Title: Extensions to the class of matched formulae

Author: Miloš Chromý

Department: Department of Theoretical Computer Science and Mathematical Logic

Supervisor: RNDr. Petr Kučera Ph.D., Department of Theoretical Computer Science and Mathematical Logic

Abstract: We can associate an incidence graph with any CNF formula. It's a bipartite graph, in which the first part corresponds to variables and the second one to clauses. We can define matched formulas and biclique satisfiable formulas, based on this incidence graph. Both of these classes share an interesting property: Given a formula F which is matched or biclique satisfiable, F remains satisfiable even after we switch polarity of any occurrence of any literal. Class of formulas with this property is called var-satisfiable.

In this thesis, we consider a parameterized algorithm introduced by Stefan Szeider for deciding satisfiability of formulas with small deficiency. Here deficiency of a formula is defined as a difference between the number of clauses and the number of variables in the formula. We explain why this algorithm cannot be simply generalized for the case of biclique satisfiable formulas.

Since the problem of determining whether a formula is biclique satisfiable is NP-complete, we introduce a heuristic, which tries to find some biclique cover in time $\mathcal{O}(n^2e)$, where n denotes the number of variables and e denotes the length of the input formula. We performed experiments testing this heuristic on random formulas. The results of these experiments suggest, that there is a phase transition in the behaviour of the heuristic. We have also made experiments checking a phase transition of the property of a formula being matched. We compare the results of experiments on matched formulas to the results of experiments performed with the heuristic. We also compare the results of experiments on matched formulas with a theoretical bound which is due to John Franco and Allen Van Gelder.

Keywords: Matched formulae, satisfiability, maximum deficiency of a formula.

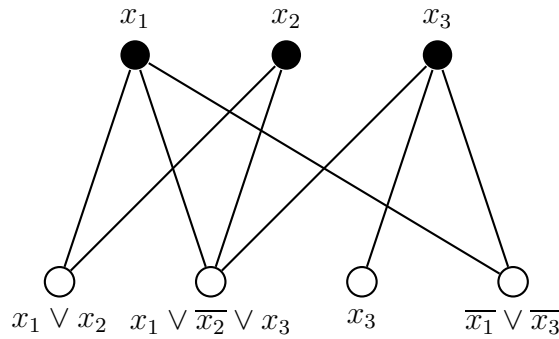
Obsah

1	Úvod	2
2	Definice a značení	5
2.1	Teorie grafů	5
2.2	Booleovské formule	6
2.3	Složitost	8
3	Třídy formulí	11
3.1	Matched formule	11
3.2	Biklikově splnitelné formule	11
3.3	Var-splnitelnost	13
4	Formule s malou deficiencí	15
4.1	Parametrizovaný algoritmus	15
4.2	Preproces redukce	16
4.2.1	Redukce $R_{G,M}$	16
4.2.2	Odstranění čistých literálů	16
4.2.3	Odstranění singulárních literálů	16
4.3	2-expandující redukce	17
5	Pokrytí bipartitního grafu omezenými biklikami	19
5.1	Algoritmy	19
5.2	Heuristiky	20
5.3	Rozšíření matched formulí s malou deficiencí na bikliky	25
6	Experimenty	29
6.1	Generátor náhodných formulí	29
6.2	Matched formule	33
6.3	Heuristika na biklikové pokrytí	39
6.4	Porovnání matched a biklikově pokrytelných formulí	49
7	Závěr	54
7.1	Zhodnocení výsledků experimentů	54
7.2	Závěr	57
	Seznam použitých zkratk	63
8	Přílohy	64
8.1	Uživatelská dokumentace	64
8.2	Hardware a software použitý pro experimenty	64
8.3	Obsah CD	65

1. Úvod

Problém splnitelnosti (SAT) je problém, ve které se ptáme, zda pro danou booleovskou formuli F v konjunktivně normální formě (KNF) existuje splňující ohodnocení. Tento problém je jedním z prvních NP-úplných problémů [8, 22], není tedy znám polynomiální algoritmus, který by ho řešil. Protože problém splnitelnosti je problémem, který nalézá uplatnění v řadě oblastí teoretické i aplikované informatiky, například v logice, umělé inteligenci či operační analýze, je věnováno značné úsilí na nalézání nových postupů řešení. Přitom jsou jednak navrhovány a zkoumány algoritmy řešící obecnou splnitelnost [11], jednak jsou zkoumány speciální třídy formulí, pro něž je možné rozhodnout splnitelnost v polynomiálním čase. Mezi takové třídy patří například hornovské formule, q-hornovské formule, formule v 2-KNF, třída SLUR a další [6, 15, 17, 21, 28, 32].

Jednou z tříd formulí uvažovaných v literatuře je třída **matched** [21, 35]. Abychom mohli tuto třídu zavést, asociujeme s každou KNF formulí F její incidenční graf $I(F)$, kde vrcholy jedné partity jsou proměnné vyskytující se ve formuli F a vrcholy druhé partity jsou klauzule formule F . Hrana vede mezi vrcholem reprezentujícím proměnnou v a vrcholem reprezentujícím klauzuli C právě tehdy, když v má výskyt v C .



Obrázek 1.1: Incidenční graf formule $F = (x_1 \vee x_2) \wedge (x_1 \vee \bar{x}_2 \vee x_3) \wedge x_3 \wedge (\bar{x}_1 \vee \bar{x}_3)$

Obrázek 1.1 znázorňuje formuli F a její incidenční graf. Můžeme si povšimnout, že v incidenčním grafu nezáleží na polaritě literálů. V několika článcích již bylo nezávisle dokázáno, že **matched** formule jsou vždy splnitelné [2, 21, 35, 36]. Maximální párování incidenčního grafu lze navíc nalézt v polynomiálním čase [14, 27] a lze tak rozpoznat, zda daná formule je **matched**. John Franco a Allen Van Gelder v [21] dokázali, že pokud poměr počtu klauzulí k počtu proměnných formule F je menší než 0.64, je 3-KNF formule téměř jistě **matched** [21]. Tento odhad se pokusíme experimentálně vyzkoušet pro k -KNF s $3 \leq k \leq 9$ v kapitole 6 sekci 6.2.

Pokud formule F není **matched**, můžeme měřit to, jak daleko je F od toho, aby **matched** byla. K tomu slouží pojem **deficiency**. Deficiency formule F s m klauzulemi a n proměnnými je definována jako $\delta(F) = m - n$. Maximální deficiency formule je potom definována jako $\delta^*(F) = \max_{F' \subseteq F} \delta(F')$. Ukazuje se [29], že ekvivalentně můžeme $\delta^*(F)$ definovat jako počet nepokrytých klauzulí v libovolném maximálním párování M incidenčního grafu $I(F)$ formule F . Formule je **matched**, pokud maximální deficiency $\delta^*(F) = 0$. Stefan Szieder [34] popsal

parametrizovaný algoritmus s pevným parametrem k pro formule s $\delta^*(F) = k$, který pracuje v čase $\mathcal{O}(2^k n^3)$, kde n je počet proměnných ve formuli F . Tento algoritmus popíšeme v kapitole 4.

U matched formulí hledáme pokrytí incidenčního grafu párovacími hranami, tedy biklikami $K_{1,1}$.

Stefan Szeider v [35] uvážil zobecnění matched formulí založené na tom, že místo pokrývání incidenčního grafu párováním, tedy hranami, pokrýváme incidenční graf většími úplnými bipartitními grafy, tedy obecnými biklikami. V pokrytí však nemůžeme uvážít libovolné bikliky, ale pouze omezené bikliky B , které splňují omezení $|V_2(B)| < 2^{|V_1(B)|}$. U matched formulí uvažujeme jen bikliky $K_{1,1}$, tedy hrany, můžeme však uvažovat i větší bikliky, například $K_{2,3}$. Při pokrývání grafu nás navíc zajímají pokrytí omezenými biklikami, které pokrývají všechny klauzule. Je-li možné pro danou formuli F najít pokrytí omezenými biklikami, řekneme o ní, že je **biklikově splnitelná**, Stefan Szeider v [35] nahlédl, že biklikově splnitelné formule jsou skutečně vždy splnitelné. Na rozdíl od matched formulí je rozpoznání toho, zda je daná formule biklikově splnitelná, NP-úplným problémem, a to i v případě, kdy v pokrytí uvažujeme jen omezené bikliky $K_{2,3}$ a $K_{1,1}$. Jak matched formule tak biklikově splnitelné formule mají tu vlastnost, že jsou splnitelné nezávisle na polaritě literálů, které se v nich vyskytují. Problémem pokrytí bipartitních grafů biklikami se zabývali D.Bein a kol. [3], nebo H. Fleischner a kol. [19]. V obou případech je hlavním parametrem pro pokrytí počet biklik, kterými chceme pokrýt graf a bikliky nemusí být omezenými biklikami. Aproximační algoritmy a heuristiky navržené v článcích [3, 19] tedy nelze jednoduše použít i pro případ omezených biklik. Pokrývání bipartitního grafu omezenými biklikami se dosud zabýval pouze Stefan Szeider v [35], který však ukázal jen negativní důkaz NP-úplnosti souvisejícího problému. Jedním z cílů této diplomové práce je prozkoumat heuristické přístupy k hledání pokrytí incidenčního grafu formule omezenými biklikami.

V kapitole 5 popíšeme různé přístupy pro pokrytí grafu biklikami a v sekci 5.2 si pak popíšeme heuristiku na nalezení pokrytí grafu omezenými biklikami, která je založena na heuristice pro pokrytí bipartitního grafu disjunktními biklikami (ne nutně omezenými) popsané v [26].

V sekci 5.3 se pokusíme zdůvodnit, proč parametrizovaný algoritmus pro splnitelnost popsaný v [34], který je parametrizovaný maximální deficiencí formule, nelze přímo zobecnit pro případ biklik a analogicky definované biklikové deficienci formule.

V kapitole 6 popíšeme experimenty, ve kterých jsme testovali, kolik procent náhodných bipartitních grafů lze pokrýt biklikovým pokrytím, pokud použijeme heuristiku popsanou v sekci 5.2. Pro bipartitní grafy, jejichž poměr počtu vrcholů druhé partity k počtu vrcholů první partity je r , označíme **K -fázový přechod** interval $\langle x_r, y_r \rangle$, kde pro všechny grafy, jejichž všechny vrcholy druhé partity mají stupeň $k < x_r$, heuristika ze sekce 5.2 téměř jistě nenalezne biklikové pokrytí a pro všechny grafy, jejichž všechny vrcholy druhé partity mají stupeň $k > y_r$, heuristika popsaná v sekci 5.2 téměř jistě nalezne biklikové pokrytí. U experimentů v sekci 6.3 budeme pozorovat, zda se vyskytne K -fázový přechod, který bude mít malou velikost $|y_r - x_r|$ a zda horní hranice y_r nebude příliš vysoká.

V poslední experimentech popsaných v sekci 6.4 zkoumáme, zdali se fázový přechod pro matched formule posune, pokud povolíme v pokrytí omezené bikliky větší než $K_{1,1}$.

2. Definice a značení

Nejdříve si zavedeme definice a značení, které budeme v práci používat.

2.1 Teorie grafů

Vzhledem k tomu, že se budeme v práci zabývat grafy a jejich vlastnostmi, uvedeme si zde některé definice, značení a některé věty z teorie grafů. Pojmy z teorie grafů, které zde nejsou popsány lze nalézt v [13, 29].

Bipartitní graf je uspořádaná trojice $G = (V_1, V_2, E)$, kde $V_1 \cup V_2$ je množina vrcholů grafu G , V_1 a V_2 jsou **partity** bipartitního grafu G a E je množina uspořádaných dvojic takových, že $E \subseteq \{(u, v) | u \in V_1 \wedge v \in V_2\}$. Prvky množiny E jsou **hrany**. Pokud navíc platí $E = \{(u, v) | u \in V_1, v \in V_2\}$, řekneme, že G je **úplný bipartitní graf** nebo též **biklika**. Graf $K_{n,m}$ je biklika s n vrcholy v první partitě a m vrcholy v druhé partitě. Množinu vrcholů grafu G budeme značit $V(G)$, a množinu hran $E(G)$.

Množina sousedů vrcholu v v grafu G je množina $N_G(v) = \{u | (u, v) \in E \vee (v, u) \in E\}$ je množina všech jeho sousedů v grafu G . Stejně jako množinu sousedů vrcholu můžeme definovat množinu sousedů pro množinu vrcholů X v grafu G , kterou je množina $N_G(X) = \cup_{v \in X} N_G(v)$. Stupeň vrcholu v je velikost množiny $N_G(v)$ a budeme ho značit $d_G(v)$.

Množina hran $M \subseteq E(G)$ je **párování** právě tehdy, když pro všechny vrcholy $v \in V(G)$ platí $|\{u | (u, v) \in M \vee (v, u) \in M\}| \leq 1$, tedy každý vrchol grafu G je incidentní maximálně jedné hraně z M . Vrchol $v \in V(G)$ je **pokrytý párováním** M pokud existuje hrana $(u, v) \in M$. Jinak řekneme, že je vrchol v **nepokrytý párováním** M . Párování M grafu G je **maximální párování** grafu G právě tehdy, když pro všechna párování M' grafu G platí $|M'| \leq |M|$.

Věta 2.1.1 ([14, 27]). *Maximální párování lze najít v čase $\mathcal{O}(\sqrt{|V|}|E|)$.*

Věta 2.1.2 ([25]). *Nechť $G = (V_1, V_2, E)$ je bipartitní graf. **Hallová podmínka** je splněna právě tehdy, když pro každou množinu vrcholů $X \subseteq V_1(G)$ platí $|N_G(X)| \geq |X|$.*

V bipartitním grafu $G = (V_1, V_2, E)$ existuje párování velikosti $|V_1|$ právě tehdy, když graf G splňuje Hallovu podmínku.

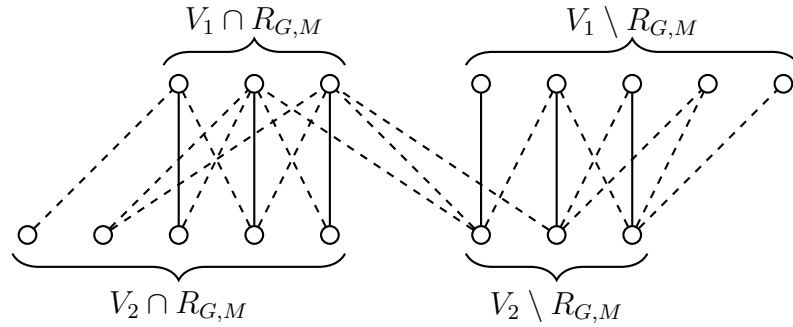
Bipartitní graf G je **q -expandující** právě tehdy, když pro každou množinu $X \subseteq V_1$ platí $|N_G(X)| \geq |X| + q$. Můžeme si povšimnout, že 0-expandující grafy splňují Hallovu podmínku a existuje v nich tedy párování velikosti $|V_1|$.

Pro graf G je sekvence $P = v_1, v_2, \dots, v_k$, pro $v_i \in V(G)$, $i \in \{1, \dots, k\}$ **cesta** právě tehdy, když $(v_i, v_{i+1}) \in E(G)$ pro $i \in \{1, \dots, k-1\}$ a $v_i \neq v_j$ pro $i, j \in \{1, \dots, k\}$. **Alternující cesta** v grafu G s párováním M je cesta $P = v_1, v_2, \dots, v_k$, v níž se střídají hrany, které jsou v párování M s hranami, které v něm nejsou. **Zlepšující cesta** je alternující cesta, u které nejsou krajní vrcholy v_1 a v_k pokryté párováním M . Pokud v grafu G s párováním M existuje zlepšující cesta P , můžeme z párování M vytvořit párování M' s $|M'| = |M| + 1$ a to tak, že M' je symetrickou diferencí párování M a zlepšující cesty P .

Deficience bipartitního grafu $G = (V_1, V_2, E)$ je $\delta(G) = |V_2| - |N_G(V_2)|$. **Maximální deficience** grafu G je $\delta^*(G) = \max_{Y \subseteq V_2} |Y| - |N_G(Y)|$.

Věta 2.1.3 ([29]). *Nechť $G = (V_1, V_2, M)$ je bipartitní graf a M je libovolné maximální párování grafu G . Počet vrcholů, jež nejsou pokryty párováním, je roven maximální deficienci δ^* .*

Díky větě 2.1.3 jsme schopni u bipartitních grafů efektivně zjistit jejich maximální deficienci.



Obrázek 2.1.1: $R_{G,M}$ množina.

Uvažme bipartitní graf $G = (V_1, V_2, E)$ a jeho maximální párování M . Množina $R_{G,M}$ je množina všech vrcholů, kterých lze dosáhnout z libovolného vrcholu $v \in V_2$ nepokrytého párováním M alternující cestou. Příklad takové množiny je zobrazen na obrázku 2.1.1.

Lemma 2.1.4 ([34]). *Nechť $G = (V_1, V_2, E)$ je bipartitní graf, $V = V_1 \cup V_2$ a M je maximální párování grafu G . Pak platí následující:*

- (1) $R_{G,M}$ lze získat v čase $\mathcal{O}(|V| + |E|)$,
- (2) mezi $V_1 \setminus R_{G,M}$ a $V_2 \cap R_{G,M}$ nevede žádná hrana; mezi $V_1 \cap R_{G,M}$ a $V_2 \setminus R_{G,M}$ nevede žádná hrana, která by patřila do M ,
- (3) vrcholy $V_1 \cap R_{G,M}$ a $V_2 \setminus R_{G,M}$ jsou vrcholy pokryté párováním M ,
- (4) pokud G není 0-expandující, pak $|V_1 \setminus R_{G,M}| > |N_G(V_1 \setminus R_{G,M})|$,
- (5) pokud je $R_{G,M}$ neprázdná, pak $R_{G,M}$ indukuje 1-expandující podgraf grafu G .

2.2 Booleovské formule

Nyní si definujeme některé pojmy spjaté s booleovskými formullemi. Další definice, které zde nejsou uvedeny, můžeme nalézt v [9].

Booleovská funkce f je zobrazení $f : \{0, 1\}^n \rightarrow \{0, 1\}$, kde n je počet proměnných booleovské funkce. Hodnota funkce $f(x)$ je **pravda** právě tehdy, když $f(x) = 1$, pro $x \in \{0, 1\}^n$. Jinak řekneme, že $f(x)$ je **nepravda**.

Negaci proměnné x budeme značit \bar{x} , konjunkci proměnných x a y budeme značit $x \wedge y$ a disjunkci proměnných x a y budeme značit $x \vee y$. **Literál** je proměnná nebo její negace. Pro literál x označíme $x^1 = x$ a negace literálu $x^0 = \bar{x}$.

Klause je disjunkcí literálů $C = (x_1 \vee \dots \vee x_n)$, pro kterou platí, že každá proměnná se vyskytuje v nejvýš jednom literálu z x_1, \dots, x_n . Klause budeme

chápat též jako množinu literálů $C = \{x_1, \dots, x_n\}$. Prázdnou klauzuli označíme \perp a hodnota prázdné klauzule je vždy nepravda. Komplement klauzule $C = \{x_1, \dots, x_n\}$ je klauzule $\overline{C} = \{\overline{x_1}, \dots, \overline{x_n}\}$. Proměnná (literál) x se **vyskytuje v klauzuli** C nebo též klauzule C **obsahuje** proměnnou x , pokud platí $x \in C \cap \overline{C}$. **Délka** klauzule je velikost její množiny $|C|$ nebo též počet proměnných, které v ní mají výskyt.

Formule v konjunktivně normální formě (KNF) je konjunkce klauzulí $F = C_1 \wedge \dots \wedge C_m$. Proměnná (literál) x má **výskyt ve formuli** F (formule F **obsahuje** proměnnou x) právě tehdy, když má proměnná (literál) x výskyt v nějaké klauzuli C_i , $1 \leq i \leq m$, formule F . Množinu všech proměnných, které mají ve formuli F výskyt budeme značit $var(F)$. Formule F je v **k -konjunktivně normální formě (k -KNF)** právě tehdy, když každá klauzule obsahuje právě k literálů. Pro formuli F a proměnnou (literál) x budeme označovat $\#_x$ počet výskytů proměnné (literálu) x ve formuli F .

Literál x je **čistý** právě tehdy, když jeho negace nemá výskyt ve formuli F tedy $\#\overline{x} = 0$. Literál x je **singulární** právě tehdy, když má výskyt ve formuli F a pro formuli F platí $\#_x = 1 \wedge \#\overline{x} \geq 1$.

Pravdivostní ohodnocení formule F je zobrazení z množiny proměnných $X \subseteq var(F)$ do množiny $\{0, 1\}$, tedy $\tau : X \rightarrow \{0, 1\}$. Množinu proměnných ohodnocení τ budeme značit $var(\tau)$. Pro proměnnou $x \in var(\tau)$ je $\tau(\overline{x}) = 1 - \tau(x)$. Proměnná x je pravda (TRUE) právě tehdy, když $\tau(x) = 1$, jinak je x nepravda (FALSE).

Mějme pravdivostní ohodnocení τ formule F s $var(\tau) \subseteq var(F)$. Pak **dosazením ohodnocení τ formule F** vznikne formule $F' = F[\tau]$ taková, že $F' = \bigwedge \{C \setminus \tau^{-1}(0) \mid C \in F, C \cap \tau^{-1}(0) = \emptyset\}$. Ohodnocení τ je **částečné ohodnocení** formule F právě tehdy, když platí $var(\tau) \subsetneq var(F)$. Ohodnocení τ je **úplné ohodnocení** formule F právě tehdy, když platí $var(\tau) = var(F)$. Ohodnocení τ **splňuje klauzuli** C právě tehdy, když existuje proměnná x , vyskytující se v klauzuli C taková, že $\tau(x) = 1$ pokud $x \in C$ a $\tau(x) = 0$ pokud $\overline{x} \in C$. Prázdná klauzule je nesplnitelná. Ohodnocení τ **splňuje formuli** F právě tehdy, když splňuje všechny klauzule formule F . Formule F je **splnitelná** právě tehdy, když existuje nějaké ohodnocení τ , které splňuje F . Jinak je formule **nesplnitelná**. Prázdná formule je splnitelná. Formule, která obsahuje prázdnou klauzuli, je nesplnitelná. Ohodnocení τ formule F je **autarky** právě tehdy, když splňuje všechny klauzule F , které obsahují nějakou proměnnou z $var(\tau)$. Formule F, F' jsou **ekvisplnitelné** právě tehdy, když jsou obě splnitelné nebo obě nesplnitelné. Tento fakt zapíšeme $F \equiv_{sat} F'$

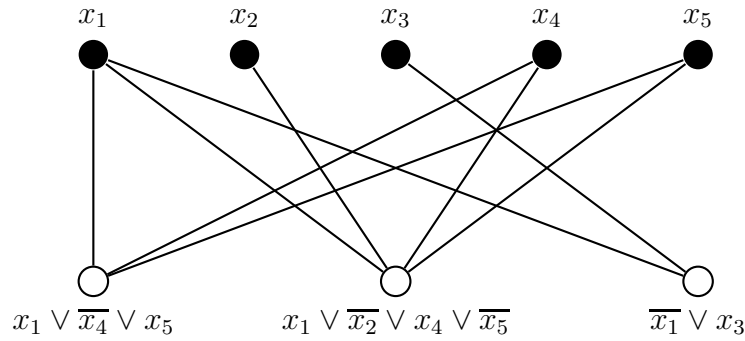
Rezolventou klauzulí C_1 a C_2 vzhledem k literálu x nazveme klauzuli $C_x = (C_1 \cup C_2) \setminus \{x, \overline{x}\}$. Rezolventa dvou klauzulí C_1 a C_2 vzhledem k literálu x je definována jen pro klauzule, pro které platí $\{x, \overline{x}\} \subseteq C_1 \cup C_2$.

Mějme formuli F a libovolný literál x , který se vyskytuje ve formuli F . Formule F' vznikne z formule F přidáním všech možných rezolvent vzhledem k x a odstraněním všech klauzulí, ve kterých se vyskytuje x . Řekneme, že F' je z F odvozená Davis-Putnamovou rezolucí. Tento fakt lze zapsat též $DP_x(F) = F'$. Tento rezoluční krok používá Davis-Putnamova procedura[12] pro řešení splnitelnosti.

Definice. *Nechť F je formule v konjunktivně normální formě. **Incidenční graf** formule F je bipartitní graf $\mathbf{I}(F) = (V_{var}, V_C, E)$, kde*

- V_{var} odpovídá proměnným vyskytujícím se ve formuli F ,

- V_C odpovídá klauzulím formule F ,
- proměnná v a klauzule c jsou spojeny hranou právě tehdy, když má v výskyt v c .



Obrázek 2.2.1: Incidenční graf formule F

Na obrázku 2.2.1 je zobrazen incidenční graf formule v konjunktivně normální formě

$$F = (x_1 \vee \overline{x_4} \vee x_5) \wedge (x_1 \vee \overline{x_2} \vee x_4 \vee \overline{x_5}) \wedge (\overline{x_1} \vee x_3)$$

Nyní můžeme převést některé pojmy z teorie grafů na booleovské formule. Uvažme formuli F v konjunktivně normální formě. Formule F má **deficienci** $\delta(F)$ právě tehdy, když má její incidenční graf $I(F)$ deficienci $\delta(I(F)) = \delta(F)$. Podobně definujeme maximální deficienci formule $\delta^*(I(F)) = \delta^*(F)$. Formule F je **k -expandující** právě tehdy, když je incidenční graf $I(F)$ k -expandující.

2.3 Složitost

Na závěr si definujeme několik pojmů ze složitosti a parametrizované složitosti. Definice neuvedené zde lze najít v [4, 16, 20, 24, 31].

Pro funkce $f, g : \mathbb{N} \rightarrow \mathbb{N}$ funkce řekneme, že:

- $f(n) = \mathcal{O}(g(n))$ právě tehdy, když existuje konstanta $c > 0$ a n_0 takové, že pro každé $n \geq n_0$ platí $f(n) \leq c \cdot g(n)$,
- $f(n) = \Omega(g(n))$ právě tehdy, když existuje konstanta $c > 0$ a n_0 takové, že pro každé $n \geq n_0$ platí $f(n) \geq c \cdot g(n)$,
- $f(n) = \Theta(g(n))$ právě tehdy, když platí $f(n) = \mathcal{O}(g(n))$ a zároveň $f(n) = \Omega(g(n))$.

Množinu znaků Σ označíme jako abecedu a Σ^* je množina všech slov všech délek z abecedy Σ . Jazyk L je množina slov $L \subseteq \Sigma^*$. **Rozhodovací problém** je dotaz, zda nějaké slovo $x \in \Sigma^*$ náleží do jazyka L .

Definice. *Deterministický k -páskový Turingův stroj je abstraktní stroj, definovaný pěticí $(Q, \Sigma, \delta, q_0, F)$, kde*

Q je konečná množina stavů, ve kterých se Turingův stroj může nacházet,

Σ je množina znaků, které můžou pásky Turingův stroje obsahovat,

δ je přechodová funkce $\delta : Q \times \Sigma^k \rightarrow Q \times \Sigma^{k-1} \times \{\leftarrow, \bullet, \rightarrow\}^k$,

q_0 je počáteční stav $q_0 \in Q$,

F je množina přijímacích stavů $F \subseteq Q$.

Nedeterministický k -páskový Turingův stroj je abstraktní stroj definovaný pěticí $(Q, \Sigma, \delta, q_0, F)$. Tato pětica je téměř stejná, jako u deterministického Turingova stroje. Nedeterministický stroj se od deterministického liší pouze v přechodové funkci:

δ je přechodová funkce $\delta : Q \times \Sigma^k \rightarrow 2^{Q \times \Sigma^{k-1} \times \{\leftarrow, \bullet, \rightarrow\}^k}$,

Konfigurace Turingova stroje je stav řídicí jednotky $q \in Q$, pozice hlav na všech páskách a obsah všech pásek. **Počáteční konfigurace Turingova stroje** je konfigurace Turingova stroje, kde řídicí jednotka je ve stavu q_0 , hlava vstupní pásky je na začátku vstupu a pracovní pásky jsou všechny prázdné. **Displej Turingova stroje** je stav řídicí jednotky $q \in Q$, spolu s obsahem všech buněk na páskách pod hlavami. **Krok Turingova stroje** je aplikace přechodové funkce na aktuální displej Turingova stroje. **Výpočet Turingova stroje** je posloupnost kroků začínající v počáteční konfiguraci Turingova stroje. **Zastavení Turingova stroje** je situace, kdy pro daný displej Turingova stroje není definována přechodová funkce. **Přijímající výpočet Turingova stroje** je výpočet Turingova stroje, který se zastaví v přijímacím stavu $f \in F$. **Odmítající výpočet Turingova stroje** je výpočet Turingova stroje, který se zastaví ve stavu jiném než přijímacím nebo se nezastaví. **Přijímané slovo Turingovým strojem** je slovo, pro které je výpočet na deterministickém Turingově stroji přijímací, nebo pro které existuje přijímací výpočet na nedeterministickém Turingově stroji. **Přijímaný jazyk Turingovým strojem** je jazyk $L(M)$, jehož slova jsou přijímané Turingovým strojem. Zároveň pro všechna slova x přijímaná Turingovým strojem platí $x \in L(M)$.

Takto definovaný Turingův stroj je **akceptor**. Modifikujeme-li deterministický Turingův stroj tak, že přidáme výstupní pásku pouze pro zápis, na které povolíme jen pohyb doprava a znak pod výstupní hlavou nezahrneme do displeje, dostaneme deterministický Turingův stroj, kterému říkáme **transducer**. Takovýto stroj interpretuje funkci $f : \Sigma^* \rightarrow \Sigma^*$. Definiční obor funkce f je přijímaný jazyk stroje $L(M)$ a obor hodnot je daný možnými obsahy výstupní pásky v momentu zastavení.

Turingův stroj M má **nedeterministickou časovou složitost** $T(n)$ právě tehdy, když existuje nějaký nedeterministický Turingův stroj, který nad vstupním slovem x provede maximálně $T(n)$ kroků, v jakémkoliv možném průběhu výpočtu, než se zastaví. Jazyk L má **nedeterministickou časovou složitost** $T(n)$ právě tehdy, když existuje Turingův stroj, který má nedeterministickou časovou složitost $T(n)$ a přijímá jazyk L .

Definice. Nechť $T : \mathbb{N} \rightarrow \mathbb{N}$ je funkce. Označme $\mathbf{NTIME}(T(n)) = \{L \mid L \text{ má nedeterministickou časovou složitost } T(n)\}$. Třidu **NP** definujeme jako třídu $\mathbf{NP} = \cup_{c \geq 0} \mathbf{NTIME}(n^c)$.

Funkce $f : \{0, 1\}^* \rightarrow \{0, 1\}^*$ je **polynomiálně vyčíslitelná** právě tehdy, když existuje polynom p a deterministický transducer A takový, že pro každý vstup $x \in \{0, 1\}^*$ je výstup deterministického transduceru výstup $f(x)$ po vykonání nejvýše $p(|x|)$ kroků, kde $|x|$ označuje délku slova x .

Jazyk L je **polynomiálně převoditelný** na L' právě tehdy, když existuje polynomiálně vyčíslitelná funkce $f : \Sigma^* \rightarrow \Sigma^*$ taková, že pro všechna slova $x \in \Sigma^*$ platí $x \in L \Leftrightarrow f(x) \in L'$. Tento fakt označíme $L \propto L'$.

Problém Q je **NP-těžký** právě tehdy, když na jazyk $L(Q)$ je polynomiálně převoditelný libovolný jazyk $L(R)$ problému $R \in \text{NP}$.

Definice. *Problém Q je NP-úplný právě tehdy, když je problém Q NP-těžký a zároveň platí, že $Q \in \text{NP}$.*

Doplňek jazyka L je jazyk **co-L** $= \{x \in \Sigma^* | x \notin L\}$. Pro třídu jazyků C je doplněk třídy jazyků **co-C** třída jazyků $\text{co-C} = \{\text{co-L} | L \in C\}$.

Definici pro **polynomiální hierarchii** zde uvádět nebudeme. V textu diplomové práce zmiňujeme pouze třídu problémů Π_2 , která spadá do druhé úrovně polynomiální hierarchie a spadají do ní problémy tříd NP a co-NP [31, 33].

Parametrizovaný problém je jazyk $L \subseteq \Sigma^* \times \mathbb{N}$. Parametrizovaný problém L je **řešitelný s pevným parametrem (fixed-parameter trackable, FPT)**, pokud existuje deterministický algoritmus, který pro danou dvojici (I, k) rozhodne, zda $(I, k) \in L$ a pracuje v čase $\mathcal{O}(f(k) \cdot n^{O(1)})$, kde $f(k)$ je algoritmicky vyčíslitelná funkce.

3. Třídy formulí

Každou formuli F v konjunktivně normální formě lze asociovat s bipartitním grafem $I(F) = (V_{var}, V_C, E)$, kterému říkáme incidenční graf formule F . Tento přístup lze využít k tomu, abychom popsali některé třídy formulí za využití prostředků teorie grafů. Tyto třídy si nadefinujeme v této kapitole.

3.1 Matched formule

Definice. *Formule F je **matched**, právě tehdy když její incidenční graf $I(F)$ má párování M takové, že $|M| = |V_C|$.*

Lemma 3.1.1 ([2, 21, 35, 36]). *Každá matched KNF formule je splnitelná.*

Důkaz. Necht' $I(F) = (V_{var}, V_C, E)$ je incidenční graf KNF formule F a necht' M je maximální párování.

Proměnnou v ohodnotíme FALSE, pokud se v klauzuli c takové, že $(v, c) \in M$, vyskytuje v negaci a TRUE jinak. Tímto ohodnocením proměnných dostaneme ohodnocení, které splňuje všechny klauzule, které jsou pokryté párováním M .

Protože F je matched, víme, že $|M| = |V_C|$ a jsou tedy splněny všechny klauzule. \square

Na rozpoznání toho, zda je formule matched, stačí využít libovolného algoritmu na hledání maximálního párování v bipartitních grafech, kupříkladu Dinitzova algoritmu [14]. Modifikace Dinitzova algoritmu na hledání maximálního párování v bipartitním grafu, Hopcroft-Karpův algoritmus, pracuje v polynomiálním čase [27] $\mathcal{O}(m\sqrt{n})$, kde m je počet hran grafu a n je počet vrcholů.

John Franco a Allen Van Gelder [21] zkoumali, s jakou pravděpodobností je náhodná formule matched. Pomocí r_F označme poměr počtu klauzulí k počtu proměnných v KNF F . Triviálně platí, že je-li $r_F > 1$, pak formule F nemůže být matched, na druhou stranu prázdná KNF s poměrem $r = 0$ jistě matched je. Autoři [21] si kladli otázku, při jakém poměru se změní jistota toho, že formule je matched, v jistotu toho, že matched není. Ukázali přitom, že uvažujeme-li náhodnou 3-KNF F s poměrem počtu klauzulí k počtu proměnných $r_F < 0,64$, pak náhodná formule 3-KNF pravděpodobnost, že daná náhodná formule je matched se blíží 1 s rostoucím počtem proměnných. Platí tedy, že náhodná formule v 3-KNF je skoro jistě matched, je-li poměr počtu klauzulí k počtu proměnných menší než 0,64. Jedním z cílů této práce je experimentální ověření tohoto odhadu, kterému se budeme věnovat v kapitole 6.

3.2 Biklikově splnitelné formule

Szeider v [35] zavedl dvě třídy zobecňující matched formulí. Zobecnění popsané v této kapitole je založeno na pokrývání incidenčního grafu formule omezenými biklikami.

Definice. *Omezenou biklikou označíme úplný bipartitní graf $G = (V_{var}, V_C, E)$ takový, že $|V_C| < |2^{V_{var}}|$.*

Na základě této definice není těžké nahlédnout, že formule, jejichž incidenční graf tvoří omezenou bikliku, jsou vždy splnitelné.

Lemma 3.2.1 ([35]). *Nechť F je formule s n proměnnými a m klauzulemi. Pokud je graf $I(F) = (V_{var}, V_C, E)$ omezenou biklikou, pak je formule F splnitelná.*

Důkaz. Klauzulí je maximálně $2^n - 1$. Každá klauzule obsahuje n proměnných a tedy zakazuje právě jedno z 2^n pravdivostních ohodnocení. Musí tedy existovat alespoň jedno, které splňuje všechny klauzule. \square

Pochopitelně každá hrana je omezená biklika, což odpovídá tomu, že KNF skládající se z jediné jednotkové klauzule, je vždy splnitelná. Můžeme však uvažovat i větší omezené bikliky, například bikliky se dvěma proměnnými a třemi klauzulemi, které jsou vždy splnitelné. Místo pokrývání incidenčního grafu po dvou disjunktními hranami, tedy párováním, které pokrývá všechny klauzule, můžeme nyní uvažovat pokrývání incidenčního grafu po dvou disjunktními omezenými biklikami.

Lemma 3.2.2 ([35]). *Nechť F je formule v KNF a nechť existuje množina $\mathcal{B} = \{B_1, \dots, B_k\}$ množin vrcholů incidenčního grafu $I(F)$ taková, že*

- (1) *každá klauzule formule F patří do právě jedné množiny B_i , $1 \leq i \leq k$,*
- (2) *každá proměnná formule F patří maximálně do jedné množiny B_i , $1 \leq i \leq k$,*
- (3) *každá množina B_i indukuje v grafu $I(F)$ bikliku.*

Pak je formule F splnitelná.

Důkaz. Podle lemmatu 3.2.1 můžeme pro každou bikliku nalézt splňující ohodnocení. Díky podmínce (3) nalezneme nějaké částečné ohodnocení β_i , které splňuje bikliku B_i . Vzhledem k podmínce (1) jsou i množiny částečných ohodnocení různých biklik B_i disjunktní a můžeme je tedy složit do jednoho totálního ohodnocení $\beta = \cup_{i=1}^k \beta_i \cup \gamma$, kde γ je libovolné částečné ohodnocení proměnných, které nejsou obsaženy v žádné množině B_i .

Ohodnocení β splňuje všechny klauzule, které jsou v nějaké množině B_i a vzhledem k podmínce (2) je splněna celá formule. \square

Nyní již můžeme přikročit k definici biklikově splnitelných formulí.

Definice. *Formule F je **biklikově splnitelná** právě tehdy, když existuje množina \mathcal{B} splňující podmínky lemma 3.2.2. Množinu \mathcal{B} nazveme **pokrytím formule F omezenými biklikami**.*

Ukazuje se, že testování toho, zda je daná KNF biklikově splnitelná, je NP-úplné [35], a to již omezíme-li se pokrytí omezenými biklikami, jež obsahuje pouze bikliky $K_{2,3}$.

Definice. *Nechť formule F je biklikově splnitelná a množina \mathcal{B} je pokrytí F omezenými biklikami. Formule je **k -biklikově splnitelná**, pokud $\forall B \in \mathcal{B}$ platí $|B \cap V_{var}| \leq k$.*

U 1-biklikové splnitelnosti je zřejmé, že půjde rozpoznat v polynomiálním čase. K tomu si stačí uvědomit, že biklika, která obsahuje právě jednu proměnnou, může mít maximálně jednu klauzuli (definice omezené bikliky, sekce 3.2) a je tedy párovací hranou. Toto tedy znamená, že problém 1-biklikové splnitelnosti je ekvivalentní problému rozpoznání matched formulí a ten je podle části 3.1 řešitelný v polynomiálním čase.

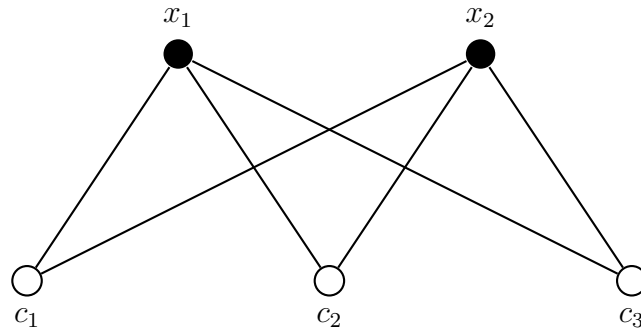
Pro KNF formule, které jsou biklikově splnitelné, ale nejsou matched, se nic podobného tvrdit nedá.

Pro $k \geq 2$ je rozpoznání k -biklikové splnitelnosti 3-KNF formulí NP-úplné, jak ukázal Stefan Szeider[35].

Příklad 3.2.3. *Formule*

$$F = (x_1 \vee x_2) \wedge (\overline{x_1} \vee x_2) \wedge (\overline{x_1} \vee x_2)$$

je biklikově splnitelná, ale není matched.



Obrázek 3.2.1: Incidenční graf formule F

Formule F je omezenou biklikou, je tedy biklikově splnitelná a to, že formule není matched, je zřejmé z toho, že formule má víc klauzulí než proměnných.

Z příkladu 3.2.3 a z toho, že matched formule jsou 1-biklikově splnitelné, lze tedy usoudit, že matched formule jsou tedy v ostré inkluzi biklikově splnitelných formulí.

3.3 Var-splnitelnost

Zajímavou vlastností matched i biklikově splnitelných formulí je to, že tyto formule zůstanou splnitelné i tehdy, změníme-li libovolně polaritu některých výskytů některých proměnných. To vyplývá z toho, že o tom, zda je formule F biklikově splnitelná, rozhoduje pouze struktura incidenčního grafu $I(F)$. Při definici incidenčního grafu však nehraje polarita proměnných v klauzulích roli. Szeider v [35] proto uvažoval třídu tzv. var-splnitelných formulí, které mají právě tuto vlastnost.

Definice. *Formule F je **var-splnitelná** právě tehdy, když každá formule F' taková, že $I(F) = I(F')$, je splnitelná.*

Problém rozpoznání, zda je 2-KNF var-splnitelná, lze vyřešit v lineárním čase, jak ukázali Kleine Bunig a Zhao [5].

Pro formule, s délkou klauzulí větší nebo rovnou třem, je problém rozpoznání, zda jsou var-splnitelné Π_2^P -úplný, jak ukázal Stefan Szeider [35].

Pozorování 3.3.1. *Matched formule i biklikově splnitelné formule jsou var-splnitelné.*

Toto je vidět přímo z konstrukce incidenčního grafu formule, kde nebereme v potaz polaritu literálů.

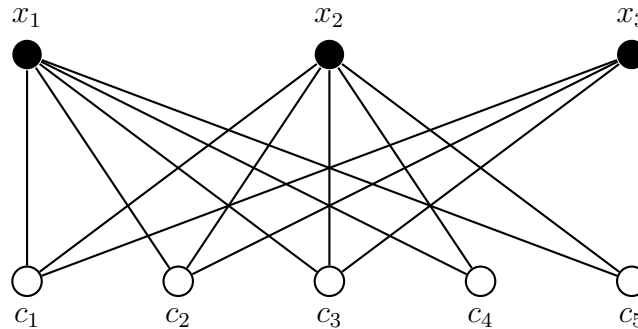
Naopak var-splnitelná formule nemusí nutně být biklikově splnitelná.

Příklad 3.3.2. *Formule*

$$F = (x_1 \vee x_2 \vee x_3) \wedge (\overline{x_1} \vee x_2 \vee x_3) \wedge (\overline{x_1} \vee x_2 \vee \overline{x_3}) \wedge (x_1 \vee \overline{x_2}) \wedge (\overline{x_1} \vee \overline{x_2})$$

je var-splnitelná, ale není biklikově splnitelná.

Přímo z incidenčního grafu formule je vidět, že není biklikově splnitelná.



Obrázek 3.3.1: Incidenční graf formule F

Nejmenší možná omezená biklika, která by mohla pokrýt tento graf, je $K_{3,5}$, tedy celý incidenční graf by musel být úplný.

Protože je var-splnitelná, použijeme podobný argument jako u lemma 3.2.1. Každá z prvních třech klauzulí zakazuje jedno pravdivostní ohodnocení a poslední dvě zakazují po dvou pravdivostní ohodnocení. Dohromady tedy máme zakázáno sedm ohodnocení a všech ohodnocení na třech proměnných je osm. Bez ohledu na polaritu proměnných tedy existuje splňující ohodnocení.

Tímto dostaneme vzájemné inkluze jednotlivých tříd:

Pozorování 3.3.3.

Matched formule \subsetneq biklikově splnitelné formule \subsetneq var-splnitelné formule

Z těchto inkluzí je zřejmé, že ani u jedné z těchto tříd formulí nezáleží na polaritě jednotlivých literálů. Na rozpoznávání toho, zda patří formule do jedné z těchto tříd, lze využít algoritmů a heuristik z teorie grafů.

4. Formule s malou deficiencí

V této kapitole popíšeme parametrizovaný algoritmus pro testování splnitelnosti formule, kde parametrem je její maximální deficiencie. Jde o algoritmus popsany v [34].

4.1 Parametrizovaný algoritmus

Polynomiální algoritmus pro formule s omezenou deficiencí popsal Herbert Fleischner a kol. v [18]. Výsledek lze formulovat do věty 4.1.1.

Věta 4.1.1 ([18]). *Formule F je splnitelná právě tehdy když $F[\tau]$ je matched pro nějaké částečné ohodnocení τ , kde $|\text{var}(\tau)| \leq \delta^*(F)$.*

Polynomiální algoritmus získáme tak, že pro všechny možné kombinace proměnných velikosti maximálně $\delta^*(F)$ zkusíme všechna ohodnocení a poté otestujeme formuli, jestli je matched.

Časová složitost tohoto algoritmu je $\mathcal{O}(|F|\text{var}(F)^{k+\frac{1}{2}})$, čímž se s pro větší $\delta^*(F)$ stává zcela nepoužitelným.

V této části se budeme zabývat parametrizovaným algoritmem, který tento problém řeší s pevným parametrem $k = \delta^*(F)$ (patří do třídy FPT) a který popsal Szeider [34].

Pro to, abychom popsali algoritmus, budeme potřebovat následující definici.

Definice. *Nechť F je formule v KNF a nechť pro každé $(x, \varepsilon) \in \text{var}(F) \times \{0, 1\}$ platí $\delta^*(F[x = \varepsilon]) \leq \delta^*(F) - 1$. Pak je formule F **δ^* -kritická**.*

Algoritmus popsany v [34] je variantou DLL algoritmu (Davis, Logemann, Loveland[11]), který provádí rozvětvení (formule F na $F[x = 0]$ a $F[x = 1]$ pro x zvolené libovolně) jen pro δ^* -kritické formule. Při každém větvení dojde ke snížení deficiencie $k = \delta^*(F)$ vstupní formule F a tudíž je hloubka prohledávacího stromu shora omezená maximální deficiencí.

Věta 4.1.2 ([34]). *Testovat splnitelnost formule F s n proměnnými a maximální deficiencí k lze v čase $\mathcal{O}(2^k n^3)$.*

V této kapitole si ukážeme, jak lze provést redukce formule F , které vedou k formuli F' , která je δ^* -kritická. Tuto redukci lze provést v polynomiálním čase. Pro naše potřeby není potřeba uvádět časovou složitost a rozbor složitosti uvedených postupů společně s důkazy všech vět lze nalézt v [34].

Uvažme sekvenci $S = (F_0, M_0), \dots, (F_q, M_q)$, kde F_i je formule a M_i je maximální párování $I(F_i)$, $0 \leq i \leq q$. Řekneme, že S je **redukční posloupnost**, pokud pro všechna $i \in \{1, \dots, q\}$ platí jedno z následujících:

- (1) $F_i = F_{i-1}$ pro neprázdné autarky α_i nebo
- (2) $F_i = \text{DP}_x(F_{i-1})$ pro singulární literál x_i formule F_{i-1} .

Nyní si popíšeme vznik redukční posloupnosti formule F , které povedou ke vzniku formule F_q , jež bude mít buď menší deficienci $\delta^*(F_q)$ než formule $\delta^*(F)$, nebo bude splňovat větu 4.1.3.

Věta 4.1.3 ([34]). *2-expandující formule bez čistých a singulárních literálů jsou δ^* -kritické.*

4.2 Preproces redukce

Lemma 4.2.1 ([34]). *Nechť F_0 je formule s n proměnnými, $\delta^*(F_0) \leq n$ a M_0 maximální párování $I(F_0)$. Pak můžeme najít v čase $\mathcal{O}(n^3)$ redukční sekvenci $S = (F_0, M_0), \dots, (F_q, M_q)$, $q \leq n$ takovou, že platí jedno z následujících:*

- (1) $\delta^*(F_q) \leq \delta^*(F_0) - 1$ nebo
- (2) $\delta^*(F_q) = \delta^*(F_0)$, a F_q je 1-expandující bez čistých a singulárních literálů

K tomu, abychom splnili podmínky, využijeme následující tři typy redukce formule F_0 . Tyto redukce budeme opakovat do té doby, než splníme jednu z podmínek lemmatu 4.2.1. Podmínky budeme testovat v daném pořadí:

- (1) Redukce $R_{G,M}$
- (2) Odstranění čistých literálů
- (3) Odstranění singulárních literálů

4.2.1 Redukce $R_{G,M}$

Libovolnému párování M grafu incidence $I(F)$ přísluší částečné ohodnocení τ_M , kde pro každou dvojici $(x, C) \in M$ je $\tau_M(x) = 1$ pokud $x \in C$ a $\tau_M(x) = 0$ pokud $\bar{x} \in C$.

Mějme formuli F_{i-1} a její incidenční graf $G_{i-1} = I(F_{i-1}) = (V_{var}, V_C, E)$, maximální párování M_{i-1} a množinu $R_{G_{i-1}, M_{i-1}}$.

Označme částečné ohodnocení $\alpha = \tau_{M'_{i-1}}$, kde $M'_{i-1} = \{(x, C) \mid (x, C) \in M_{i-1} \wedge x \notin R_{G_{i-1}, M_{i-1}}\}$. Toto částečné ohodnocení je autarky, což přímo plyne z vlastností množiny $R_{G_{i-1}, M_{i-1}}$.

Touto aplikací dostaneme $F_i = F_{i-1}[\alpha]$ a maximální párování F_i je $M_i = M_{i-1} \cap I(F_i)$. Incidenční graf $I(F_i)$ je indukován množinou $R_{G_{i-1}, M_{i-1}}$.

4.2.2 Odstranění čistých literálů

Nechť x^ε je čistý literál a pro libovolné $\varepsilon \in \{0, 1\}$. Pak formule $F_i = F_{i-1}[x = \varepsilon]$ a F_{i-1} jsou ekvisplnitelné a $x = \varepsilon$ je autarky.

Vzhledem k tomu, že F_{i-1} je 1-expandující, platí $\delta^*(F_i) \leq \delta^*(F_{i-1}) - 1$. Párování $M'_i = M_{i-1} \cap I(F_i)$ není nutně maximální, ale počet nepokrytých klauzulí F_i je maximálně takový, jako počet nepokrytých klauzulí v F_{i-1} , tudíž potřebujeme nejvíce $\delta^*(F_i)$ zlepšení pro nalezení maximálního párování M_i .

Pokud provedeme tuto redukci, dosadíme $q = i$ a další redukce už neaplikujeme, protože jsme splnili podmínku 1 lemma 4.2.1.

4.2.3 Odstranění singulárních literálů

Nechť x^ε je singulární literál formule F_{i-1} . Formule F_i vznikne aplikací DP rezoluce, tedy $F_i = \text{DP}_x(F_{i-1})$.

Pro nalezení maximálního párování musíme rozlišit dva případy:

- (1) počet klauzulí F_i je o jedna menší než počet klauzulí F_{i-1}

Označme C_x klauzuli spárovanou s proměnnou x , C_y klauzuli, která obsahuje literál x^e a y proměnnou, která je spárovaná s C_y . Pokud $C_y = C_x$, pak $M_i = M_{i-1} \cap I(F_i)$. Jinak F_i obsahuje klauzuli $\{C_x \cap C_y\} \setminus x, \bar{x}$. Označme tuto klauzuli C_{xy} .

Nové párování pak bude $M_i = \{M_{i-1} \cap I(F_i)\} \cup (y, C_{xy})$. Vzhledem k tomu, že se maximální deficiencie nesníží, budeme pokračovat v redukci.

- (2) počet klauzulí F_i je alespoň o dva menší než počet klauzulí F_{i-1}

Nové párování můžeme dostat novým spuštěním Dinicova algoritmu [14, 27] a redukci končíme, tedy $q = i$. Tímto postupem dosáhneme trochu horší časové složitosti. Popis rychlejšího řešení a důkaz toho, že podmínka 1 lemmatu 4.2.1 je splněna, jsou popsány v [35].

4.3 2-expandující redukce

Pokud je formule F_q 1-expandující a neobsahuje čisté ani singulární literály, nemůžeme aplikovat žádnou redukci na daný graf. Nyní chceme zjistit, je-li formule δ^* -kritická. Na to nám stačí, aby splňovala větu 4.1.3.

Formule F_q neobsahuje čisté ani singulární literály. Musíme tedy zjistit, je-li formule F_q 2-expandující a pokud není, najít redukci takovou, která sníží maximální deficienci.

Následující věta nám poskytne prostředek k tomu, abychom mohli efektivně otestovat, zda je graf q -expandující. K tomu budeme potřebovat graf G_{qx} , který vznikne z bipartitního grafu $G = (V_{var}, V_C, E)$ tak, že vytvoříme q kopií vrcholu x a spojíme je hranou se všemi sousedy vrcholu x , tedy $G_{qx} = (V_{var} \cup \{x_1, \dots, x_q\}, V_C, E \cup \{(x_i, y) | 1 \leq i, \leq q, (x, y) \in E\})$.

Věta 4.3.1 ([29]). *0-expandující graf $G = (V_{var}, V_C, E)$ je q -expandující právě tehdy, když G_{qx} je 0-expandující pro každé $x \in V_{var}$.*

Lemma 4.3.2 ([34]). *Nechť $G = (V_{var}, V_C, E)$ je bipartitní graf, M jeho maximální párování a $q \geq 0$ libovolné celé číslo. Rozhodnout, zda je G q -expandující, a pokud není, tak nalézt „svědka“ $X \subseteq V_{var}$, pro kterého platí $|N_G(X)| < |X| + q$, lze v čase $\mathcal{O}(|V_{var}| \cdot |E| + |V_C|)$.*

Pro každý vrchol $x \in V_{var}$ provedeme následující proceduru. Vytvoříme graf $G_{qx} = (V'_{var}, V_C, E')$ podle předchozí konstrukce. Párování M grafu G je též párováním grafu G_{qx} a vrcholy $\{x_1, \dots, x_q\}$ jsou všechny vrcholy z V'_{var} nepokryté párováním M . Nyní se pokusíme rozšířit pomocí zlepšujících cest párování M na párování M' . G_{qx} je 0-expandující právě tehdy, když $|M'| = |V'_{var}| = |V_{var}| + q$.

Pokud G_{qx} není 0-expandující, obsahuje V'_{var} vrcholy nepokryté párováním M' . Nyní sestrojíme množinu $R_{G_{qx}, M'}$ a množinu $V'_{var} \setminus R_{G_{qx}, M'}$ označíme X' . Všechny vrcholy nepokryté párováním M' patří do množiny $\{x_1, \dots, x_q\}$. Množina $X = X' \setminus \{x_1, \dots, x_q\}$ je „svědkem“ toho, že graf G není q -expandovatelný.

Pokud tohoto „svědka“ nenalezneme pro žádný vrchol x , graf G je q -expandující.

Lemma 4.3.3 ([34]). *Budte F 1-expandující formule bez čistých a singulárních literálů a $X \subseteq \text{var}(F)$, $|N_{I(F)}(X)| \leq |X| + 1$. Pak $F \setminus N_{I(F)}(X) \equiv_{\text{sat}} F$ a $\delta^*(F \setminus N_{I(F)}(X)) \leq \delta^*(F) - 1$.*

Pomocí lemma 4.3.2 můžeme efektivně otestovat, je-li formule F_q 2-expandující. Pokud by byla, provedeme rozvětvení.

Jinak máme díky lemma 4.3.2 „svědka“, tedy množinu vrcholů X takovou, že $|N_{I(F)}(X)| \leq |X| + 1$. Maximální deficiencie podformule F' formule F_q , která obsahuje jen proměnné z množiny X , je nejvýš jedna. Díky lemmatu 4.3.3 je tato podformule splnitelná a můžeme tedy nalézt autarky α na proměnných z množiny X , pro které platí $\delta^*(F_q[\alpha]) \leq \delta^*(F_q) - 1$. V tomto případě žádné rozvětvení neproběhne, ale redukce začne znova od prvního kroku, tedy redukcí $R_{G,M}$ 4.2.1.

5. Pokrytí bipartitního grafu omezenými biklikami

V této kapitole si definujeme problémy pokrytí grafu biklikami a prozkoumáme, jaká je jejich složitost. Vzhledem k tomu, že problém pokrytí bipartitního grafu omezenými biklikami je NP-úplný, navrhneme heuristiku, která se bude tento problém snažit vyřešit. Nakonec zkusíme rozšířit přístup popsany v kapitole 4, který využívá párování, na přístup využívající pokrytí omezenými biklikami.

5.1 Algoritmy

Uvažme bipartitní graf G . Základní otázkou, kterou můžeme položit, je ta, zda existuje nějaká množina $\mathcal{B} = \{B_1, \dots, B_k\}$, kde každá množina B_i je množina vrcholů, splňující podmínky

- (1) každá množina B_i indukuje úplný podgraf grafu G ,
- (2) každý vrchol je obsažen v právě jedné množině B_i .

Vyřešit takovýto rozklad je triviální. Vybereme libovolný vrchol v bipartitního grafu G a všechny jeho sousedy $N_G(v)$. Množina $\{v\} \cup N_G(v)$ indukuje úplný podgraf v G , proto můžeme tuto množinu označit jako B_i pro nové i a opakovat tento krok na grafu indukovaném vrcholy $V(G) \setminus (\{v\} \cup N_G(v))$.

Přidáním následující podmínky dostaneme další dva problémy, jejichž složitost už je značně těžší.

- (3) velikost množiny \mathcal{B} je maximálně k

První možností je to, že parametr k , tedy počet podgrafů, je vstupním parametrem problému. Takto definovaný problém nazveme *BVPP* (Biclique Vertex Partition Problem). Tento problém je NP-úplný [26] a neexistuje pro něj k -aproximační algoritmus běžící v polynomiálním čase, kde k je maximální počet biklik pokrytí \mathcal{B} [3, 19].

Uvažme nyní k jako konstantu, která je součástí zadání problému *BVPP* a označme takto parametrizovaný problém *BVPP*(k). Problém *BVPP*(k) lze vyřešit v čase $\mathcal{O}(VE)$ pro $k = 2$ [3] a pro $k \geq 3$ je *BVPP*(k) NP-úplný, ačkoli pro něj existuje aproximační algoritmus s aproximačním poměrem 2, který pracuje v polynomiálním čase $\mathcal{O}(E^{k+1}V)$ [3] a pro problém *BVPP* neběžící v polynomiálním čase, protože u problému *BVPP* je k proměnná.

U problému biklikové splnitelnosti nepožadujeme, aby všechny vrcholy reprezentující proměnné byly pokryté ani nemáme omezení na počet podgrafů pokrývajících graf. Na druhou stranu však máme omezení na velikost druhé partity úplného bipartitního podgrafu v závislosti na velikosti partity první.

Tato obměna zadání je natolik zásadní, že nemůžeme využít většinu přístupů řešících předešlé problémy.

V řešení problému biklikové splnitelnosti budeme vycházet z heuristiky popsané v [26] pro podobný problém, v němž pro bipartitní graf G hledáme množinu $\mathcal{B} = B_1, \dots, B_k$, kde

- (1) každá množina vrcholů B_i obsahuje alespoň jeden vrchol z každé partity G ,
- (2) B_i indukuje úplný podgraf G ,
- (3) každý vrchol z G je obsažen právě v jednom B_i ,
- (4) počet množin B_i , tedy k , není omezený.

5.2 Heuristiky

V této podkapitole popíšeme heuristiky použité pro hledání pokrytí bipartitního grafu omezenými biklikami. Začneme definicí zárodku, což je základní pojem použitý v dále popsanych algoritmech.

Definice. *Nechť F je formule a $I(F)$ je incidenční graf $I(F) = (V_{var}, V_C, E)$. **Zárodek** je neomezená biklika $Z = (V_1, V_2, E)$, kde $E = V_1 \times V_2$, $\emptyset \neq V_1 \subseteq V_{var}$ a $\emptyset \neq V_2 \subseteq V_C$.*

Můžeme též říci, že zárodek je množina klauzulí takových, že všechny klauzule v zárodku obsahují tytéž proměnné.

V této části si popíšeme heuristiku, která vychází z heuristiky na pokrytí bipartitního grafu úplnými bipartitními grafy [26]. Vzhledem k tomu, že se budeme zabývat biklikovým pokrytím, budeme uvažovat bipartitní graf $G = I(F)$ pouze jako bipartitní graf G , nikoliv jako incidenční graf nějaké formule F . Značení partit pro celý graf G budeme používat $V_1(G) = V_{var}$ a $V_2(G) = V_C$.

Při odhadech časové složitosti budeme velikost první V_{var} partity grafu G značit n , velikost druhé partity V_C grafu G označíme m a e je počet hran grafu G , tedy velikost množiny $|E(G)|$.

Nejdříve si popíšeme funkce, které v heuristice použijeme a poté si popíšeme i samotnou heuristiku, která je založena na hladovém přístupu.

Funkce 5.2.1: rozšiřZárodek

Vstup: zárodek Z , rozšiřující proměnná x_j

Výstup: zárodek Z' , což je zárodek Z rozšířený o proměnnou x_j , velikost druhé partity nového zárodku Z'

`incidenceArray[1, ..., m] ← [0, ..., 0];`

`$Z' = (V_1(Z) \cup \{x_j\}, \emptyset, \emptyset)$;`

forall the $C_i \in V_2(Z)$ do

 | `incidenceArray[C_i] ← 1;`

end

forall the $C_i \in N_G(x_j)$ do

 | **if** `incidenceArray[C_i] = 1` **then**

 | `$V_2(Z') \leftarrow V_2(Z') \cup \{C_i\}$;`

 | **end**

end

`$E(Z') \leftarrow V_1(Z') \times V_2(Z')$;`

return `$Z', |V_2(Z')|$`

Funkce rozšiřZárodek vytvoří zárodek Z' , který vznikne ze zárodku Z tak, že přidáme vrchol $x \in V_{var}$ do první partity a druhá partita se bude skládat

z vrcholů $V_2(Z') = V_2(Z) \cap N_G(x)$. Navíc tato funkce vrátí i velikost druhé partity nově vytvořeného zárodku. Toto bude použito pro zjištění proměnné, která maximalizuje velikost druhé partity zárodku při jeho rozšiřování.

Nejdříve projdeme všechny vrcholy druhé partity a poté všechny sousedy vrcholu x_i , kterých je maximálně tolik, kolik je vrcholů v druhé partitě. Časová složitost tohoto kroku je tedy $\mathcal{O}(m)$.

Tuto funkci lze zavolat s již zinicizovaným polem `poleIncidence`. Časová složitost jednoho volání se tím nezlepší, ale tohoto můžeme využít u volání dalších funkcí.

Funkce 5.2.2: zvolRozšiřujícíVrchol

Vstup: zárodek Z

Výstup: proměnná x_i , která nejméně omezí velikost druhé partity zárodku Z

```

count(Z,x) ← 0, maxi ← -1;
forall the  $x_i \in V_{var} \setminus V_1(Z)$  do
    |  $Z', count_{(Z,x_i)} \leftarrow rozšiřZárodek(Z, x_i);$ 
    | if  $count_{(Z,x)} < count_{(Z,x_i)}$  then
    |     |  $max_i \leftarrow x_i;$ 
    |     |  $count_{(Z,x)} \leftarrow count_{(Z,x_i)};$ 
    | end
end
return  $x_i$ 

```

Funkce `zvolRozšiřujícíVrchol` najde vrchol $x_i \in V_{var}(G) \setminus V_1(Z)$, jehož přidáním do zárodku Z nejméně omezíme velikost druhé partity nového zárodku.

Funkci `rozšiřZárodek` zavoláme pro každý vrchol x_i , který není obsažen v zárodku Z , abychom zjistili, kolik vrcholů bude mít druhá partita nového zárodku. Vrchol x_i , který bude mít tuto hodnotu nejvyšší, je návratovou hodnotou funkce.

Pole incidence si pro funkci `rozšiřZárodek` můžeme zinicizovat na začátku této funkce, což nám zabere $\mathcal{O}(m) \subseteq \mathcal{O}(e)$ v těle cyklu, pak budeme už jen využívat tohoto pole bez úprav. Každé zavolání funkce `rozšiřZárodek` projde sousedy pro každý vrchol $x_i \in V_{var}$, všemi voláními funkce `rozšiřZárodek` projdeme všechny hrany grafu, časová složitost pro vrchol tedy bude $\mathcal{O}(e)$. Časová složitost této funkce bude tedy $\mathcal{O}(e)$.

Funkce 5.2.3: vygenerujZárodky

Vstup: bipartitní graf G

Výstup: seznam zárodků $\mathcal{Z} = (Z_{ij} | i, j \in \{1, \dots, n\})$

```

forall the  $x_i \in V_{var}$  do
    | forall the  $x_j \in V_{var}(G), i \leq j$  do
    |     |  $Z_{ij} \leftarrow (\{x_i, x_j\}, N_G(x_i) \cap N_G(x_j), \{x_i, x_j\} \times (N_G(x_i) \cap N_G(x_j)));$ 
    |     |  $\mathcal{Z} \leftarrow (\mathcal{Z}, Z_{ij});$ 
    | end
end
return  $\mathcal{Z}$ 

```

Funkce `vygenerujZárodky` vygeneruje pro každou dvojici vrcholů x_i a x_j z první partity maximální bipartitní podgraf grafu G . Generovat bychom mohli i zárodky, jež mají v první partitě tři a více vrcholů, ale každým zvětšením první partity generovaných zárodků by nám stoupl stupeň polynomu časové složitosti celé heuristiky 5.2.1. Tyto zárodky budeme generovat podobně, jako jsme rozšiřovali zárodky ve funkci 5.2.1. Pro x_i vytvoříme `poleIncidence`, což lze provést v m krocích a pak pro každý vrchol $x_j \in V_{var}$ zjistíme, kolik má společných sousedů s vrcholem x_i pomocí `poleIncidence`, projdeme tedy všechny hrany grafu $E(G)$. Tento proces provedeme pro všechny vrcholy $x_i \in V_{var}$, kterých je n . Celková časová složitost generování je tedy $\mathcal{O}(n(m + e))$.

Funkce `zvolZárodek` vybere ze seznamu zárodků \mathcal{Z} zárodek, podle jedné ze tří strategií s , které označíme S_{min} , S_{rand} nebo S_{max} .

- (1) Strategie S_{min} vybere zárodek Z , jehož druhá partita má minimální velikost $|V_2(Z)|$,
- (2) strategie S_{rand} vybere náhodný zárodek,
- (3) strategie S_{max} vybere zárodek Z , který má maximální velikost druhé partity $|V_2(Z)|$.

Domněnka 5.2.1. *První strategie vede k výběru menších bipartitních grafů, které tedy nepůjdou moc rozšířit. Tento přístup je vhodný, pokud velikost omezené bikliky omezíme malým k . Pokud zvolíme omezení k na velikost bikliky větší, je lepší zvolit poslední možnost, která nejspíše povede k tomu, že budeme mít graf pokryt většími biklikami.*

Tuto domněnku ověříme v sekci 6.3.

Pro volbu náhodného zárodku, tedy strategii S_{rand} stačí vybrat zárodek na náhodně vybrané pozici v seznamu zárodků \mathcal{Z} . Pokud si v každém zárodku uložíme velikost druhé partity do proměnné, pak pro strategie (1) a 3 funkce (`zvolZárodek`) projde všechny zárodky a u každého z nich zkontroluje velikost, což lze udělat v konstantním čase. Časová složitost funkce `zvolZárodek` je tedy $\mathcal{O}(n^2)$.

Funkce `odstraňBikliku(G, Z)` vytvoří nový graf G' , který vznikne z grafu G tak, že odstraníme všechny vrcholy $V_1(Z) \cup V_2(Z)$ a všechny hrany incidentní vrcholům $V_1(Z) \cup V_2(Z)$. Návrátová hodnota funkce je nový graf G' .

Pokud G' obsahuje izolovaný vrchol $C_i \in V_2(G')$, heuristika skončí neúspěchem.

Pro každý vrchol musíme projít seznam sousedů a pokud je soused v zárodku Z , pak ho ze seznamu odstraní. Jestli soused leží v zárodku, lze zjistit obdobně jako u funkce 5.2.1. Pro obě partity si vytvoříme `poleIncidenceVvar` a `poleIncidenceVC`, jejichž inicializace bude mít složitost $\mathcal{O}(n + m)$. Pak už jen projdeme všechny hrany, tedy seznamy sousedů $N_G(x)$ každého vrcholu x a každý vrchol $y \in N_G(x)$, pro který platí $y \in V_1(Z) \cup V_2(Z)$, odstraníme ze seznamu sousedů. Tímto tedy projdeme všechny hrany a každou nejvíce dvakrát. Časová složitost odebrání hran je tedy $\mathcal{O}(e)$. Díky tomu, že velikost první partity grafu G je menší než velikost druhé partity a každý vrchol druhé partity má alespoň jednoho souseda, je celková časová složitost této funkce $\mathcal{O}(n + m + e) = \mathcal{O}(e)$.

Funkce 5.2.4: jednotkováGPropagace

Vstup: bipartitní graf G

Výstup: bipartitní graf G , kde pro všechny vrcholy $C_i \in V_C(G)$ platí
 $|N_G(V_C)| > 2$

fronta $\leftarrow \emptyset$;

forall the $C_i \in V_C(G)$ **do**

if $|N_G(C_i)| = 1$ **then**
 fronta \leftarrow (fronta , C_i);
 end

end

while |fronta| > 0 **do**

$C_i \leftarrow$ fronta.pop;
 $x_j \leftarrow N_G(C_i)$;
 $V_{var}(G) \leftarrow V_{var}(G) \setminus \{x_j\}$;
 $V_C(G) \leftarrow V_C(G) \setminus \{C_i\}$;
 forall the $C_k \in N_G(x_j)$ **do**
 $N_G(C_k) \leftarrow N_G(C_k) \setminus \{x_j\}$;
 if $|N_G(C_k)| = 0$ **then**
 heuristika neuspěla;
 end
 if $|N_G(C_k)| = 1$ **then**
 fronta \leftarrow (fronta , C_k);
 end

end

end

return G

Funkce `jednotkováGPropagace(G)` je funkce, která z bipartitního grafu G odstraní všechny vrcholy $C_i \in V_C$, které mají právě jednoho souseda x_j , protože ty musí být pokryty právě hranou (x_j, C_i) .

Hranu (C_i, x_j) přidáme do biklikového pokrytí a odstraníme ji z grafu společně se všemi hranami incidentními vrcholu x_j . Pokud touto modifikací vznikne izolovaný vrchol $C_k \in V_C(G)$, heuristika selže. Pokud vznikne vrchol $C_k \in V_C(G)$ s právě jedním sousedem, pak tento vrchol musíme též zpracovat.

Tato funkce zpracuje každý vrchol $C_i \in V_C(G)$ maximálně n -krát, protože může mít maximálně n sousedů, které můžeme odebrat a celkem můžeme odebrat $\mathcal{O}(m)$ vrcholů $C_i \in V_C(G)$. Výsledná časová složitost této funkce bude tedy $\mathcal{O}(nm)$.

Tuto funkci by bylo možné provést v lineárním čase [15], ale vzhledem k tomu, že v heuristice používáme časově náročnější funkce, přístup, který je zvolen ve funkci `jednotkováGPropagace`, nám postačuje.

Funkce `vyčistiZárodky` odstraní z jednotlivých zárodků všechny hrany, které byly odebrány z grafu funkcí `odstraňBikliku`. Pokud první partita nemá dvě proměnné nebo velikost druhé parity je menší než tři, pak je zárodek nevalidní a ze seznamu aktuálních zárodků ho odstraníme.

Funkce 5.2.5: vyčistiZárodky

Vstup: seznam zárodků \mathcal{Z} , bipartitní graf G

Výstup: aktuální seznam zárodků \mathcal{Z}'

```
forall the  $Z \in \mathcal{Z}$  do
  forall the  $x \in V_1(Z)$  do
    if  $x \notin V_1(G)$  then
      |  $V_1(Z) \leftarrow V_1(Z) \setminus \{x\}$ ;
    end
  end
  forall the  $C \in V_2(Z)$  do
    if  $C \notin V_2(G)$  then
      |  $V_2(Z) \leftarrow V_2(Z) \setminus \{C\}$ ;
    end
  end
  if  $|V_1(Z)| = 2 \wedge |V_2| \geq 3$  then
    |  $\mathcal{Z}' \leftarrow (\mathcal{Z}', Z)$ ;
  end
end
return  $\mathcal{Z}'$ 
```

Každá hrana může být maximálně v n zárodcích a při každém zavolání této funkce projdeme všechny zárodky a odstraníme hrany $\notin E(G)$. Výsledná časová složitost této funkce je $\mathcal{O}(ne)$.

Funkce `omezZárodek` upraví zárodek Z tak, že odebere vrcholy druhé partity zárodku Z , aby platilo $2^{|V_1(Z)|} \leq |V_2(Z)|$.

K tomu musíme projít všechny vrcholy $|V_2(Z)|$, časová složitost tohoto kroku tedy bude $\mathcal{O}(|V_2(Z)|)$.

Funkce `testMatched` otestuje, jestli lze v bipartitním grafu G nalézt párování velikosti druhé partity $|V_C(G)|$. Nejdříve ale ověří, zdali má smysl tento test provádět. K tomu si stačí uvědomit, že nutná podmínka k tomu, aby existovalo v bipartitním grafu G párování velikosti druhé partity $|V_C(G)|$, musí být mít jeho první partita $V_{var}(G)$ velikost alespoň $|V_C(G)|$, tedy $|V_{var}| \geq |V_C(G)|$.

K nalezení maximálního párování použijeme Hopcroft–Karpův algoritmus [14, 27] na hledání maximálního párování v bipartitních grafech. Pokud nalezneme párování velikosti m , vrátíme `true`. Ve všech ostatních případech vrací `false`.

Pokud vstupem do této funkce je graf, který má druhou partitu $|V_2(G)|$ prázdnou, triviálně vrací `true`.

Pokud máme uložený počet proměnných pro každou klauzuli, můžeme tento proces provést s časovou složitostí $\mathcal{O}(m + e\sqrt{n+m}) \subseteq \mathcal{O}(e\sqrt{n+m})$.

Heuristika nejdříve provede odstranění vrcholů $C_i \in V_2(G)$, které mají stupeň jedna, tedy musí být pokryty párovací hranou. Poté se vygeneruje seznam zárodků \mathcal{Z} a tento seznam se pročistí o všechny zárodky Z_i , které splňují $|V_2(Z)| < 3$.

V hlavním cyklu heuristiky potom zvolíme zárodek Z podle předem určené strategie $s \in \{S_{min}, S_{rand}, S_{max}\}$. Tento zárodek se pokusíme rozšiřovat o vrcholy $x_i \in V_{var}(G) \setminus V_1(Z)$ do té doby, dokud není splněna podmínka pro velikost druhé partity omezené bikliky $2^{|V_1|} \leq |V_2|$ nebo je splněna maximální velikost první partity zárodku Z . Tento cyklus proběhne $\mathcal{O}(k)$, jeho časová složitost bude

Algoritmus 5.2.1: Heuristika na pokrytí grafu.

Vstup: $k \in \{2, \dots, |V_{var}|\}$ - maximální počet proměnných v omezené biklice

Výstup: Nalezeno bilikové pokrytí grafu G

```
 $G \leftarrow \text{jednotkováGPropagace}(G);$  //  $\mathcal{O}(nm)$ 
 $\mathcal{Z} \leftarrow \text{vygenerujZárodky}(G);$  //  $\mathcal{O}(ne)$ 
 $\mathcal{Z} \leftarrow \text{vyčistiZárodky}(\mathcal{Z});$  //  $\mathcal{O}(ne)$ 
while  $|\mathcal{Z}| > 0$  do
   $Z \leftarrow \text{zvolZárodek}(\mathcal{Z}, s);$  //  $\mathcal{O}(n^2)$ 
  while  $|V_1| < k \wedge |2^{V_1}| \leq |V_2|$  do
     $x_i \leftarrow \text{zvolRozšiřujícíVrchol}(Z, V_{var} \setminus V_1);$  //  $\mathcal{O}(e)$ 
     $\text{zárodek} \leftarrow \text{rozšiřZárodek}(Z, x_i);$  //  $\mathcal{O}(m)$ 
  end
   $Z \leftarrow \text{omezZárodek}(Z);$  //  $\mathcal{O}(|V_2(Z)|)$ 
   $G \leftarrow \text{odstraňBikliku}(G, Z);$  //  $\mathcal{O}(e)$ 
   $G \leftarrow \text{jednotkováGPropagace}(G);$  //  $\mathcal{O}(nm)$ 
   $\mathcal{Z} \leftarrow \text{vyčistiZárodky}(\mathcal{Z});$  //  $\mathcal{O}(ne)$ 
  if  $\text{testMatched}(G)$  then //  $\mathcal{O}(e\sqrt{n+m})$ 
    return nalezeno pokrytí
  end
end
return nenalezeno pokrytí
```

tedy $\mathcal{O}(ke)$. Pokud rozšiřování skončí pouze z důvodu maximální velikosti první partity, pak musíme omezit druhou partitu, aby zárodek Z byl omezenou biklikou.

Tuto omezenou bikliku Z a všechny hrany, které jsou jí incidentní, odebereme z grafu G a pokud vznikne izolovaný vrchol $C_f \in V_C(G)$, pak heuristika selže.

Z grafu G odebereme všechny hrany incidentní vrcholům, které mají počet sousedů jedna a pročistíme zárodky, abychom měli pouze validní zárodky.

Hlavní cyklus proběhne nejvýše n -krát, neboť více po dvou disjunktních biklik v grafu nenajdeme. Časová složitost je tedy $\mathcal{O}(\max(\sqrt{m+n}, n)ne)$. Pokud si ale uvědomíme, že funkce `testMatched` bude zavolána jen v případě, že je poměr velikosti druhé partity $V_2(G)$ a první partity $V_1(G)$ grafu G $\frac{m}{n} \leq 1$, je $\sqrt{m+n} \leq \sqrt{2n} \leq n$. Časová složitost heuristiky je tedy $\mathcal{O}(n^2e)$.

5.3 Rozšíření matched formulí s malou deficiencí na bikliky

Nyní bychom chtěli přístup, který uvedl Szeider [34] a který jsme popsali v kapitole 4, zobecnit pro pokrytí biklikami.

Protože nalezení biklik je NP-úplné ([35]), budeme nyní předpokládat, že máme funkci, která je schopna nalézt maximální biklikové pokrytí pro libovolný incidenční graf $I(F)$ formule F . K této funkci budeme přistupovat jako k černé skřínce.

Algoritmus z kapitoly 4 používá redukce na formuli F , které v polynomiálním čase převedou F na formuli s nižší deficiencí, nebo na formuli, která splňuje

předpoklady věty 4.1.3, a je tedy δ^* -kritická. V této sekci ukážeme, že analogie věty 4.1.3 neplatí pro případ biklikové splnitelnosti a že nelze tedy přímo použít tento postup i pro biklikovou splnitelnost.

Uvažme nyní formuli F , která splňuje předpoklady věty 4.1.3, je to tedy formule 2-expandující a nemá čisté ani singulární literály. Uvažme dále libovolnou proměnnou x , která se ve formuli F vyskytuje. Dosadíme-li za x hodnotu $\varepsilon \in \{0, 1\}$, nutně musí být splněny alespoň dvě klauzule F , neboť x není čistý ani singulární literál. Pokud bychom splnili právě dvě klauzule, pak díky tomu, že formule F je 2-expandující, formule $F[x = \varepsilon]$ je určitě 0-expandující a maximální párování formule F se dá převést na maximální párování formule $F[x = \varepsilon]$. Velikost maximálního párování formule $F[x = \varepsilon]$ zůstane stejná, ale počet klauzulí se zmenší, tudíž se zmenší i počet klauzulí nepokrytých párováním a deficience se sníží (Věta 4.1.3).

Uvažme nyní **biklikovou deficienci** δ^B , kterou budeme definovat jako počet klauzulí, které zůstanou nepokryté při maximálním pokrytí omezenými biklikami.

Nyní můžeme zobecnit podmínku δ^* -kritičnosti dvěma způsoby.

- Pokud je formule F δ_1^B -kritická, libovolným ohodnocením $\varepsilon \in \{0, 1\}$ libovolné proměnné x formule F vznikne formule $F[x = \varepsilon]$, která má ostře menší biklikovou deficienci δ^B než formule F ,
- pokud je formule F δ_2^B -kritická, libovolným částečným ohodnocením τ na proměnných z libovolné bikliky B z nějakého maximálního biklikového pokrytí \mathcal{B} vznikne formule $F[\tau]$, která má ostře menší biklikovou deficienci δ^B než formule F .

Oba zmíněné způsoby definice δ^B -kritičnosti zobecňují δ^* -kritičnost v tom smyslu, že pokud uvážíme pokrývání incidenčního grafu pouze hranami, δ_1^B -kritičnost i δ_2^B -kritičnost splynou s pojmem δ^* -kritičnosti.

Mějme formuli F a biklikové pokrytí \mathcal{B} . Pojem alternující cesty vzhledem k párování, můžeme rovněž zobecnit různými způsoby. V obou dále uvedených způsobech zobecnění je **biklikově alternující cesta** p cestou v incidenčním grafu. Budeme rozlišovat následující dva způsoby definice biklikově alternující cesty:

(1) BAC_1

- Každou hranu z cesty p , která není v p poslední a která není pokryta biklikou, následuje hrana z nějaké bikliky B z pokrytí \mathcal{B} a
- každou hranu z cesty p , která není v p poslední a která je pokryta biklikou B z pokrytí \mathcal{B} , následuje hrana nepokrytá žádnou biklikou z pokrytí \mathcal{B} .

(2) BAC_2

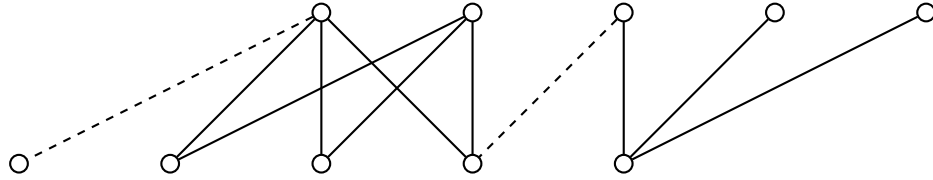
- Každou hranu z cesty p , která není v p poslední a která není pokryta biklikou, následuje hrana z nějaké bikliky B z pokrytí \mathcal{B} a
- je-li q maximální podcesta z p složená jen z hran obsažených v nějakých biklikách B_1, B_2, \dots z \mathcal{B} , pak všechny hrany z q jsou pokryty právě jednou biklikou $B \in \mathcal{B}$ a zároveň je počet hran v q liché.

Pokud vybereme pro definici alternující cesty druhou podmínku BAC_2 , získáme obecnější definici, která zahrnuje i podmínku první. Opačná inkluze však neplatí. V této sekci budeme uvažovat obě varianty definice.

Na základě této definice můžeme zobecnit množinu $R_{G,M}$ pro případ pokrytí omezenými biklikami následujícím způsobem.

Definice. *Nechť F je formule, $G = I(F)$ její incidenční graf a \mathcal{B} pokrytí omezenými biklikami. Pak množina $\mathbf{R}_{G,\mathcal{B}_1}$ ($\mathbf{R}_{G,\mathcal{B}_2}$) je množina vrcholů grafu $I(F)$ takových, že buď je jich možno dosáhnout z vrcholů grafu $I(F)$ nepokrytých biklikovým pokrytím \mathcal{B} biklikově alternující cestou BAC_1 (BAC_2), nebo jsou incidentní biklice, která obsahuje takový vrchol.*

Takto definovaná množina nemusí indukovat ani 0-expandující (obrázek 5.3.1), což znamená, že pro taková redukce není vhodná ani pro původní přístup.



Obrázek 5.3.1: $\mathbf{R}_{G,\mathcal{B}_1}$ i $\mathbf{R}_{G,\mathcal{B}_2}$ množina, která není 1-expandující.

Problém této definice je v tom, že bikliky mají horní omezení na velikost druhé partity, nikoliv však dolní. Zároveň je zjevné, že pokud pro libovolnou bikliku $B_i \in \mathcal{B}$, pro kterou platí, že $|V_{var}(B_i)| \geq |V_C(B_i)|$, můžeme ji nahradit právě $|V_C(B_i)|$ párovacími hranami B_{i_1}, \dots, B_{i_m} , kde $m = |V_C(B_i)|$.

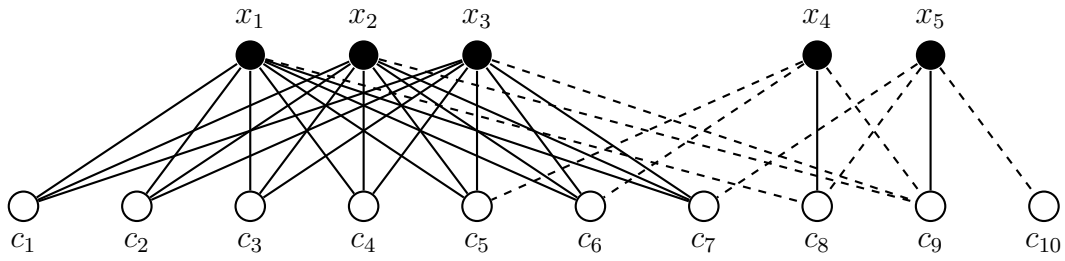
Pokud tedy povolíme jen pokrytí \mathcal{B} , které indukuje 0-expandující graf, pak je množina $R_{G,\mathcal{B}}$ alespoň 0-expandující. Tuto redukci jsme vyzkoušeli u náhodných formulí i u formulí ze SAT competition [1], ale redukce u náhodných dat provedla stejné zmenšení formule jako redukce pomocí $R_{G,M}$ množiny a na formulích ze SAT competition neproběhla vůbec.

V další části algoritmu z kapitoly 4 využíváme toho, že 2-expandující formule bez čistých a jednotkových literálů je δ^* -kritická (věta 4.1.3). Díky tomu můžeme použít rozvětvení v DLL proceduře a takovýmto rozvětvením dosáhneme jen omezené hloubky stromu. Navíc můžeme dosadit za libovolnou proměnnou. Při přímém převodu na pokrytí grafu biklikami bychom nemohli zaručit snížení biklikové deficience δ^B .

Nyní si ukážeme protipříklad k přímému použití větu 4.1.3 v tom znění, jakém je, ale s δ^B -kritičností místo δ^* kritičností.

Příklad 5.3.1. $F_1 = (x_1 \vee x_2 \vee x_3) \wedge (\bar{x}_1 \vee x_2 \vee x_3) \wedge (x_1 \vee \bar{x}_2 \vee x_3) \wedge (x_1 \vee x_2 \vee \bar{x}_3) \wedge (\bar{x}_1 \vee \bar{x}_2 \vee x_3 \vee x_4) \wedge (\bar{x}_1 \vee x_2 \vee \bar{x}_3 \vee \bar{x}_4) \wedge (x_1 \vee \bar{x}_2 \vee \bar{x}_3 \vee \bar{x}_5) \wedge (x_1 \vee x_4 \vee x_5) \wedge (x_2 \vee x_3 \vee \bar{x}_4 \vee \bar{x}_5) \vee (x_5)$

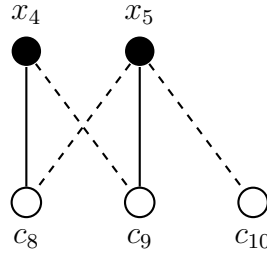
Formule F_1 (obrázek 5.3.2) nemá čisté ani singulární literály, je 2-expandující, její bikliková deficience δ^B je rovná jedné a navíc všechny vrcholy jejího incidenčního grafu je $R_{G,\mathcal{B}}$ množiny, pokud uvážíme první definici alternující cesty BAC_1 i BAC_2 . Pokud bychom větu 4.1.3 chtěli rozšířit na bikliky, pak by musela formule F_1 být δ_1^B -kritická, tedy dosazením za libovolnou proměnnou by se bikliková deficience δ^B měla snížit, nebo δ_2^B -kritická, tedy libovolným ohodnocením proměnných libovolné bikliky z nějakého maximálního biklikového pokrytí by měla vzniknout formule s nižší biklikovou deficiencí.



Obrázek 5.3.2: Incidenční graf formule F_1 .

Dosažením za x_1 hodnotu 0 splníme tři klauzule c_2, c_5 a c_6 a deficiencie se nám nejenže nesníží, ale vzroste o jedna, formule tedy nemůže být δ_1^B -kritická.

Zvolením bikliky B_1 , jejíž první partita je $\{x_1, x_2, x_3\}$, z formule F_1 a částečného ohodnocení, které splňuje bikliku, získáme formuli $F_1[x_1 = 0, x_2 = 0, x_3 = 0] = (x_4 \vee x_5) \wedge (\overline{x_4} \vee \overline{x_5}) \vee (x_5)$. Můžeme si povšimnout, že ohodnocení proměnných bikliky B_1 , pro které platí $\tau(x_1) = 0$, $\tau(x_2) = 0$ a $\tau(x_3) = 0$, je jediné, které tuto bikliku splňuje. Formule $F_1[x_1 = 0, x_2 = 0, x_3 = 0] = (x_4 \vee x_5) \wedge (\overline{x_4} \vee \overline{x_5}) \vee (x_5)$, jejíž incidenční graf je na obrázku 5.3.3, by měla mít biklikovou deficienci δ^B ostře menší než formule F_1 , má ji však stejnou.



Obrázek 5.3.3: Incidenční graf formule $F_1[x_1 = 0, x_2 = 0, x_3 = 0]$.

Formule F_1 nemůže být δ_1^B -kritická ani δ_2^B -kritická. Větu 4.1.3 nelze tedy přímo zobecnit na biklikové pokrytí.

6. Experimenty

Byla provedena řada experimentů na náhodných bipartitních grafech. O tom, jestli je daná formule F matched nebo biklikově splnitelná, rozhoduje totiž jen struktura jejího incidenčního grafu $I(F)$, na druhou stranu ke každému bipartitnímu grafu G najdeme formuli F , pro kterou platí, že $I(F) = G$ (připustíme-li, že se mohou opakovat klauzule). Při biklikové splnitelnosti nezáleží na polaritě jednotlivých literálů. Uvážíme-li omezenou bikliku B z biklikového pokrytí \mathcal{B} incidenčního grafu $I(F)$ formule F , pak díky podmínce omezující velikost partity $V_2(B)$ vzhledem k velikosti $V_1(B)$ platí, že bez ohledu na to, jaké mají proměnné z $V_1(B)$ polarity v klauzulích v $V_2(B)$, najdeme ohodnocení proměnných z $V_1(B)$, které splňuje všechny klauzule z $V_2(B)$. Z tohoto důvodu jsou experimenty prováděny na náhodných bipartitních grafech, ačkoli v popisu výsledků se budeme odkazovat i na formule, jejichž incidenční grafy uvažujeme.

Specifikace hardware a software, který byl použit pro vyhotovení a vyhodnocení experimentů a celkový strojový čas, který zabralo jejich vyhodnocení, je popsán v příloze 8.2. Návod, jak lze experimenty replikovat, je uveden v příloze 8.1. Všechny kompletní tabulky je možno nalézt na přiloženém CD 8.3.

6.1 Generátor náhodných formulí

Pro experimenty jsme využili dvou různých způsobů pro generování náhodných incidenčních grafů formulí. V obou případech se snažíme omezit počet literálů v klauzuli.

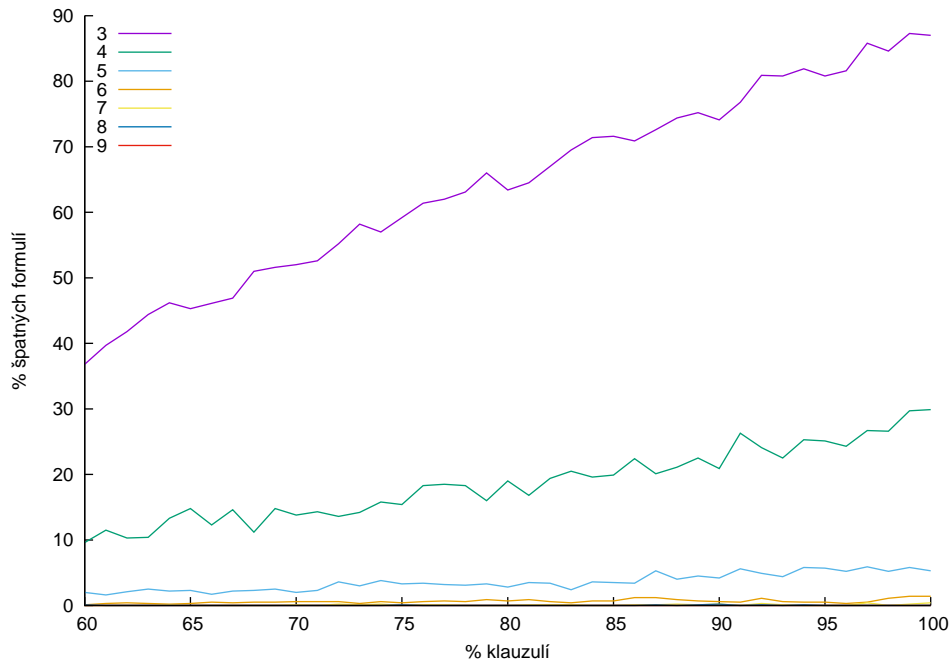
První způsob generování incidenčního grafu $G = I(F) = (V_{var}, V_C, E)$ vede k omezení průměrné délky klauzule odpovídající formule F . Pro dané $k \geq 3$ určíme pravděpodobnost výskytu hrany (x_i, C_j) jako $p = \frac{k}{|V_{var}(G)|}$, s touto pravděpodobností vložíme hranu do grafu. Tímto generováním nám vznikají prázdné klauzule a proměnné, které nemají žádný výskyt. V incidenčním grafu $I(F)$ jsou to izolované vrcholy první i druhé partity. Takto izolované vrcholy nebudeme brát v potaz. U experimentů zkoumáme vlastnosti incidenčních grafů na základě poměru druhé partity a první partity, tento poměr budeme u náhodných grafů počítat až po generování.

Při tomto přístupu se však často stává, že již funkce 5.2.4 (jednotkováGPropagace) vytvoří v grafu izolované vrcholy a již v tomto kroku tedy zjistíme, že ve vygenerovaném grafu neexistuje ani párování pokrývající všechny klauzule, formule tedy není splnitelná, tím spíše není matched ani biklikově splnitelná.

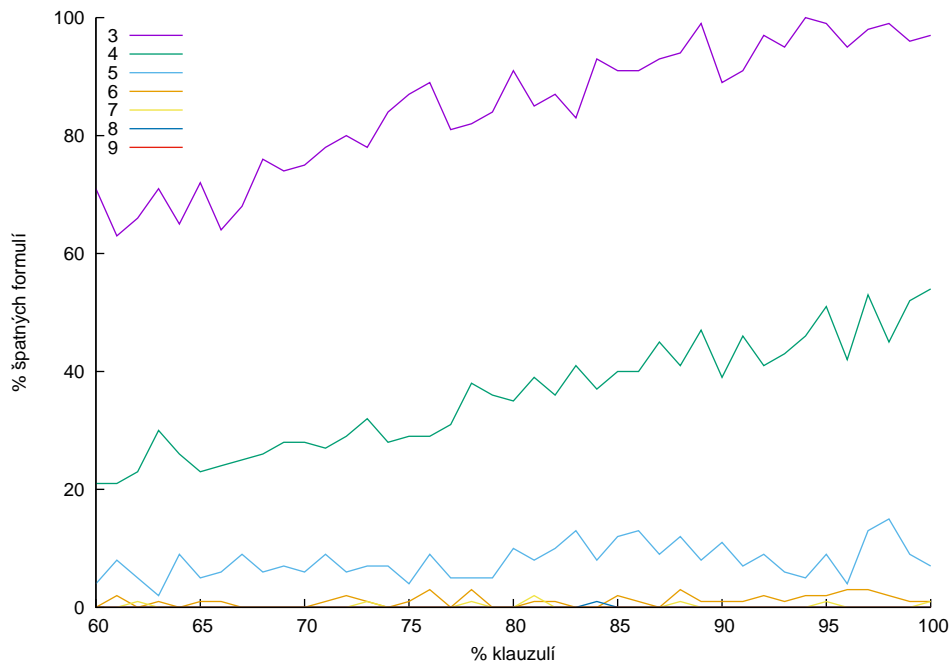
V prvním typu experimentů, které si popíšeme, chceme právě ověřit, jak často k této situaci dochází. Výsledky těchto experimentů popisujeme dále v této podkapitole.

Provedli jsme čtyři různé experimenty, kde ve všech experimentech zkoumáme procento grafů, pro které funkce 5.2.4 neuspěje. Experimenty se liší v počtu vrcholů druhé partity $|V_C|$ a v tom, kolik testů jsme provedli.

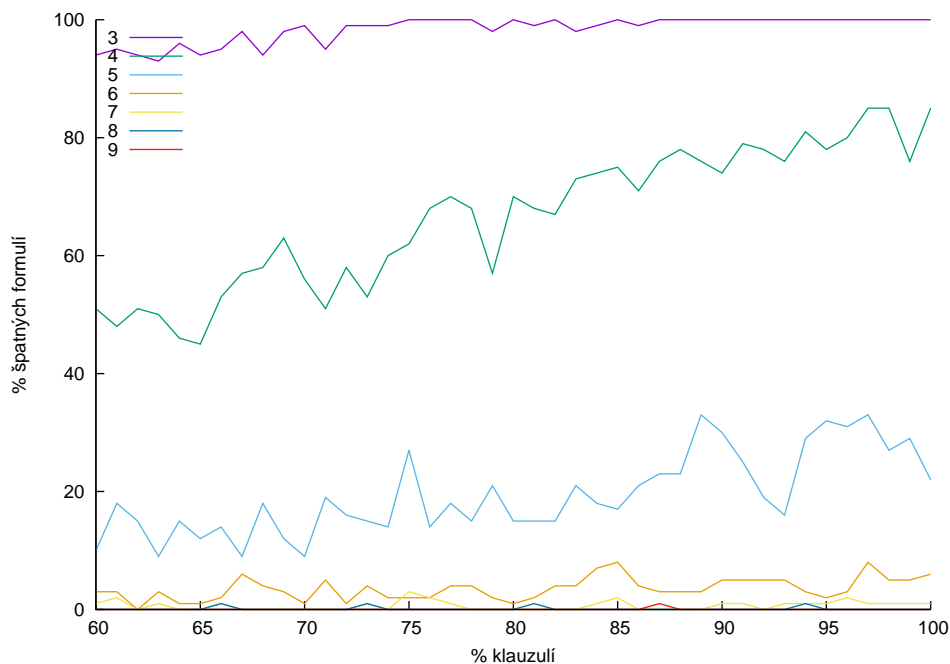
Naopak všechny experimenty zkoumají, jak se chovají grafy, jejichž vrcholy z druhé partity $C \in V_C$ mají průměrný stupeň $k \in \{3, \dots, 9\}$ a poměr velikosti první partity V_{var} ku druhé partitě V_C je 0.6 až 1.



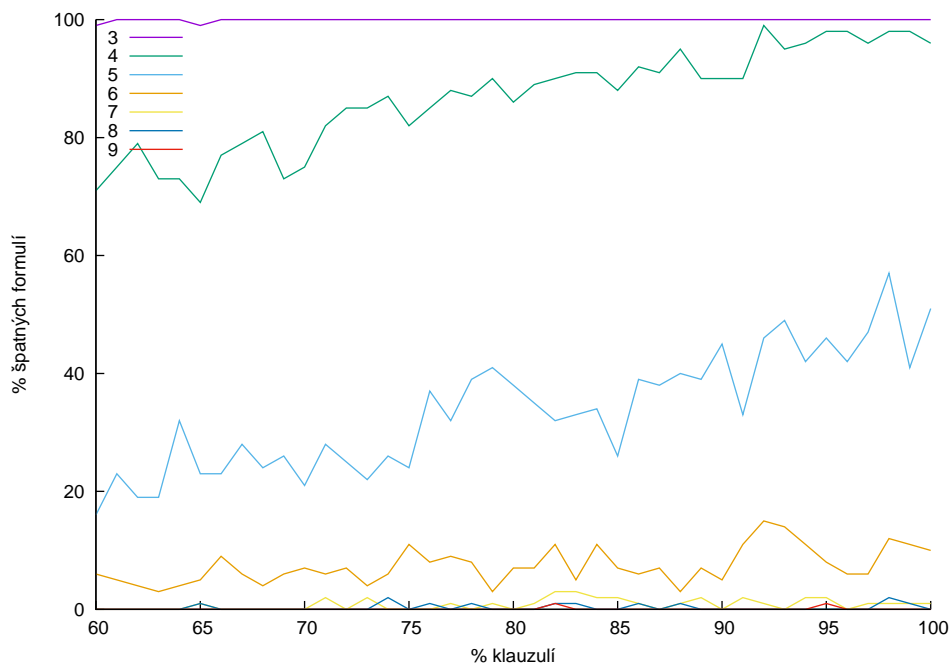
Obrázek 6.1.1: Tisíc náhodných KNF formulí s $|V_{var}| = 100$, průměrně k proměnných v klauzuli, poměrem $r = \frac{|V_C|}{|V_{var}|} 100$, pro každé $3 \leq k \leq 9$ a každý poměr $r \in \{60\%, 61\%, \dots, 100\%\}$.



Obrázek 6.1.2: Sto náhodných KNF formulí s $|V_{var}| = 200$, průměrně k proměnných v klauzuli, poměrem $r = \frac{|V_C|}{|V_{var}|} 100$, pro každé $3 \leq k \leq 9$ a každý poměr $r \in \{60\%, 61\%, \dots, 100\%\}$.



Obrázek 6.1.3: Sto náhodných KNF formulí s $|V_{var}| = 500$, průměrně k proměnných v klauzuli, poměrem $r = \frac{|V_C|}{|V_{var}|}100$, pro každé $3 \leq k \leq 9$ a každý poměr $r \in \{60\%, 61\%, \dots, 100\%\}$.



Obrázek 6.1.4: Sto náhodných KNF formulí s $|V_{var}| = 1000$, průměrně k proměnných v klauzuli, poměrem $r = \frac{|V_C|}{|V_{var}|}100$, pro každé $3 \leq k \leq 9$ a každý poměr $r \in \{60\%, 61\%, \dots, 100\%\}$.

Výsledky experimentů jsou zachyceny na obrázcích 6.1.1–6.1.4.

- 6.1.1 s grafy G pro které platí $|V_{var}| = 100$, a kde jsme provedli 1000 testů,
- 6.1.2 s grafy G pro které platí $|V_{var}| = 200$, a kde jsme provedli 100 testů,
- 6.1.3 s grafy G pro které platí $|V_{var}| = 500$, a kde jsme provedli 100 testů,
- 6.1.4 s grafy G pro které platí $|V_{var}| = 1000$, a kde jsme provedli 100 testů.

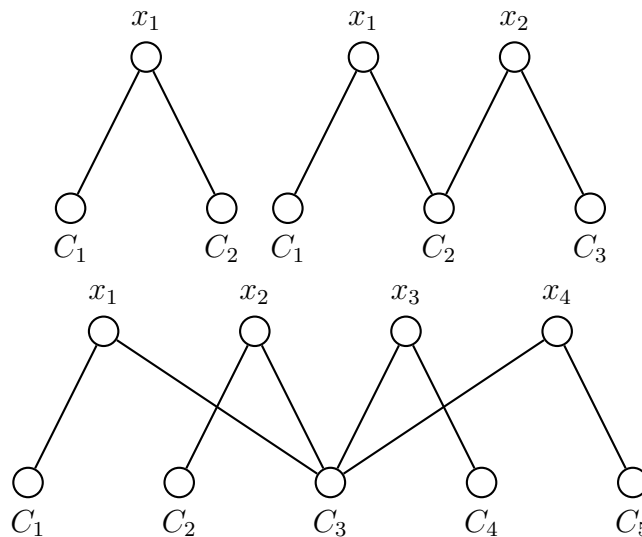
U experimentů z obrázků 6.1.2–6.1.4 jsme provedli jen 100 testů a nikoliv 1000, z důvodu časové náročnosti výpočtů.

Na ose x je v obrázcích 6.1.1–6.1.4 zobrazena proměnná $100 \frac{|V_C|}{|V_{var}|}$, tedy poměr mezi počtem klauzulí a proměnných v procentech. Na ose y je zachyceno, pro jakou část grafů již funkce 5.2.4 zjistí, že není matched, respektive biklikově splnitelná.

Můžeme si povšimnout, že u všech čtyř testů se s rostoucím průměrným stupněm vrcholů z druhé partity snižuje počet grafů, na kterých funkce 5.2.4 (jednotkovaGPropagace) neuspěje. Zároveň ale roste počet grafů, na kterých funkce 5.2.4 (jednotkovaGPropagace) neuspěje, spolu s počtem proměnných ve formuli nezávisle na průměrném stupni vrcholů z druhé partity.

V grafech na obrázcích 6.1.2 u funkcí, které reprezentují výsledky experimentu pro grafy s průměrným stupněm vrcholů druhé partity 3 a 4 a 6.1.4 u funkcí, které reprezentují výsledky experimentu pro grafy s průměrným stupněm vrcholů druhé partity 4 a 5, si můžeme povšimnout, že je jejich průběh velice podobný. Tento jev lze vysledovat i u dalších funkcí z obrázků 6.1.1–6.1.4.

V neúspěšných případech se vyskytují struktury, podobné strukturám na obrázku 6.1.5.



Obrázek 6.1.5: Grafy, na kterých neuspěje funkce 5.2.4.

Při druhém způsobu generování náhodných bipartitních grafů generujeme grafy, v nichž vrcholy druhé partity mají stupeň právě k , takže platí, že pro každý vrchol $C \in V_C$ platí $|N_G(C)| = k$. Pro každou klauzuli zvolíme náhodně k různých proměnných, které v ní budou mít výskyt. Všechny proměnné mají stejnou pravděpodobnost výskytu v dané klauzuli.

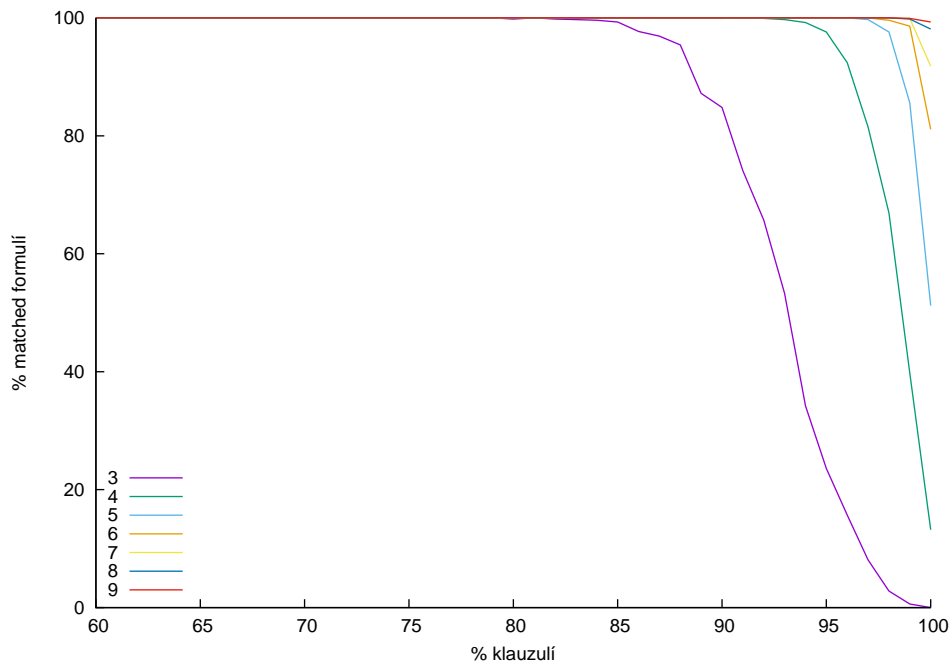
U grafů generovaných druhým způsobem nebude mít funkce 5.2.4 žádný účinek, protože každý vrchol $C \in V_C(G)$ má alespoň tři sousedy.

6.2 Matched formule

V této podkapitole shrneme výsledky experimentů, pomocí nichž jsme ověřili odhad Johna Franca a Allena Van Geldera [21], který jsme zmínili v kapitole 3, sekci 3.1, a který tvrdí, že 3-KNF formule F s n proměnnými a m klauzulemi, kde $m = 0.64n$, jsou téměř jistě matched. Zjistíme, jak moc těsný je tento odhad pro náhodně generované k -KNF formule a pro jaké poměry bude většina k -KNF formulí matched, s $k \geq 3$. Nakonec ještě porovnáme výsledky k -KNF formulí s výsledky formulí, jejichž klauzule mají průměrný počet proměnných k , tedy formulí generovaných prvním způsobem.

Nechť F je k -KNF formule s n proměnnými a m klauzulemi. Označme $r = 100 \frac{m}{n}$ poměr klauzulí a proměnných v procentech.

V tomto experimentu chceme zjistit, jaká část formulí k -KNF je matched a jestli existuje nějaký interval $\langle x_k, y_k \rangle$ s $x_k, y_k \in \mathbb{Q}$ takový, že formule, jejichž poměr počtu klauzulí a počtu proměnných $r < x_k$, jsou ve většině případů matched a formule, jejichž poměr počtu klauzulí a počtu proměnných $r > y_k$, ve většině případů matched nejsou. U formulí, jejichž poměr počtu klauzulí a počtu proměnných r je v intervalu $\langle x_k, y_k \rangle$, tedy $x_k \leq r \leq y_k$, není jednoznačné, zdali formule je nebo není matched. Tomuto intervalu budeme říkat **fázový přechod** a lze ho nalézt kupříkladu u problému splnitelnosti k -KNF formulí [7, 10, 23, 30].



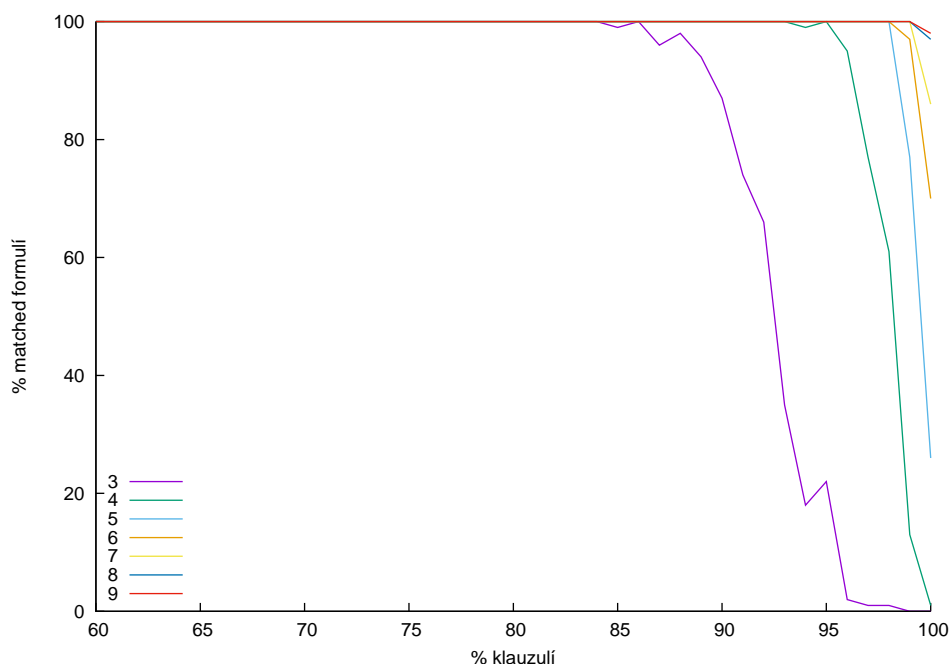
Obrázek 6.2.1: Tisíc náhodných k -KNF formulí s $|V_{var}| = 200$, poměrem $r = \frac{|V_C|}{|V_{var}|} 100$, pro každé $3 \leq k \leq 9$ a každý poměr $r \in \{60\%, 61\%, \dots, 100\%\}$.

Na obrázku 6.2.1 jsou výsledky testů, ve kterých jsme uvažovali formule, které mají sto proměnných, jsou v k -KNF pro $k \in \{3, \dots, 9\}$ a poměr klauzulí a proměnných r se pohybuje mezi 60 a 100 procenty. Pro každou konfiguraci jsme

vygenerovali tisíc náhodných k -regulárních grafů, odpovídajících incidenčním grafům k -KNF formulí.

Pro všechny k -KNF formule s $k \geq 4$ je horní hranice fázového přechodu $y_k = 100\%$, což znamená, že pokud je maximálně tolik klauzulí co proměnných, našli jsme v experimentech vždy alespoň jednu formuli, která je matched. Pokud je více klauzulí než proměnných, formule už triviálně nemůže být matched.

Fázový přechod pro 3-KNF je interval $\langle 80\%, 99\% \rangle$, pro 4-KNF je to interval $\langle 90\%, 100\% \rangle$ a všechny k -KNF formule s $k \geq 5$ mají dolní hranici x_k nad 98%. Fázové přechody jsou zobrazeny v tabulce 6.2.5. Pokud není uvedena horní hranice y_k , pak je její hodnota 100%.



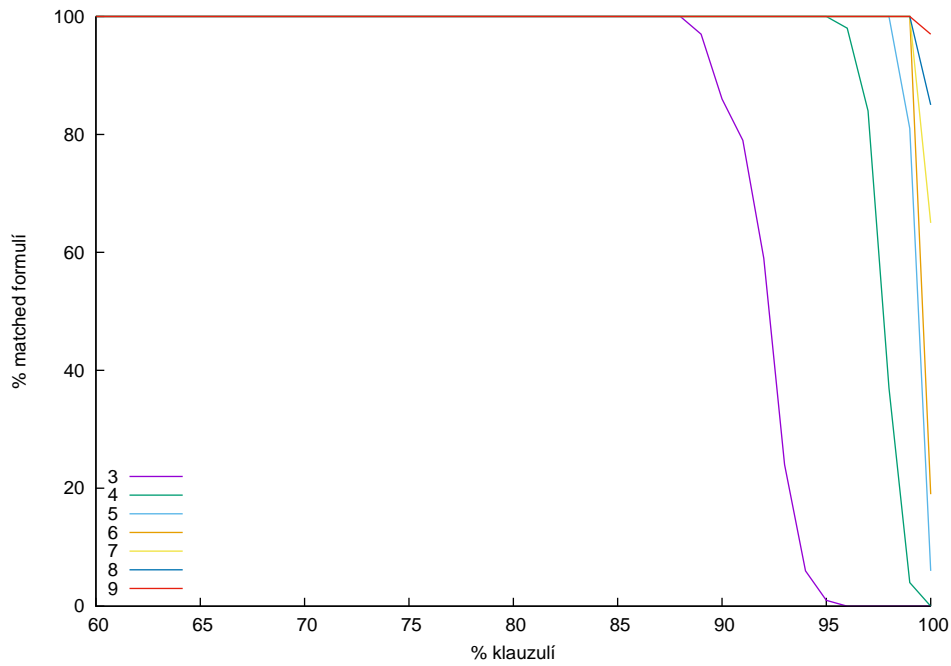
Obrázek 6.2.2: Sto náhodných k -KNF formulí s $|V_{var}| = 200$, poměrem $r = \frac{|V_C|}{|V_{var}|} 100$, pro každé $3 \leq k \leq 9$ a každý poměr $r \in \{60\%, 61\%, \dots, 100\%\}$.

Na obrázku 6.2.2 jsou výsledky testů, ve kterých jsme uvažovali formule, které mají 200 proměnných, jde o k -KNF pro $k \in \{3, \dots, 9\}$ a poměr klauzulí a proměnných r se pohybuje mezi 60 a 100 procenty. Pro každou konfiguraci jsme vygenerovali sto náhodných k -KNF formulí.

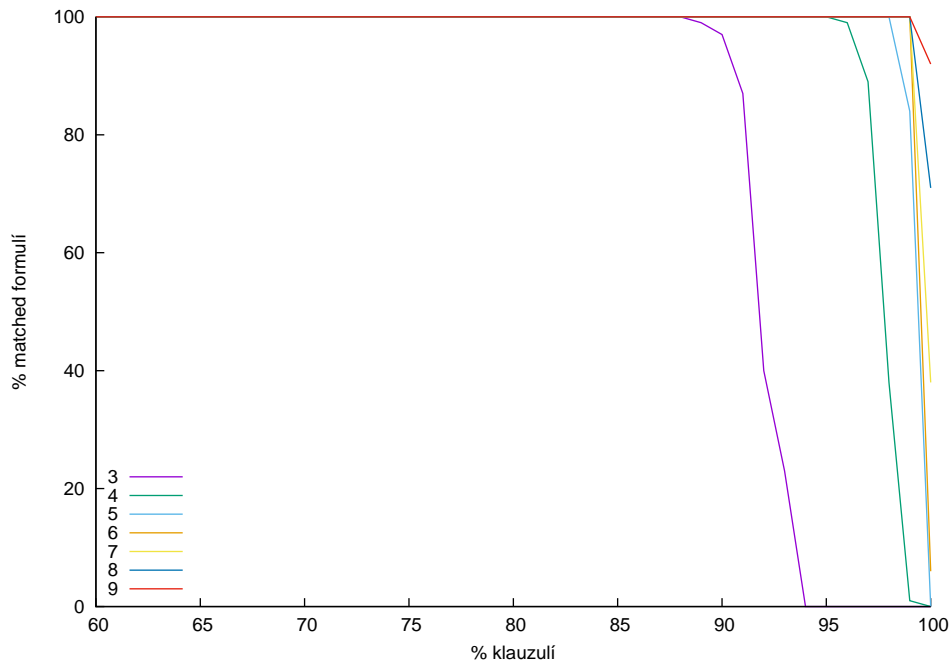
Fázové přechody, jejichž výpis můžeme najít v tabulce 6.2.5, mají střed velice podobný jako formule z experimentu zobrazeném na obrázku 6.2.1, jen se zmenšila velikost intervalu $|y_k - x_k|$.

Na obrázcích 6.2.3 a 6.2.4 jsou zobrazeny výsledky testů, ve kterých jsme uvažovali formule, které mají pět set a tisíc proměnných, jsou k -regulární pro $k \in \{3, \dots, 9\}$ a poměr klauzulí a proměnných r se pohybuje mezi 60 a 100 procenty. Pro každou konfiguraci jsme vygenerovali sto náhodných k -regulárních formulí.

Fázové přechody jsou u obou experimentů téměř stejné a změnu velikosti intervalu a zachování středu fázového přechodu, jež jsme pozorovali porovnáním experimentů na obrázku 6.2.1 a na obrázku 6.2.2. Zmenšení intervalu fázového přechodu můžeme pozorovat i porovnáním experimentů na obrázku 6.2.2 na



Obrázek 6.2.3: Sto náhodných k -KNF formulí s $|V_{var}| = 500$, poměrem $r = \frac{|V_C|}{|V_{var}|} 100$, pro každé $3 \leq k \leq 9$ a každý poměr $r \in \{60\%, 61\%, \dots, 100\%\}$.



Obrázek 6.2.4: Sto náhodných k -KNF formulí s $|V_{var}| = 1000$, poměrem $r = \frac{|V_C|}{|V_{var}|} 100$, pro každé $3 \leq k \leq 9$ a každý poměr $r \in \{60\%, 61\%, \dots, 100\%\}$.

obrázku 6.2.3, a stejně tak porovnáním experimentů na obrázku 6.2.3 na obrázku 6.2.4.

Výsledky těchto čtyř případů, které jsou zobrazeny na obrázcích 6.2.1–6.2.4, jsou shrnuty v tabulce 6.2.5.

$ V_{var} $	3-KNF		4-KNF		5-KNF	k -KNF ($k \geq 6$)
	x_3	y_3	x_4	y_4	x_5	x_k
100	80	99	90	100	97	≥ 98
200	85	98	94	100	99	≥ 99
500	89	95	96	99	99	100
1000	89	93	96	99	99	100

Tabulka 6.2.5: Dolní hranice fázového přechodu matched formulí.

Můžeme si povšimnout, že odhad dokázaný J. Frankem s A. Gelderem [21] je dolní odhad a experimenty nasvědčují tomu, že by hranice poměru mezi počtem proměnných a počtem klauzulí formule, která bude téměř jistě matched, mohl být vyšší.

Střed fázového přechodu matched 3-KNF formulí je 92%. Přestože 3-KNF formule s 1000 proměnnými s poměrem počtu klauzulí k počtu hran větším než 93% nejsou téměř nikdy matched, jsou většinou splnitelné, protože fázový přechod pro splnitelnost 3-KNF formulí jehož dolní hranice se pohybuje mezi 300% a 400% a horní hranice je mezi 400% a 500%. Jako střed tohoto intervalu bývá velice často uváděn poměr počtu klauzulí a proměnných formule 420%. Dolní a horní hranice se liší podle výběru testovaných nebo generovaných dat [7, 10, 23, 30].

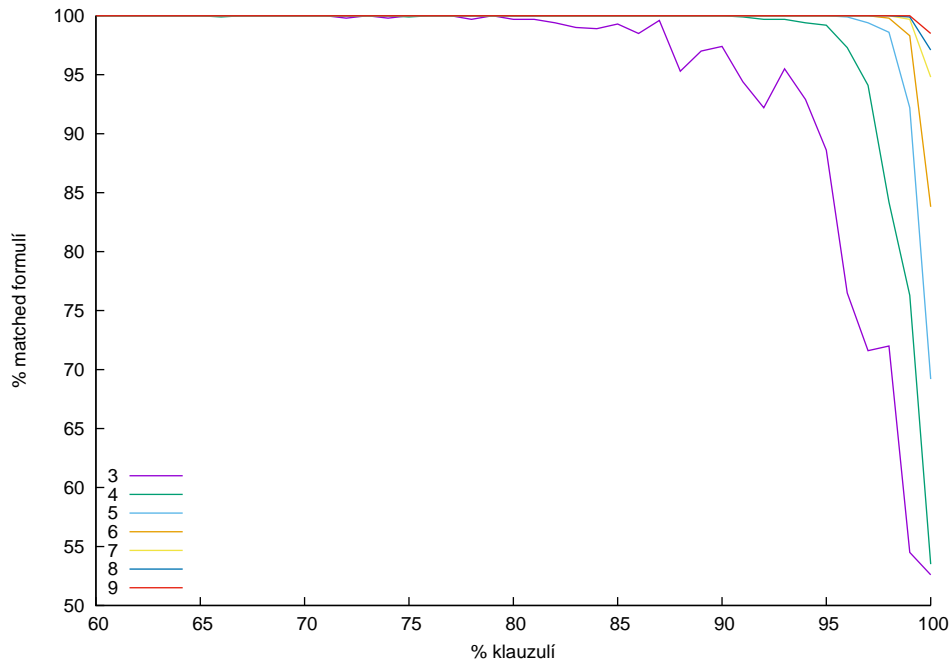
Nyní ještě krátce zhodnotíme experimenty na grafech, které byly generovány prvním způsobem, tedy jejichž klauzule mají průměrný počet proměnných k . U těchto experimentů je vygenerováno vždy sto grafů pro každou konfiguraci, ale všechny grafy, které neuspějí ve funkci 5.2.4 (`jednotkováGPropagace`), odfiltrujeme a zobrazíme pouze poměr mezi formulemi, které jsou matched a těmi, jež neodfiltrovala funkce 5.2.4.

Na obrázcích 6.2.6–6.2.9 jsou výsledky experimentů, u kterých jsme ve všech případech uvažovali formule s průměrným počtem proměnných v klauzuli $k \in \{3, \dots, 9\}$ a s poměrem klauzulí a proměnných $r \in \{60\%, 61\%, \dots, 100\%\}$.

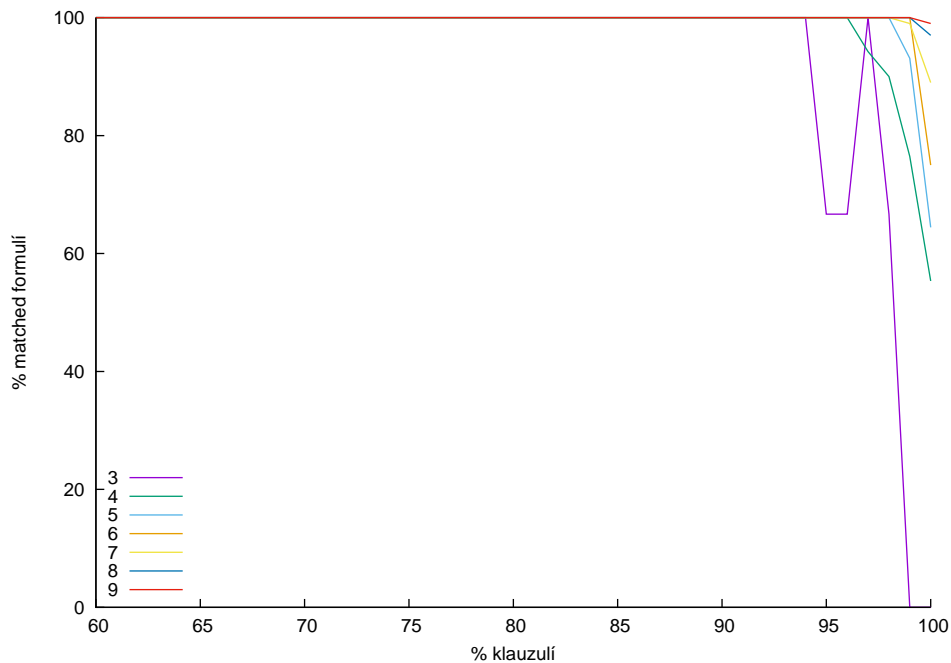
Oproti k -KNF formulím, je u experimentů provedených na formulích, jejichž průměrný počet klauzulí je k , je dolní hranice fázového přechodu o mnoho vyšší a horní hranice je pro všechna k rovna 100%.

Navíc si u experimentů zobrazených na obrázcích 6.2.8 a 6.2.9, u kterých je velikost první partity V_{var} pět set a tisíc, můžeme povšimnout toho, že formule, které mají průměrný počet proměnných $k = 3$, mají velké skoky (na obrázku 6.2.8 se tyto skoky dějí mezi 71% a 83% a na obrázku 6.2.9 je jeden skok v okolí 62%). Pokud všechny incidenční grafy formulí v experimentu pro daný poměr počtu klauzulí a proměnných vyvolaly neúspěch funkce 5.2.4 (`jednotkováGPropagace`), označili jsme, že 0% klauzulí je matched, což ilustrují výše zmíněné skoky ve výsledcích na obrázcích 6.1.1–6.1.4.

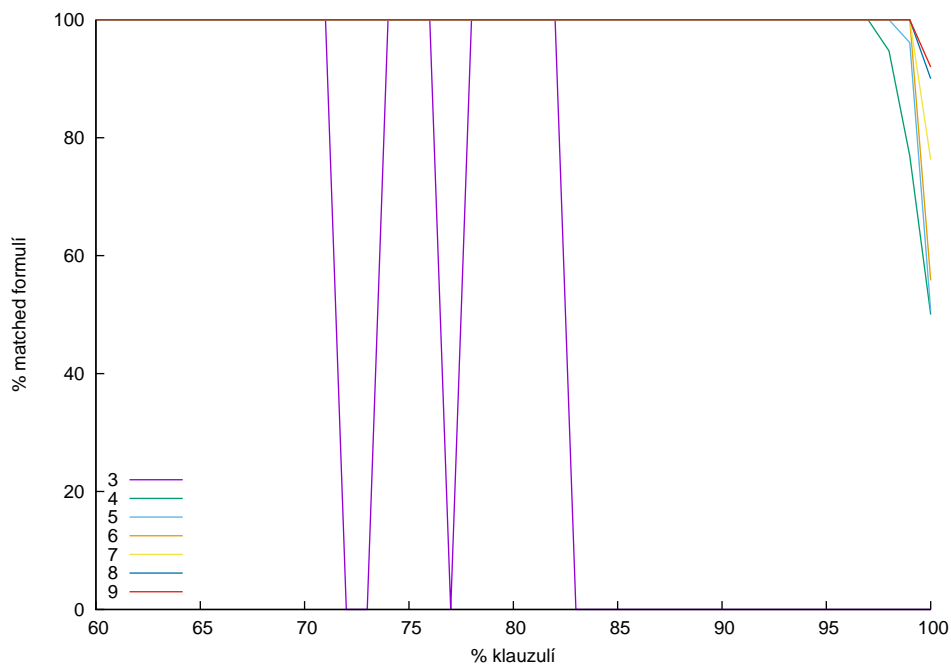
Výsledky experimentů pro druhý způsob generování ukazují, že je příliš velké procento formulí s klauzulemi, jež průměrně obsahují málo proměnných, jsou neúspěšné již ve funkci `jednotkováGPropagace` a ve zbývajících případech jsou formule téměř vždy matched.



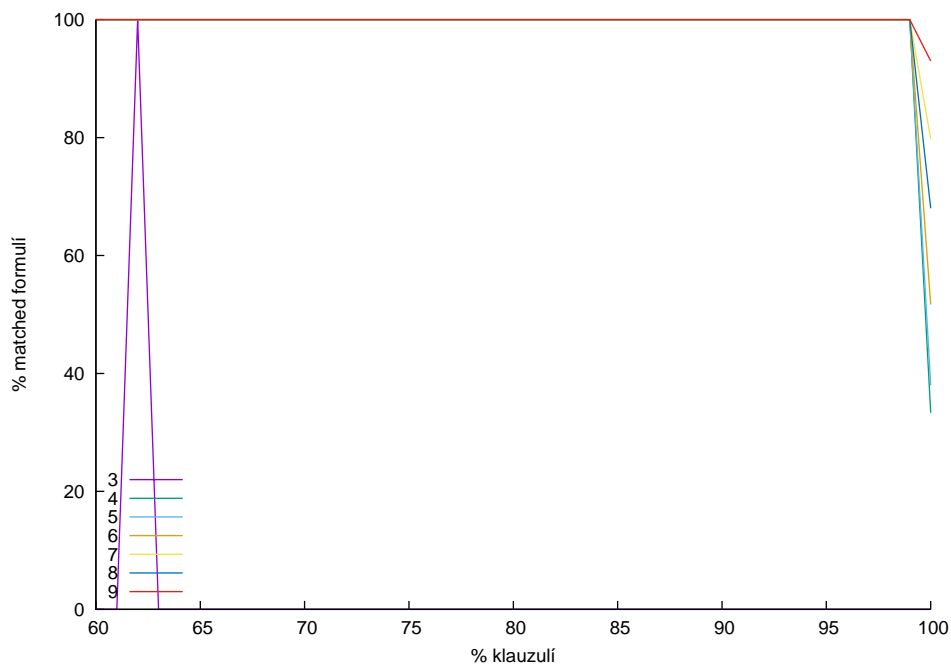
Obrázek 6.2.6: Tisíc náhodných KNF formulí s $|V_{var}| = 100$, průměrně k proměnných v klauzuli, poměrem $r = \frac{|V_C|}{|V_{var}|} 100$, pro každé $3 \leq k \leq 9$ a každý poměr $r \in \{60\%, 61\%, \dots, 100\%\}$.



Obrázek 6.2.7: Sto náhodných KNF formulí s $|V_{var}| = 200$, průměrně k proměnných v klauzuli, poměrem $r = \frac{|V_C|}{|V_{var}|} 100$, pro každé $3 \leq k \leq 9$ a každý poměr $r \in \{60\%, 61\%, \dots, 100\%\}$.



Obrázek 6.2.8: Sto náhodných KNF formulí s $|V_{var}| = 500$, průměrně k proměnných v klauzuli, poměrem $r = \frac{|V_C|}{|V_{var}|}100$, pro každé $3 \leq k \leq 9$ a každý poměr $r \in \{60\%, 61\%, \dots, 100\%\}$.



Obrázek 6.2.9: Sto náhodných KNF formulí s $|V_{var}| = 1000$, průměrně k proměnných v klauzuli, poměrem $r = \frac{|V_C|}{|V_{var}|}100$, pro každé $3 \leq k \leq 9$ a každý poměr $r \in \{60\%, 61\%, \dots, 100\%\}$.

6.3 Heuristika na biklikové pokrytí

V této sekci popíšeme experimenty, ve kterých jsme zjišťovali, jakou část bipartitních grafů lze biklikově pokrýt za použití heuristiky, představené v sekci 5.2. Skutečnou část grafů z celého experimentu, kterou můžeme pokrýt omezenými biklikami jsme neurčovali, vzhledem k tomu, že problém je NP-úplný a k ověření bychom museli zřejmě využít prohledávání hrubou silou, což by bylo mimo výpočetní možnosti vzhledem k počtu provedených experimentů.

Provedli jsme experimenty na bipartitních grafech s několika různými velikostmi první partity. V každém z experimentů provádíme testy pro různé poměry počtu vrcholů druhé partity m bipartitního grafu G k počtu vrcholů první partity n , tedy $r = 100 \frac{m}{n}$ a pro různé stupně vrcholů druhé partity k . Bipartitní grafy pro experiment generujeme druhým způsobem uvedeným v sekci 6.1, to jest, generují se grafy, v nichž vrcholy odpovídají klauzulím, mají přesně stupeň k .

V experimentech budeme sledovat obdobně jako u experimentu v sekci 6.2, zdali existuje nějaký K -fázový přechod. K -fázový přechod je interval $\langle x_r, y_r \rangle$, kde r je poměr mezi počtem vrcholů druhé partity grafu k počtu vrcholů první partity v procentech a x_r, y_r je stupeň vrcholů druhé partity. Zároveň platí, že pro všechny grafy s poměrem partit r a stupněm vrcholů druhé partity grafu ostře menším než x_r heuristika nenalezne biklikové pokrytí téměř nikdy a pro grafy, jejichž poměr partit je r a stupněm vrcholů druhé partity grafu ostře větším než y_r , biklikové pokrytí nalezne téměř vždy.

V experimentech též porovnáme různé strategie S_{min} , S_{rand} , S_{max} , uvedené v sekci 5.2 u funkce zvolZárodek, a různá omezení na velikost první i druhé partity bikliky.

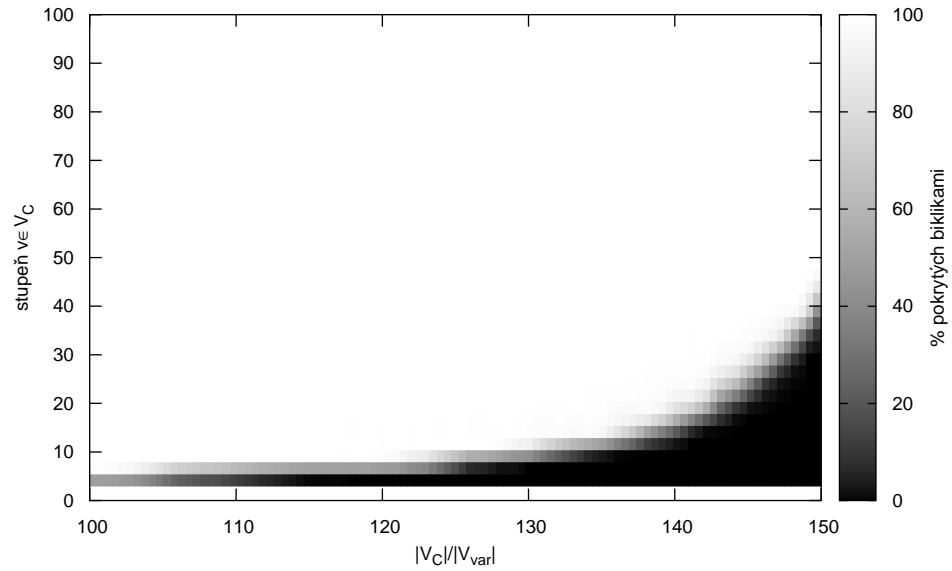
Výsledky experimentů jsou zaznamenány na obrázcích 6.3.1–6.3.11, kde na ose x je poměr počtu vrcholů druhé partity grafu V_C a první partity grafu V_{var} , tedy $r = 100 \frac{|V_C|}{|V_{var}|}$ v procentech. Na ose y je zobrazen stupeň každého vrcholu $C \in V_C$. Stupně šedé zobrazují, pro jakou část z celkového počtu testů heuristika našla biklikové pokrytí, kde černá vyznačuje to, že se pro žádný graf nepodařilo nalézt biklikové pokrytí a bílá vyznačuje to, že se podařilo nalézt pokrytí pro všechny bipartitní grafy.

Na obrázcích 6.3.1–6.3.3 jsou zobrazeny výsledky testů heuristiky, u které jsme měli omezenou velikost první partity každé bikliky dvěma a velikost první partity celého grafu, který jsme se snažili pokrýt, je sto.

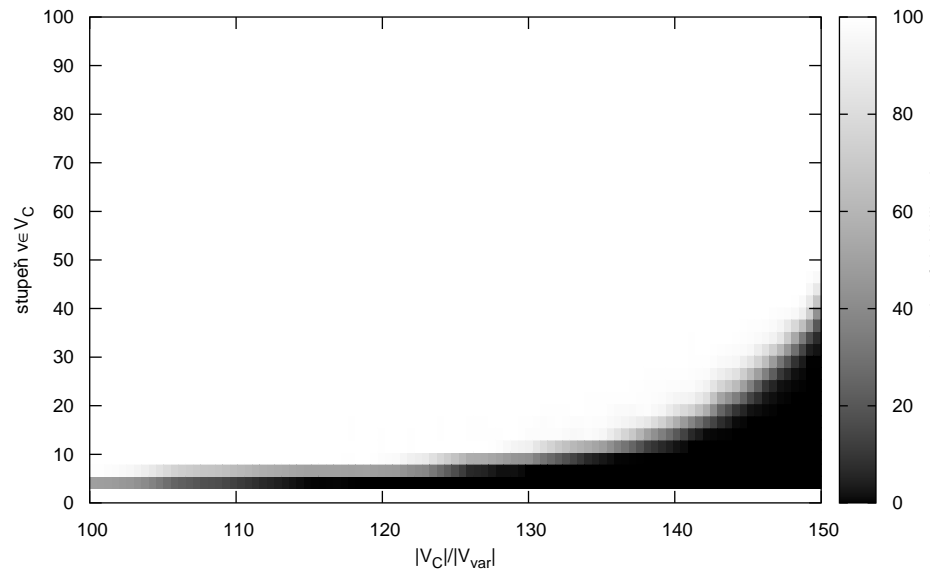
Biklikové pokrytí tohoto experimentu může být složeno pouze z omezených biklik $K_{1,1}$ a $K_{2,3}$. Pro takovéto omezení může existovat biklikové pokrytí jen do poměru partit 150%. Pro poměr 150% se musí biklikové pokrytí skládat právě z biklik $K_{2,3}$. Z toho důvodu jsou všechny experimenty omezeny tímto poměrem.

Strategie S_{min} , S_{rand} a S_{max} použité v tomto experimentu budeme značit $S_s^{2,3}$, $s \in \{min, rand, max\}$ kvůli pozdějšímu porovnání. Průběh u jednotlivých strategií výběru zárodků S_{min} , S_{rand} a S_{max} je velice podobný.

K tomu, aby pro graf G se stem vrcholů v první partitě a s $m \geq 100$ vrcholy v druhé partitě existovalo pokrytí, potřebujeme alespoň $(m - 100)$ $K_{2,3}$ biklik. Zbylé vrcholy můžou být pokryty párovacími hranami. Ve výsledcích na obrázcích 6.3.1–6.3.3 si můžeme povšimnout, že pokud je $m < 130$, nalezne heuristika biklikové pokrytí už pro malé k , kde k je stupeň vrcholů druhé partity grafu.



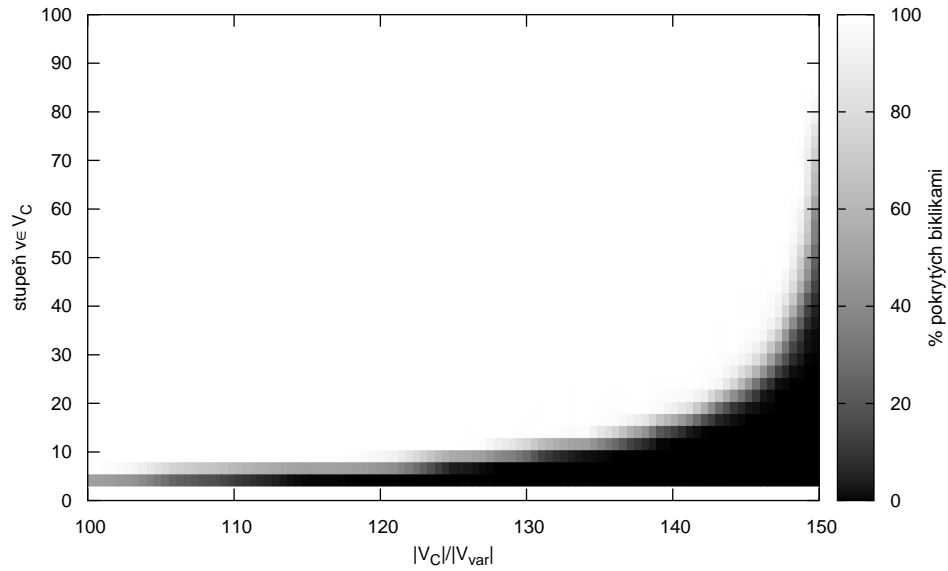
Obrázek 6.3.1: $|V_{var}| = 100$, $s = S_{min}$, pro každou omezenou bikliku $B \in \mathcal{B}$ platí $|V_1(B)| \leq 2$.



Obrázek 6.3.2: $|V_{var}| = 100$, $s = S_{rand}$, pro každou omezenou bikliku $B \in \mathcal{B}$ platí $|V_1(B)| \leq 2$.

Z toho můžeme usuzovat, že graf obsahuje poměrně mnoho biklik $K_{2,3}$, které pokrývají značnou část druhé partity.

Všechny strategie se v experimentu ukázaly srovnatelné pro menší poměry. Můžeme si povšimnout, že pokud se blížíme k poměru 150%, zvětšuje se rozdíl mezi heuristikami S_{max} a S_{min} , zatímco S_{min} a S_{rand} zůstávají stejné. Toho si



Obrázek 6.3.3: $|V_{var}| = 100$, $s = S_{max}$, pro každou omezenou bikliku $B \in \mathcal{B}$ platí $|V_1(B)| \leq 2$.

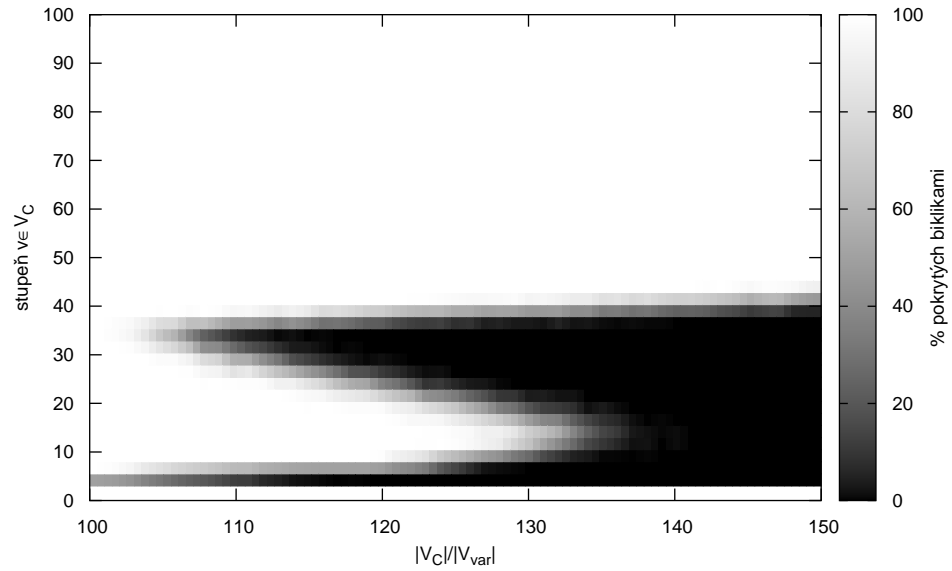
můžeme povšimnout u výsledků experimentu prezentovaném v tabulce 6.3.4. Dá se tedy usuzovat, že domněnka 5.2.1 je správná.

strategie	100		110		120		130		140		150	
	x_k	y_k	x_k	y_k	x_k	y_k	x_k	y_k	x_k	y_k	x_k	y_k
$S_{min}^{2,3}$	4	5	5	6	7	8	9	15	13	24	33	47
$S_{rand}^{2,3}$	4	5	5	6	7	8	9	15	13	24	33	47
$S_{max}^{2,3}$	4	5	5	6	7	8	9	15	14	24	41	87

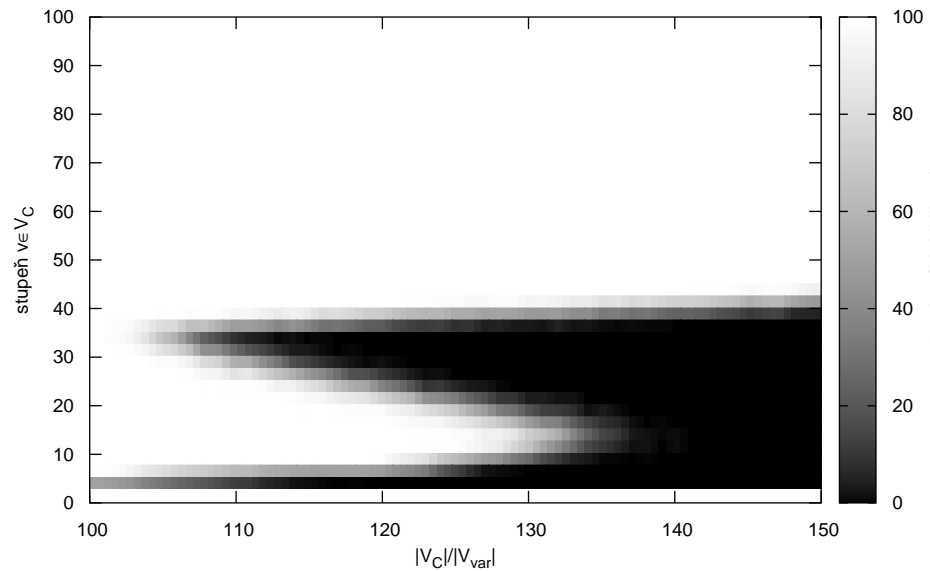
Tabulka 6.3.4: Fázový přechod typu K , velikost první partity omezené bikliky $|V_1| \leq 2$, $|V_{var}| = 100$.

Na obrázcích 6.3.5–6.3.7 jsou zobrazeny výsledky testů heuristiky, u které jsme neomezovali velikost první partity každé bikliky z pokrytí a velikost první partity celého grafu, který jsme se snažili pokrýt, je sto. Pro každou konfiguraci jsme vygenerovali sto náhodných grafů. Strategie S_{min} , S_{rand} a S_{max} použité v této heuristice označujeme S_{min}^{max} , S_{rand}^{max} a S_{max}^{max} .

U obrázků 6.3.5 a 6.3.6 reprezentujících experimenty, u nichž jsme použili heuristiky se strategiemi S_{min}^{max} a S_{rand}^{max} , si můžeme povšimnout toho, že pro grafy, jejichž poměr partit je menší než 135%, existuje interval v K -fázovém přechodu, kde jsou všechny grafy biklikově pokrytelné, ale při zvyšování stupně u vrcholů druhé partity toto pokrytí nenalezneme pro žádný graf. Pro dané r mezi 105% a 130% existují tři „ K -fázové přechody“ (jsou podintervaly K -fázového přechodu). První a třetí jsou stejného charakteru, tedy přechod od konfigurací, ve kterých heuristika nenalezne biklikové pokrytí ke konfiguracím, ve kterých heuristika nalezne pokrytí. Druhý přechod je přesně opačný. Tento jev je v menší míře pozorovatelný i u experimentu, v němž jsme použili heuristiku se strategií S_{max}^{max} a

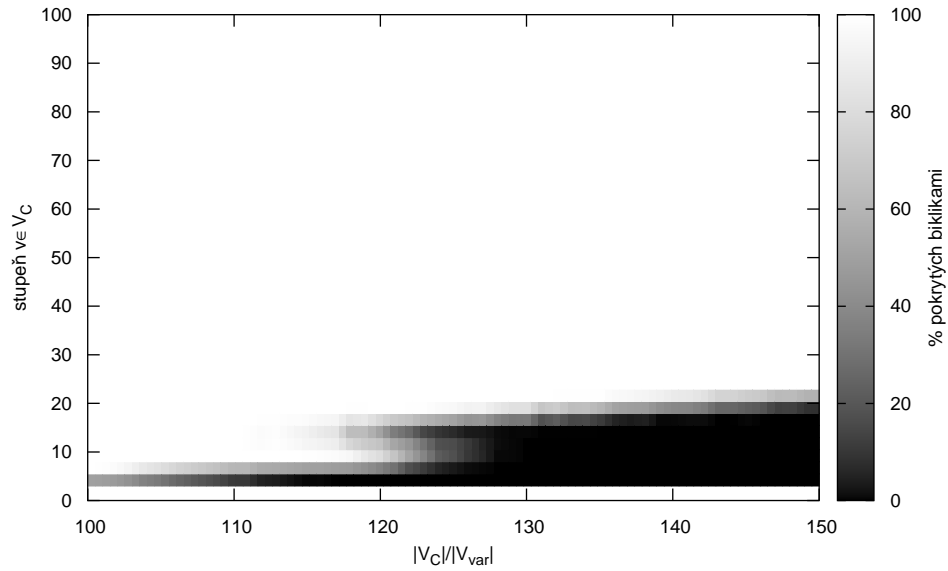


Obrázek 6.3.5: $|V_{var}| = 100$, $s = S_{min}$, pro každou omezenou bikliku $B \in \mathcal{B}$ platí $|V_1(B)| \leq |V_{var}|$. Pro každý zárodek Z platí $|V_2(Z)| \geq 3$.



Obrázek 6.3.6: $|V_{var}| = 100$, $s = S_{rand}$, pro každou omezenou bikliku $B \in \mathcal{B}$ platí $|V_1(B)| \leq |V_{var}|$. Pro každý zárodek Z platí $|V_2(Z)| \geq 3$.

jehož výsledky jsou zachyceny na obrázku 6.3.7. U prvního „ K -fázového přechodu“ u všech třech experimentů z obázků 6.3.5–6.3.7 je vidět, že je stejný, jako u experimentů z obrázků 6.3.1–6.3.3, tedy experimentu, kde jsme pro pokrytí používali jen omezené bikliky $K_{2,3}$ a $K_{1,1}$. Zároveň se druhý a třetí „fázový přechod“ u experimentů 6.3.1–6.3.3 projevuje mnohem více než u experimentu 6.3.7, z čehož by se dalo usoudit, že pokud použijeme strategii S_{max}^{max} , tedy strategii, kte-



Obrázek 6.3.7: $|V_{var}| = 100$, $s = S_{max}$, pro každou omezenou bikliku $B \in \mathcal{B}$ platí $|V_1(B)| \leq |V_{var}|$. Pro každý zárodek Z platí $|V_2(Z)| \geq 3$.

rá ve funkci zvolZárodek vybírá zárodky, které mají největší potenciál, funguje nejlépe. To potvrzují i data z tabulky 6.3.8 fázových přechodů pro tyto tři experimenty. Domněnka 5.2.1 je tedy správná a v tomto případě je rozdíl strategií mnohem větší.

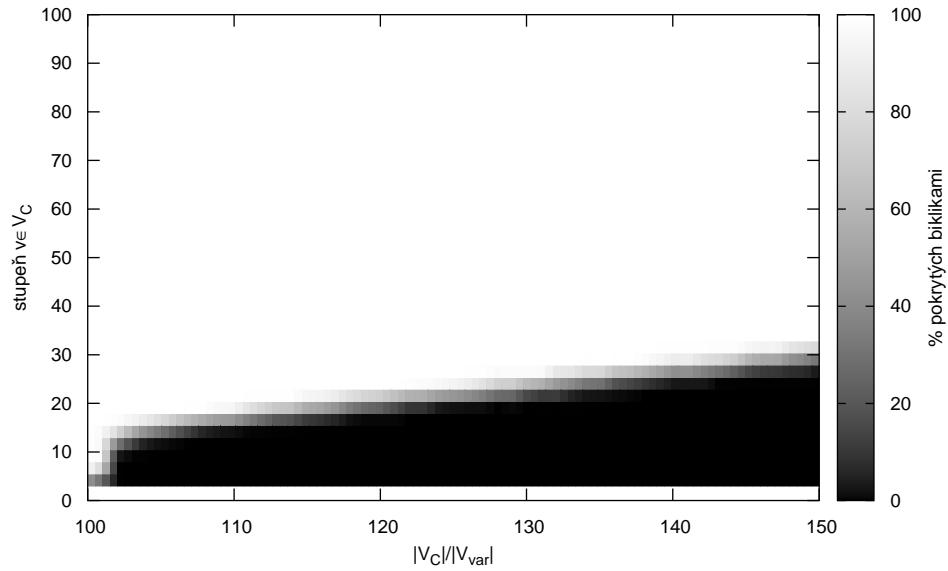
strategie	100		110		120		130		140		150	
	x_k	y_k	x_k	y_k	x_k	y_k	x_k	y_k	x_k	y_k	x_k	y_k
S_{min}^{max}	4	5	5	39	7	40	9	40	39	42	39	44
S_{rand}^{max}	4	5	5	39	7	40	9	40	39	42	39	44
S_{max}^{max}	4	5	5	6	7	17	15	20	18	22	20	23

Tabulka 6.3.8: Fázový přechod typu K , velikost první partity omezené bikliky $|V_1| \leq |V_{var}| \cdot |V_{var}| = 100$.

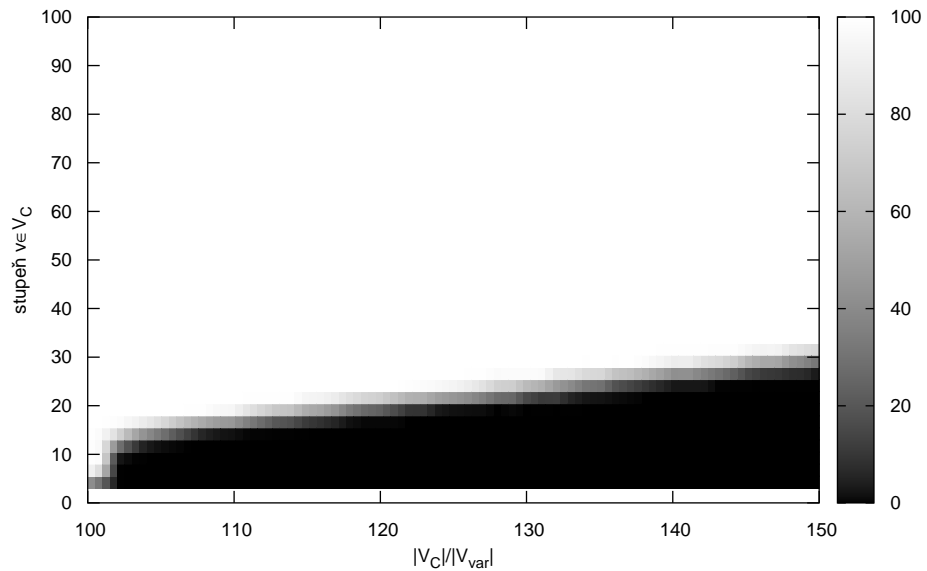
strategie	100		110		120		130		140		150	
	x_k	y_k	x_k	y_k	x_k	y_k	x_k	y_k	x_k	y_k	x_k	y_k
S_{min}^{max7}	4	6	15	19	18	22	22	26	24	30	27	32
S_{rand}^{max7}	4	6	15	19	18	22	22	26	24	30	27	32
S_{max}^{max7}	4	6	15	17	17	19	19	21	20	23	22	23

Tabulka 6.3.12: Fázový přechod typu K , velikost první partity omezené bikliky $|V_1| \leq |V_{var}|$. Velikost druhé partity zárodku minimálně 7. $|V_{var}| = 100$.

Další experiment, který jsme provedli, má téměř stejné parametry jako experiment, jehož výsledky jsou na obrázcích 6.3.5–6.3.7, jen jsme v heuristice 5.2.1 ve funkci 5.2.5 (vyčistiZárodky) smazali zárodky, jejichž druhá partita byla menší



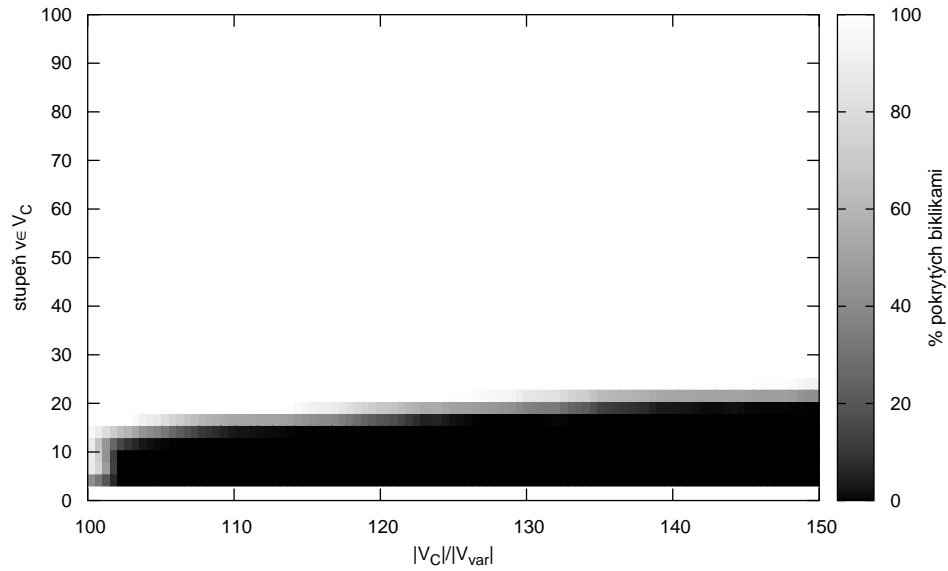
Obrázek 6.3.9: $|V_{var}| = 100$, $s = S_{min}$, pro každou omezenou bikliku $B \in \mathcal{B}$ platí $|V_1(B)| \leq |V_{var}|$. Pro každý zárodek Z platí $|V_2(Z)| \geq 7$.



Obrázek 6.3.10: $|V_{var}| = 100$, $s = S_{rand}$, pro každou omezenou bikliku $B \in \mathcal{B}$ platí $|V_1(B)| \leq |V_{var}|$. Pro každý zárodek Z platí $|V_2(Z)| \geq 7$.

než sedm. Tímto krokem bychom se mohli zbavit kolísání uvnitř K -fázového přechodu. Strategie S_{min} , S_{rand} a S_{max} použité v této heuristice označujeme S_{min}^{max7} , S_{rand}^{max7} a S_{max}^{max7} .

Můžeme si povšimnout, že se problém s více „ K -fázovými přechody“ neprojevil jako u experimentů na obrázcích 6.3.5–6.3.7. Zároveň porovnáním obrázků 6.3.5–6.3.7 a 6.3.9–6.3.11, nebo tabulek 6.3.8 a 6.3.12 zjistíme, že pro strategie



Obrázek 6.3.11: $|V_{var}| = 100$, $s = S_{max}$, pro každou omezenou bikliku $B \in \mathcal{B}$ platí $|V_1(B)| \leq |V_{var}|$. Pro každý zárodek Z platí $|V_2(Z)| \geq 7$.

S_{min}^{max7} a S_{rand}^{max7} mají sice mnohem vyšší dolní hranici K -fázového přechodu než strategie S_{min}^{max} a S_{rand}^{max} , ale zároveň mají výrazně nižší horní hranici K -fázového přechodu oproti strategiím S_{min}^{max} a S_{rand}^{max} . U strategie S_{max}^{max7} je vyšší jak dolní hranice K -fázového přechodu ve srovnání se strategií S_{max}^{max} , tak dolní hranice K -fázového přechodu oproti strategii S_{max}^{max} . Vzhledem k tomu, že strategie S_{max}^{max} je lepší než strategie S_{max}^{max7} , ale i než S_{min}^{max7} , S_{rand}^{max7} , S_{min}^{max} a S_{rand}^{max} , tak se omezení velikosti zárodků zespodu příliš nevyplatí.

strategie	100		110		120		130		140		150	
	x_k	y_k	x_k	y_k	x_k	y_k	x_k	y_k	x_k	y_k	x_k	y_k
$S_{min}^{2,3}$	4	5	5	6	7	8	9	15	13	24	33	47
S_{max}^{max}	4	5	5	6	7	17	15	20	18	22	20	23
S_{max}^{max7}	4	6	15	17	17	19	19	21	20	23	22	23

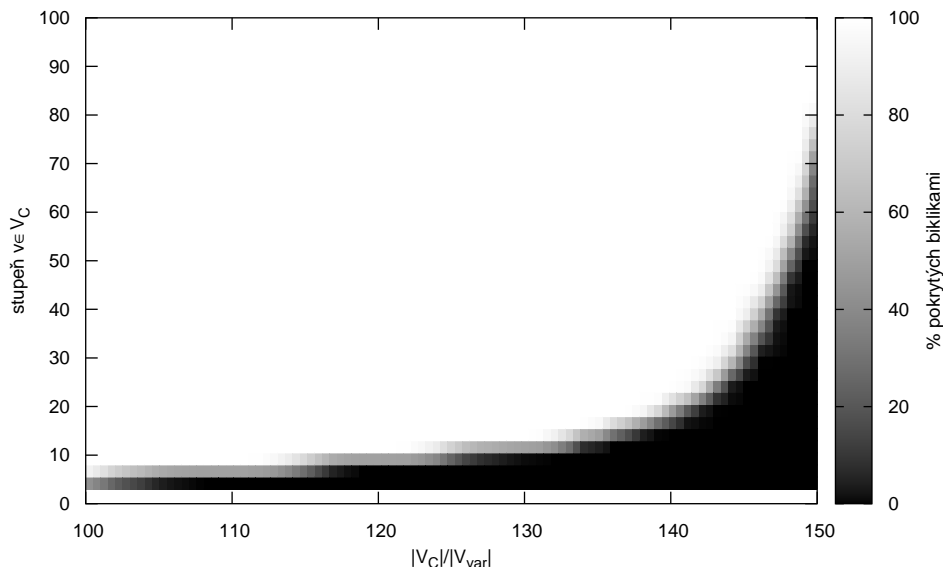
Tabulka 6.3.13: Fázový přechod typu K , porovnání heuristiky pro různé strategie a různá omezení na velikost první partity biklik. $|V_{var}| = 100$.

V tabulce 6.3.13 jsou zaznamenány výsledky experimentu, kde jsme sledovali počet grafů s velikostí první partity 100, které pokryje heuristika s různými strategiemi S_{min} , S_{rand} a S_{max} a různými omezeními na první nebo druhou partitu zárodků a biklik z výsledného biklikového pokrytí.

Pokud bychom měli zvolit jednu strategii a omezení na bikliky pro heuristiku, pak se jeví jako nejlepší heuristika S_{max}^{max} , tedy heuristika, která neklade žádné omezení na velikost partit zárodku nebo bikliky, a pro volbu zárodku ve funkci zvolZárodek použije strategii S_{max} , tedy zvolí takový zárodek, který má největší možnost rozšíření, což znamená, že má maximální druhou partitu mezi všemi zárodky.

Na formule s poměrem partit mezi 100% a 140% by bylo lepší zvolit heuristiku se strategií $S_{min}^{2,3}$, která omezuje velikost první partity biklik z pokrytí dvěma a funkce zvolZárodek vybírá nejmenší možný, vzhledem k velikosti druhé partity, tedy používá strategii S_{min} .

Všechny experimenty 6.3.1–6.3.11 jsme provedli i pro grafy, jejichž první partita má velikost 200 a výsledky jsou zobrazeny v obrázcích 6.3.14–6.3.23 a v tabulce 6.3.25.

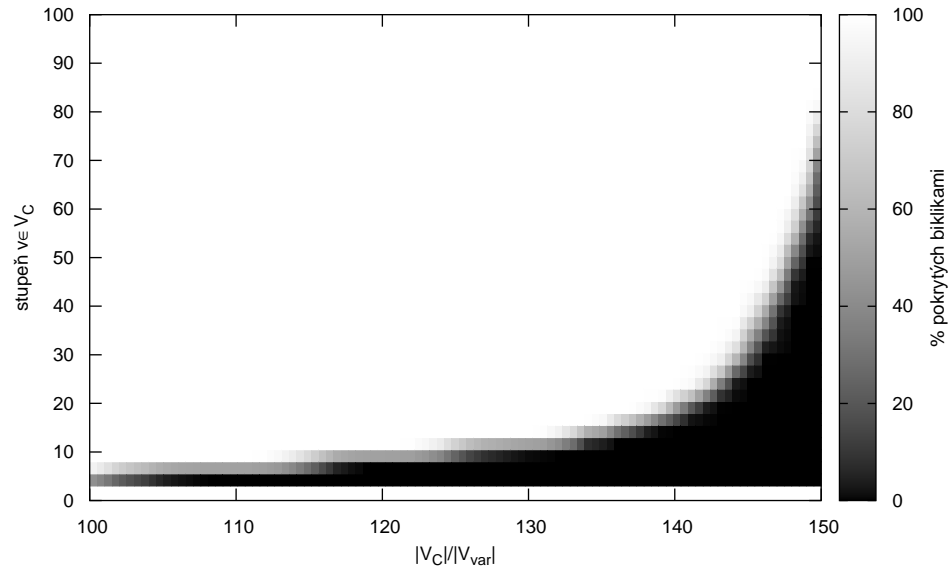


Obrázek 6.3.14: $|V_{var}| = 200$, $s = S_{min}$, pro každou omezenou bikliku $B \in \mathcal{B}$ platí $|V_1(B)| \leq 2$.

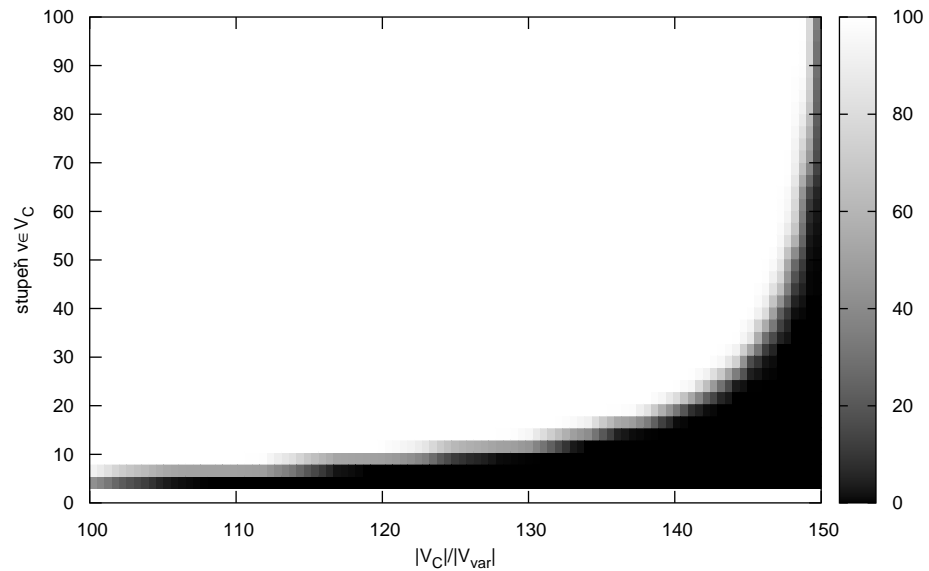
Na obrázcích 6.3.14–6.3.16 jsou zobrazeny výsledky experimentu, ve kterém jsme testovali všechny strategie S_{min} , S_{rand} , S_{max} v heuristice, ve které jsme omežili velikost biklik použitých v pokrytí grafy $K_{2,3}$. V tabulce 6.3.17 používáme pro výsledky těchto experimentů u jednotlivých strategií označení $S_{min}^{2,3}$, $S_{rand}^{2,3}$ a $S_{max}^{2,3}$. Porovnáním obrázků 6.3.1–6.3.3 reprezentujících výsledky experimentu s grafy, jejichž první partita je rovná 100 a obrázků 6.3.14–6.3.16, které reprezentují výsledky experimentu, v němž jsme uvažovali grafy s velikostí první partity 200, zjistíme, že průběh pro různé strategie je velice obdobný s rozdílem takovým, že pro experiment, ve kterém jsme použili grafy s větší velikostí první partity, se K -fázový přechod posunul k vyšším hodnotám. Pro poměry partit $r < 140\%$ je tento rozdíl K -fázových přechodů velice malý, ale u grafů s vyšším poměrem než 140% roste K -fázový přechod u experimentu s grafy s velikostí první partity 200 mnohem více, než u experimentů s velikostí první partity 100 a k tomu, aby heuristika našla biklikové pokrytí, musí mít graf velice vysoký stupeň vrcholů druhé partity.

I u grafů s velikostí první partity $|V_{var}| = 200$ je z obrázků 6.3.14–6.3.16 a z tabulky 6.3.17 patrné, že heuristika, která používá jednu ze strategií $S_{min}^{2,3}$, $S_{rand}^{2,3}$, je lepší, tedy nalezne pokrytí na více grafech, než heuristika ze strategií $S_{max}^{2,3}$.

Na obrázcích 6.3.18–6.3.20 jsou výsledky experimentu, v němž jsme porovnávali heuristiku 5.2.1 se strategiemi S_{min} , S_{rand} a S_{max} a bez omezení na ve-



Obrázek 6.3.15: $|V_{var}| = 200$, $s = S_{rand}$, pro každou omezenou bikliku $B \in \mathcal{B}$ platí $|V_1(B)| \leq 2$.

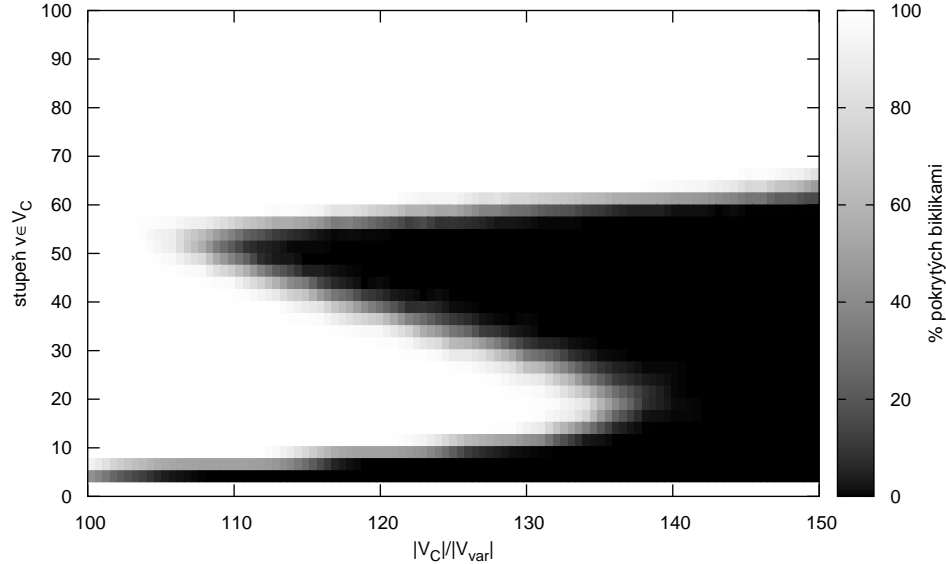


Obrázek 6.3.16: $|V_{var}| = 200$, $s = S_{max}$, pro každou omezenou bikliku $B \in \mathcal{B}$ platí $|V_1(B)| \leq 2$.

likost první partity zárodků a u experimentů, jejichž výsledky jsou zobrazeny na obrázcích 6.3.21–6.3.23 jsme neomezovali velikost první partity, ale ve funkci vyčistiZárodky jsme odebrali zárodky s velikostí druhé partity menší než sedm. Rozdíl mezi K -fázovými přechody u heuristiky, jež zárodky zdola neomezuje a heuristiky, která je omezuje sedmi, je u experimentů na obrázcích 6.3.21–6.3.23 velice podobný jako u experimentů na obrázcích 6.3.9–6.3.11 a z toho důvodu

strategie	100		110		120		130		140		150	
	x_k	y_k	x_k	y_k	x_k	y_k	x_k	y_k	x_k	y_k	x_k	y_k
$S_{min}^{2,3}$	4	5	7	7	9	9	11	12	17	24	59	86
$S_{rand}^{2,3}$	4	5	7	7	9	9	11	12	17	24	59	86
$S_{max}^{2,3}$	4	5	7	7	9	10	11	13	18	23	73	> 100

Tabulka 6.3.17: Fázový přechod typu K , velikost první partity omezené bikliky $|V_1| \leq 2 \cdot |V_{var}| = 200$.



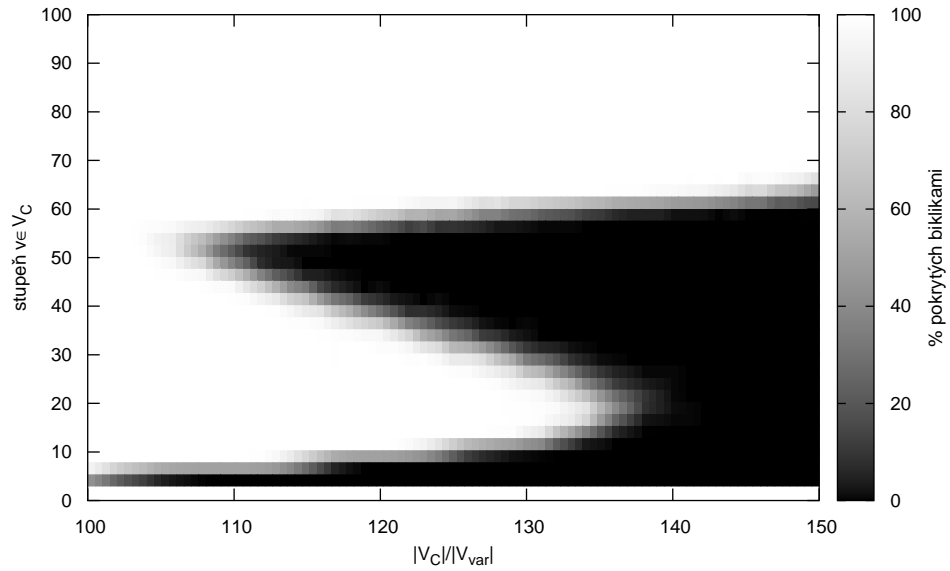
Obrázek 6.3.18: $|V_{var}| = 200$, $s = S_{min}$, pro každou omezenou bikliku $B \in \mathcal{B}$ platí $|V_1(B)| \leq |V_{var}|$. Pro každý zárodek Z platí $|V_2(Z)| \geq 3$.

tyto výsledky nebudeme širěji popisovat. Výsledky z těchto experimentů jsou zaznamenány v tabulce 6.3.24.

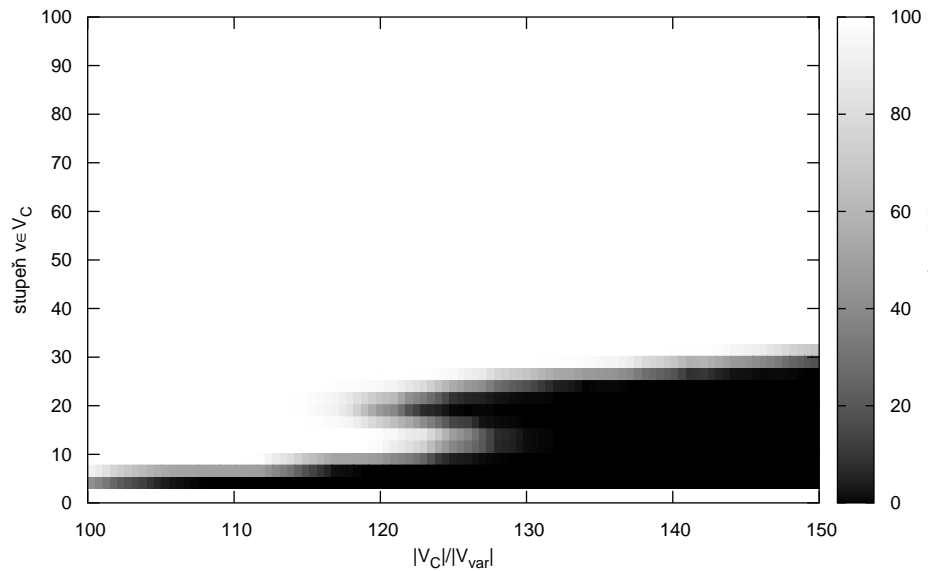
Heuristiky, které neomezují velikost první partity biklik z pokrytí, je nejspěšnější heuristika se strategií $S_{max}^{max[7]}$, zatímco heuristiky $S_{min}^{max[7]}$ a $S_{rand}^{max[7]}$, které se chovají stejně, naleznou pokrytí u mnohem méně grafů. Toto je patrné jak z obrázků 6.3.18–6.3.23 tak z tabulky K -fázových přechodů 6.3.24.

Stejně jako u experimentu 6.3.14–6.3.16 je i u tohoto experimentu potřeba vyššího stupně vrcholů $C \in V_C$ pro stejný poměr partit $r = \frac{|V_C|}{|V_{var}|} 100$ grafu než u experimentu 6.3.5–6.3.11 se sto vrcholy v první partitě.

V tabulce 6.3.25 jsou shrnuty nejlepší strategie ze všech experimentů 6.3.1–6.3.23. Strategie $S_{min}^{2,3}$ nalezne více pokrytí až do poměru partit grafů 140% jak u grafů se stem vrcholů v první partitě, tak se dvěma sty vrcholy v první partitě. Pro grafy se stem vrcholů v první partitě je tato hranice rovna 145%. Dolní hranice strategie K -fázového přechodu S_{max}^{max} začne být nižší až při poměru partit 145%. Nad touto hranicí je již lepší použít heuristiku se strategií S_{max}^{max} .



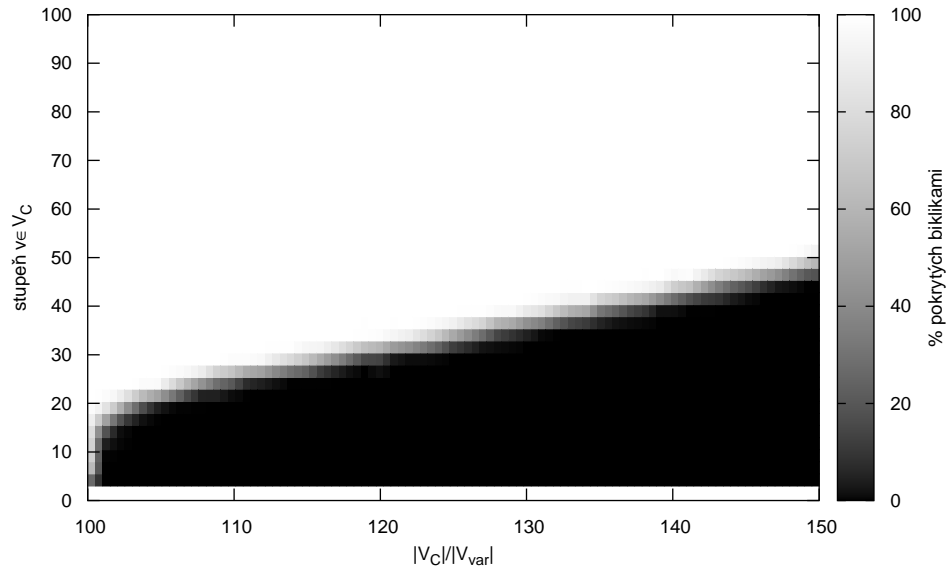
Obrázek 6.3.19: $|V_{var}| = 200$, $s = S_{rand}$, pro každou omezenou bikliku $B \in \mathcal{B}$ platí $|V_1(B)| \leq |V_{var}|$. Pro každý zárodek Z platí $|V_2(Z)| \geq 3$.



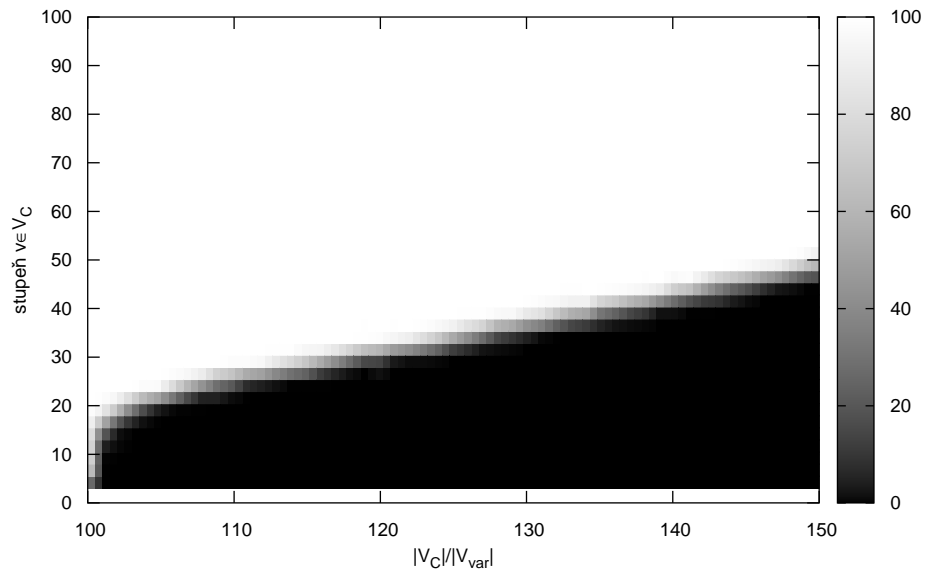
Obrázek 6.3.20: $|V_{var}| = 200$, $s = S_{max}$, pro každou omezenou bikliku $B \in \mathcal{B}$ platí $|V_1(B)| \leq |V_{var}|$. Pro každý zárodek Z platí $|V_2(Z)| \geq 3$.

6.4 Porovnání matched a biklikově pokrytelných formulí

Posledním experimentem, který jsme provedli, zkusíme, jak moc se posune fázový přechod pro matched formule, pokud povolíme rozšíření pokrytí o bikliky.

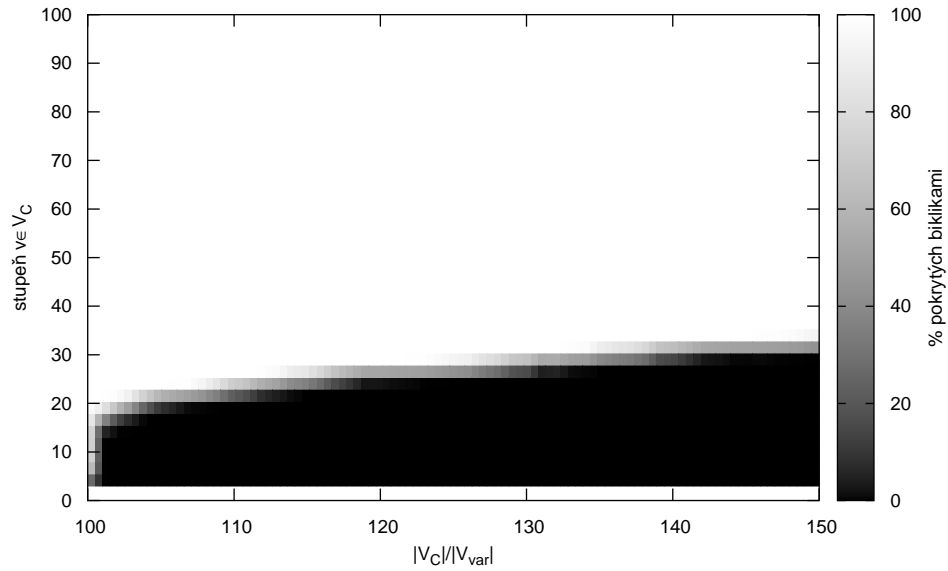


Obrázek 6.3.21: $|V_{var}| = 200$, $s = S_{min}$, pro každou omezenou bikliku $B \in \mathcal{B}$ platí $|V_1(B)| \leq |V_{var}|$. Pro každý zárodek Z platí $|V_2(Z)| \geq 7$.



Obrázek 6.3.22: $|V_{var}| = 200$, $s = S_{rand}$, pro každou omezenou bikliku $B \in \mathcal{B}$ platí $|V_1(B)| \leq |V_{var}|$. Pro každý zárodek Z platí $|V_2(Z)| \geq 7$.

Na obrázcích 6.4.1 a 6.4.1 jsou výsledky pro grafy se stem vrcholů a dvěma sty vrcholů v první partitě. Experimenty jsme provedli pro k -KNF formule, $3 \leq k \leq 5$, jejichž poměr klauzulí ku proměnným je mezi 80% a 120% a tento poměr je zobrazen na ose x . Na ose y je pak poměr formulí, které jsou matched respektive pro které našla heuristika 5.2.1 biklikové pokrytí. Fázové přechody pro k -KNF matched formule mají horní hranici intervalu na 100%, což platí z toho důvodu, že



Obrázek 6.3.23: $|V_{var}| = 200$, $s = S_{max}$, pro každou omezenou bikliku $B \in \mathcal{B}$ platí $|V_1(B)| \leq |V_{var}|$. Pro každý zárodek Z platí $|V_2(Z)| \geq 7$.

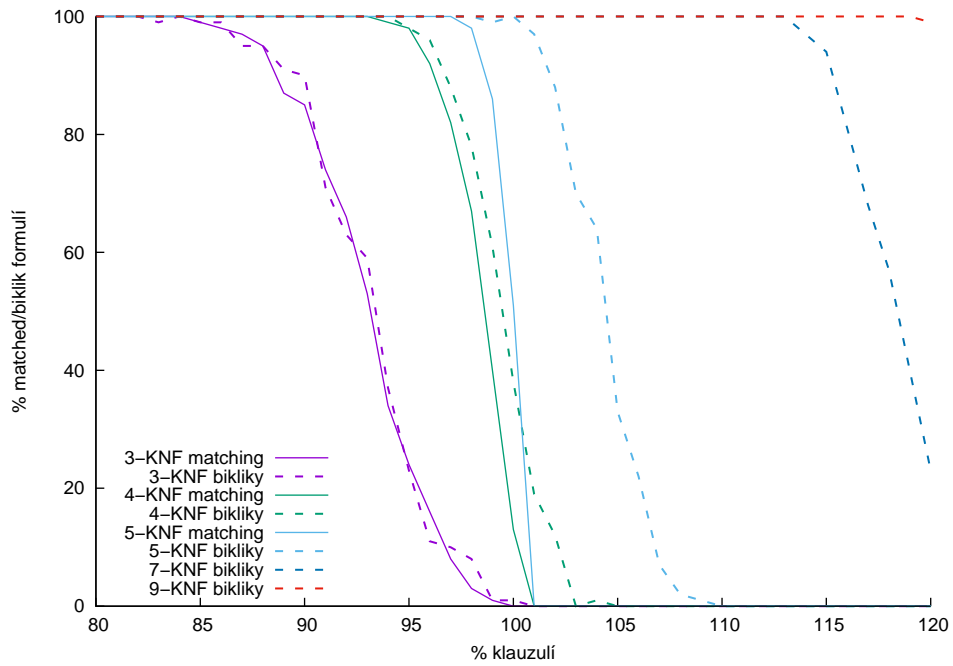
strategie	100		110		120		130		140		150	
	x_k	y_k	x_k	y_k	x_k	y_k	x_k	y_k	x_k	y_k	x_k	y_k
S_{min}^{max}	4	5	7	57	9	60	11	62	60	63	61	68
S_{rand}^{max}	4	5	7	57	9	60	11	62	60	63	61	68
S_{max}^{max}	4	5	7	7	9	25	12	27	27	30	29	31
S_{min}^{max7}	4	9	23	28	29	34	34	40	39	45	46	52
S_{rand}^{max7}	4	9	23	28	29	34	34	40	39	45	46	52
S_{max}^{max7}	4	9	22	24	25	27	28	29	30	31	32	34

Tabulka 6.3.24: Fázový přechod typu K , velikost první partity omezené bikliky $|V_1| \leq |V_{var}|$. Velikost druhé partity zárodku minimálně 7. $|V_{var}| = 200$.

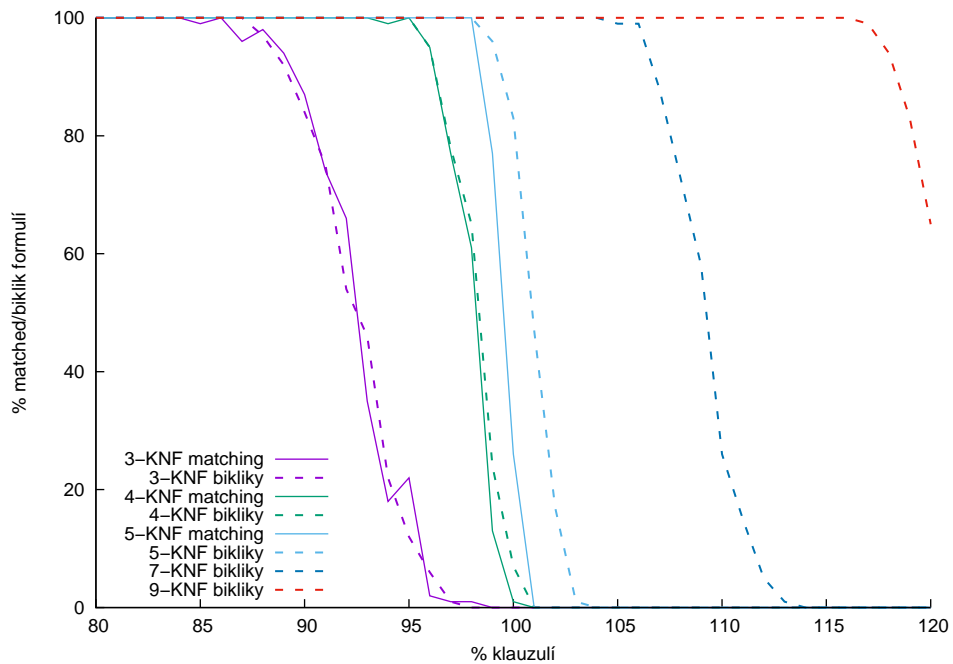
$ V_{var} $	strategie	100		110		120		130		140		150	
		x_k	y_k	x_k	y_k	x_k	y_k	x_k	y_k	x_k	y_k	x_k	y_k
100	$S_{min}^{2,3}$	4	5	5	6	7	8	9	15	13	24	33	47
100	S_{max}^{max}	4	5	5	6	7	17	15	20	18	22	20	23
100	S_{max}^{max7}	4	6	15	17	17	19	19	21	20	23	22	23
200	$S_{min}^{2,3}$	4	5	7	7	9	9	11	12	17	24	59	86
200	S_{max}^{max}	4	5	7	7	9	25	12	27	27	30	29	31
200	S_{max}^{max7}	4	9	22	24	25	27	28	29	30	31	32	34

Tabulka 6.3.25: Fázový přechod typu K , porovnání heuristiky pro různé strategie a různá omezení na velikost první partity biklik.

pokud je víc klauzulí než proměnných, nemůže už být formule matched. Můžeme si povšimnout, že pro $k \geq 4$ křivka znázorňující procento matched formulí končí v hodnotě 101%. To nastává z toho důvodu, že testy děláme pro poměr partit



Obrázek 6.4.1: Porovnání fázového přechodu matched formulí a biklikově splnitelných formulí. $|V_{var}| = 100$. Provedeno sto testů.



Obrázek 6.4.2: Porovnání fázového přechodu matched formulí a biklikově splnitelných formulí. $|V_{var}| = 200$. Provedeno sto testů.

$r = 80\%, 81\%, \dots, 100\%, 101\%, \dots, 120\%$, a zároveň, pokud je stejně klauzulí jako proměnných, tak je ještě malé procento k -KNF formulí s $k \geq 4$ matched.

Z obrázků 6.4.1 a 6.4.2 si můžeme povšimnout, že fázové přechody pro každé k -KNF, $k \leq 4$ jsou téměř stejné pro křivky, které reprezentují poměr matched formulí a pro křivky, které reprezentují poměr formulí, pro které našla heuristika

5.2.1 biklikové pokrytí (zde nezáleží na omezení velikosti partit zárodků ani na použité strategii).

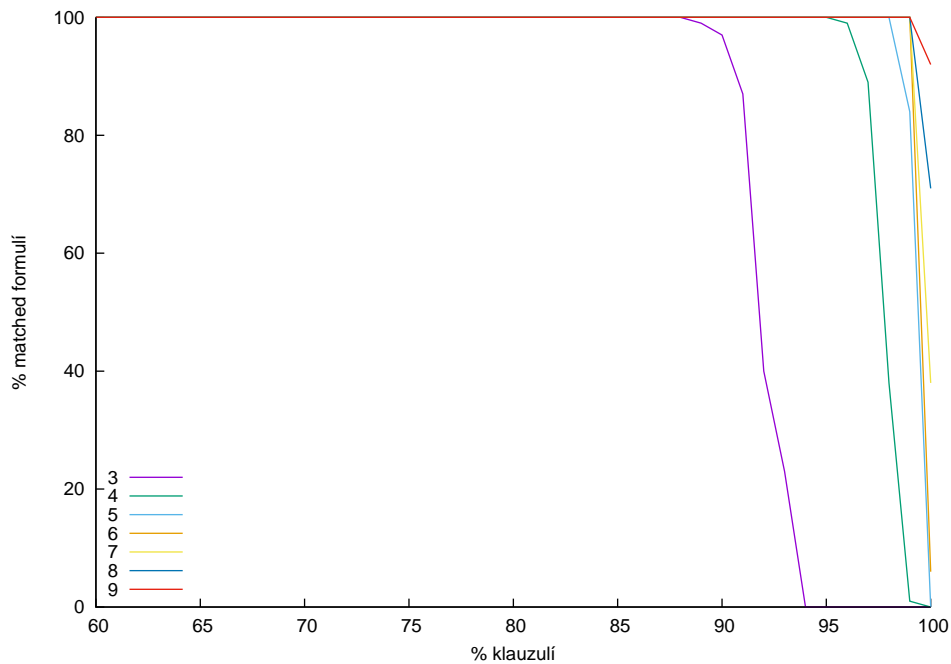
U 5-KNF, 7-KNF a 9-KNF je horní hranice fázového přechodu pro matched formule 100%, zatímco pro biklikově splnitelné formule je už dolní hranice fázového přechodu větší než 100%. Pro 9-KNF je to dokonce 120%.

7. Závěr

Nejdříve si zhodnotíme výsledky experimentů z kapitoly 6 a poté shrneme poznatky, které jsme získali v průběhu práce.

7.1 Zhodnocení výsledků experimentů

V kapitole 6, sekci 6.2 jsme se zabývali experimenty pro ověření odhadu dokázaného Johnem Francem a Allen Van Gelderem [21], který ukazuje, že 3-KNF formule s poměrem počtu klauzulí k počtu proměnných rovným nebo menším 0.64 jsou téměř jistě matched. Zkoumali jsme, jestli se v experimentech objeví nějaký fázový přechod $\langle x_k, y_k \rangle$, pro který platí, že k -KNF formule s poměrem počtu klauzulí k počtu proměnných menším než x_k jsou téměř jistě matched a formule s poměrem počtu klauzulí k počtu proměnných větším než y_k téměř jistě matched nejsou. Experiment jsme provedli pro k -KNF s $3 \leq k \leq 9$ a formule, na nichž jsme experiment prováděli jsme generovali náhodně.



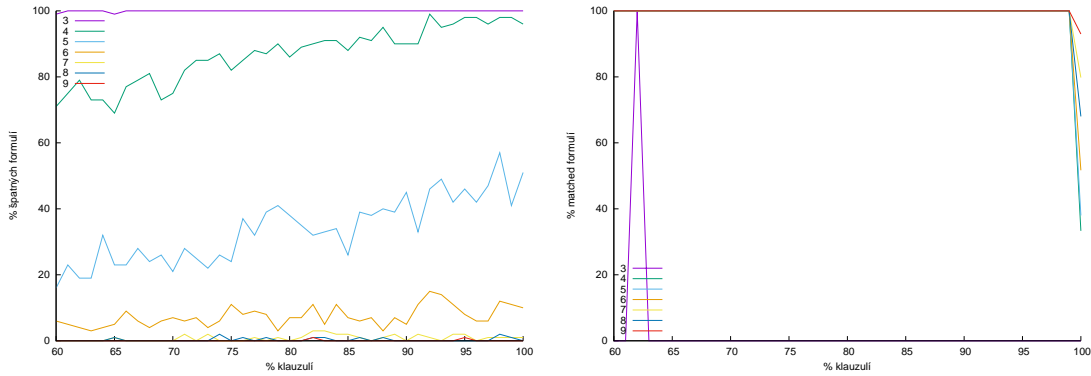
Obrázek 7.1.1: Fázový přechod pro matched k -KNF formule.

3-KNF		4-KNF		5-KNF	k -KNF ($k \geq 6$)
x_3	y_3	x_4	y_4	x_5	x_k
89	93	96	99	99	100

Tabulka 7.1.2: Dolní hranice fázového přechodu matched formulí.

Výsledky experimentu jsou zobrazeny na obrázku 7.1.1 a v tabulce 7.1.2 a můžeme si povšimnout, že dolní hranice fázového přechodu pro 3-KNF byla v experimentu mnohem vyšší, než byl odhad dokázaný Johnem Francem a Allen Van Gelderem [21]. Zároveň pro k -KNF pro $k \geq 5$ je dolní hranice fázového přechodu

na 99%, k -KNF jsou téměř jistě matched pro poměr mezi počtem klauzulí k počtu proměnných menším než 99%.



(a) Formule odfiltrované funkcí jednotkováGPropagace.

(b) Poměr matched formulí z celkového počtu formulí neodfiltrovaných funkcí jednotkováGPropagace.

Obrázek 7.1.3: Formule s průměrným počtem klauzulí k .

Dále jsme vyzkoušeli generovat formule, jejichž klauzule obsahují průměrně k proměnných. U takto generovaných formulí nastal problém v tom, že se pro malé k ($3 \leq k \leq 5$) u většiny formulí ukázalo již ve funkci 5.2.4 (jednotkováGPropagace), že nejsou matched. Výsledky tohoto experimentu jsou zachyceny na obrázku 7.1.3. Na formulích, na který funkce 5.2.4 (jednotkováGPropagace) uspěla, jsme provedli testy, které jsou zobrazeny na obrázku 7.1.3, kde si můžeme povšimnout, že po odfiltrování formulí funkcí 5.2.4 (jednotkováGPropagace) jsou téměř všechny formule matched. Z výsledků lze tedy vyčíst to, že u zcela náhodně generovaných formulí s průměrným počtem proměnných v klauzuli k s malým k není většina formulí matched a můžeme to zjistit jen jednotkovou grafovou propagací.

V dalších experimentech jsme zkoumali, pro jakou část formulí heuristika 5.2.1 popsaná v sekci 5.2 s různými strategiemi S_{min} , S_{rand} a S_{max} pro výběr zárodku a s různými dalšími omezeními kladenými na omezené biklikly v pokrytí, nalezneme biklikové pokrytí.

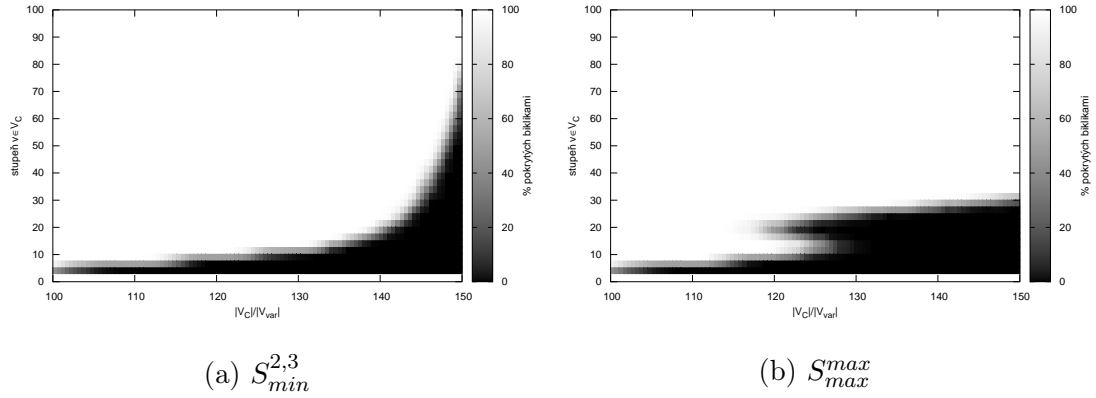
- (1) Strategie S_{min} vybere zárodek Z , který má minimální velikost druhé partity $|V_2(Z)|$,
- (2) strategie S_{rand} vybere náhodný zárodek,
- (3) strategie S_{max} vybere zárodek Z , který má maximální velikost druhé partity $|V_2(Z)|$.

Pro strategii S_s , $s \in \{min, rand, max\}$, jsme použili následující omezení na zárodky a biklikly z nalezeného pokrytí:

- (1) pro omezení $S_s^{2,3}$ platí, že omezené biklikly z biklikového pokrytí jsou buď $K_{2,3}$ nebo $K_{1,1}$, tedy párovací hrany a pro každý zárodek Z platí $|V_2(Z)| \geq 3$,
- (2) pro omezení S_s^{max} platí, že v biklikovém pokrytí může být libovolná omezená biklika B taková, že $|V_2(Z)| > |V_1(Z)|$ a pro každý zárodek Z platí $|V_2(Z)| \geq 3$,

- (3) pro omezení S_s^{max7} platí, že v biklikovém pokrytí může být libovolná omezená biklika B taková, že $|V_2(Z)| > |V_1(Z)|$ a pro každý zárodek Z platí $|V_2(Z)| \geq 7$.

Ověřili jsme domněnku 5.2.1, tedy že pro strategie $S_s^{2,3}$ bude nejlepší použít strategii $s = min$ a pro strategie s většími omezenými biklikami v pokrytí, jako u $S_s^{max[7]}$, bude nejlepší strategie $s = max$. Vzhledem k velikosti obrázků a tabulek, jež reprezentují výsledky testování heuristiky, je zde nebudeme opakovat, ale můžeme je nalézt v kapitole 6, v sekci 6.3 na obrázcích a v tabulkách 6.3.1–6.3.25. Z výsledků těchto experimentů je zřejmé, že domněnka 5.2.1 je správná. Navíc jsme testovali i strategii $s = rand$ a zjistili jsme, že se chová obdobně jako strategie $s = min$ a tudíž jí dále nebudeme zmiňovat.



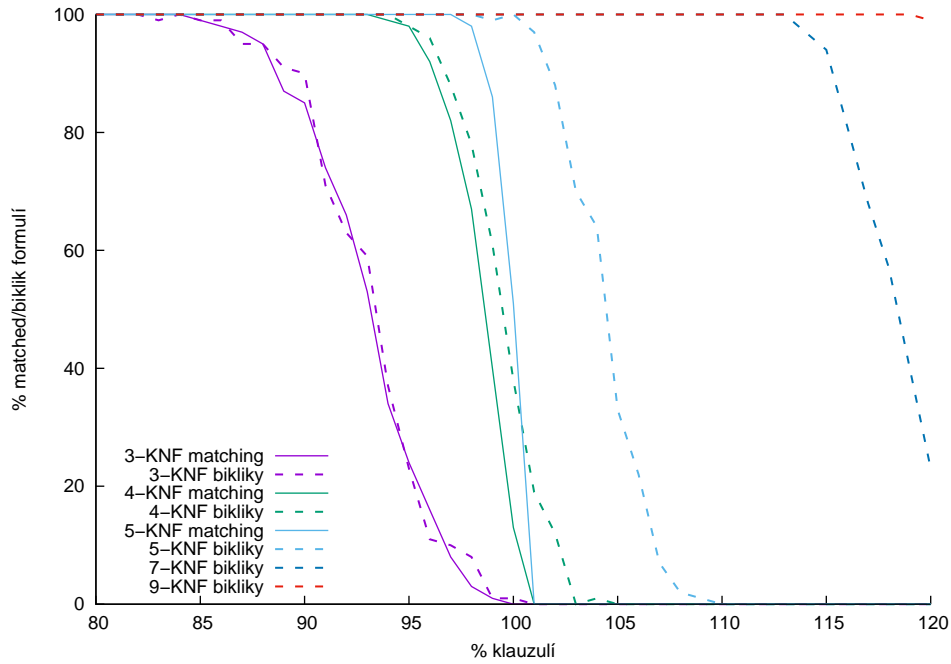
Obrázek 7.1.4: $|V_{var}| = 200$, pro každou omezenou biklika $B \in \mathcal{B}$ platí $|V_1(B)| \leq 2$.

$ V_{var} $	strategie	100		110		120		130		140		150	
		x_k	y_k	x_k	y_k	x_k	y_k	x_k	y_k	x_k	y_k	x_k	y_k
200	$S_{min}^{2,3}$	4	5	7	7	9	9	11	12	17	24	59	86
200	S_{max}^{max}	4	5	7	7	9	25	12	27	27	30	29	31
200	S_{max}^{max7}	4	9	22	24	25	27	28	29	30	31	32	34

Tabulka 7.1.5: K -Fázový přechod, porovnání heuristiky pro různé strategie a různá omezení na velikost první partity biklik. $|V_{var}| = 200$.

Pro formule, jejichž poměr počtu klauzulí k počtu proměnných r je menší než 140%, je výhodnější používat strategii $S_{min}^{2,3}$, tedy strategii, která omezuje biklikly použité v biklikovém pokrytí na $K_{2,3}$ a $K_{1,1}$, zatímco strategie, v níž neomezuje velikost první partity omezených biklik v pokrytí, je vhodnější pro vyšší poměry, jak je možno vidět z tabulky 7.1.5.

V posledním experimentu jsme zkoumali, jaký je rozdíl mezi fázovým přechodem u experimentu na matched k -KNF formulí a fázovým přechodem u experimentu s biklikově splnitelnými k -KNF formulemi. V experimentu jsme zjistili, že tyto fázové přechody jsou téměř stejné a pokud povolíme v pokrytí incidenčního grafu omezené biklikly větší než $K_{1,1}$, pouze nám to přirozeně rozšíří fázový přechod pro formule, jež mají více klauzulí než proměnných a nemohou tedy být matched. Výsledky tohoto experimentu jsou zachyceny na obrázku 7.1.6.



Obrázek 7.1.6: Porovnání fázového přechodu matched formulí a biklikově splnitelných formulí. $|V_{var}| = 100$. Provedeno sto testů.

Nalezli jsme tedy fázový přechod i K -fázový přechod u experimentů provedených nad náhodnými formulemi a potvrdili jsme, že nějaký vzor easy-hard-easy v rozpoznávání biklikově splnitelných formulí existuje.

7.2 Závěr

V kapitole 3 jsme si nadefinovali třídy formulí matched, biklikově splnitelných a var-splnitelných, a ukázali jsme popsali jejich vzájemné inkluze. Díky tomu, že matched formule i biklikově splnitelné formule jsou v inkluzi s var-splnitelnými, jsou tyto třídy uzavřené na změnu polarity libovolných literálů ve formulích z daných tříd.

V kapitole 4 jsme popsali parametrizovaný algoritmus uvedený Stefanem Szeidrem [34], který řeší splnitelnost formulí s pevným parametrem k , kde $k = \delta^*(F)$. Principy použité v algoritmu [34] jsme se pokusili zobecnit pro biklikovou deficienci δ^B v kapitole 5 sekci 5.3, což skončilo neúspěchem zejména kvůli problému zobecnění δ^* -kritičnosti na biklikovou δ^B -kritičnost. Navrhli jsme dvě možná zobecnění.

- Pokud je formule F δ_1^B -kritická, libovolným ohodnocením $\varepsilon \in \{0, 1\}$ libovolné proměnné x formule F vznikne formule $F[x = \varepsilon]$, která má ostře menší biklikovou deficienci δ^B než formule F ,
- pokud je formule F δ_2^B -kritická, libovolným částečným ohodnocením τ na proměnných z libovolné bikliky B z nějakého maximálního biklikového pokrytí \mathcal{B} vznikne formule $F[\tau]$, která má ostře menší biklikovou deficienci δ^B než formule F .

To, že rozšíření definice δ^* -kritičnosti jedním z těchto způsobů není postačující pro zobecnění věty 4.1.3 z kapitoly 4, jsme ukázali na protipříkladu formule F_1 rozebraném v příkladu 5.3.1.

Pro zobecnění algoritmu prezentovaného Stephanem Szeidrem [34] lze zkoumat zobecnění požadavku na 2-expandovatelnost a prozkoumat, zdali existuje nějaká k -expandovatelnost poskytující to samé biklikové deficienci jako 2-expandovatelnost párování.

V kapitole 5 jsme si popsali různé algoritmy a heuristiky, které se zabývají pokrýváním bipartitního grafu úplnými bipartitními podgrafy. Z heuristiky [26], fungující na hladovém přístupu jsme navrhli heuristiku na hledání biklikového pokrytí. V heuristice jsme navrhli několik strategií pro výběr zárodku S_{min} , S_{rand} a S_{max} a zformulovali jsme domněnku 5.2.1, která tvrdí, že pro biklikové pokrytí, jež omezuje velikost první partity omezených biklik dvěma, bude výhodnější strategie S_{min} . Naopak pro biklikové pokrytí, jež nemá omezení na omezené bikliky z biklikového pokrytí bude nejlepší heuristika 5.2.1 používající strategii S_{max} , která vybírá zárodky, jež mají největší potenciál k rozšíření funkcí 5.2.1 (rozšiřZárodek) a mají tedy potenciál k tomu, aby v pokrytí byly co největší omezené bikliky.

Dalšími možnými strategiemi pro heuristiku 5.2.1, které by bylo možno prozkoumat, ale v této práci jsme se jim nevěnovali, by mohly být strategie, které by nevybíraly zárodek s nejmenším/největším počtem vrcholů v druhé partitě, ale náhodný zárodek z k zárodků, jež mají nemenší/největší počet vrcholů v druhé partitě.

V experimentech popsanych v sekci 6.3 jsme zkoumali, zdali se u navržené heuristiky 5.2.1 projeví nějaký K -fázový přechod, tedy zdali pro formule s poměrem počtu klauzulí k počtu proměnných existuje interval $\langle x_r, y_r \rangle$ takový, že pro téměř všechny k -KNF formule s $k < x_r$ heuristika 5.2.1 nenalezne biklikové pokrytí a pro téměř všechny k -KNF formule s $k > y_r$ heuristika 5.2.1 nalezne biklikové pokrytí téměř vždy. V experimentech popsanych v sekci 6.3 jsme takovýto K -fázový přechod našli. Vzhledem k tomu, že se jedná o heuristiku, která není úplná, je tento K -fázový přechod jen přechodem, jehož dolní i horní hranice budou větší než hranice skutečného K -fázového přechodu.

Experimenty ze sekce 6.3 bychom mohli rozšířit o experimenty, které by omezovaly velikost první partity pro bikliky z biklikového pokrytí třemi a více, což jsme v práci neprovedli vzhledem k časové náročnosti experimentu. Tím bychom mohli testovat i formule s větším poměrem klauzulí k proměnným.

Dále by bylo možné pokusit se vytvořit dolní odhad K -fázového přechodu biklikového pokrytí pro k -KNF formule s poměrem klauzulí a proměnných r obdobně, jako John Franco a Allen Van Gelder dokázali pro fázový přechod k -KNF formulí [21]. V této diplomové práci jsme se vytvořením odhadu K -fázového přechodu biklikového pokrytí pro k -KNF nezabývali, protože by to bylo kombinatoricky náročné a vzhledem k tomu, že experimenty ukázaly, že 0.64 je výrazně méně než pravděpodobná skutečné hodnoty prahu, zřejmě by tento dolní odhad nepřinesl nic zajímavého.

Posledním experimentem, jehož výsledky jsme prezentovali v sekci 6.4, byl experiment, v němž jsme sledovali posun fázového přechodu matched formulí, pokud v pokrytí budeme uvažovat omezené bikliky. V experimentu se ukázalo, že pro k -KNF formule, $3 \leq k \leq 5$ jejichž počet klauzulí je maximálně takový

jako počet proměnných, zůstal fázový přechod $\langle x_r, y_r \rangle$ stejný. Pokud ale uvážíme k -KNF, s $k > 5$, fázový přechod matched formulí zůstává na 100%, zatímco pro biklikové pokrytí se posouvá k mnohem vyšším hodnotám. Pokud je tedy formule s poměrem počtu klauzulí a proměnných 120% v 9-KNF, najde naše heuristika téměř jistě biklikové pokrytí.

Použití heuristiky na hledání biklikového pokrytí neposunulo fázový přechod pro k -KNF formule s $k \leq 5$. Pro vyšší $k > 5$ jsme ukázali, že se tento fázový posun posouvá k mnohem vyšším hodnotám. Využitím heuristiky jsme tedy schopni rozpoznat biklikově splnitelné k -KNF formule, s $k = 7$ jejichž poměr počtu proměnných a klauzulí je menší než 115%. Pro $k = 9$ je dolní hranice fázového přechodu dokonce 120% a pro vyšší k se zvyšuje i tato hranice x_k .

Literatura

- [1] Sat competition data. <http://www.satcompetition.org/2014/downloads.shtml>. Dostupné: 19. srpna 2015.
- [2] Ron Aharoni and Nathan Linial. Minimal non-two-colorable hypergraphs and minimal unsatisfiable formulas. *Journal of Combinatorial Theory, Series A*, 43(2):196 – 204, 1986.
- [3] D. Bein, W. Bein, Z. Meng, and L. Morales. Clustering and the biclique partition problem.
- [4] A. Biere, M. Heule, H. van Maaren, and T. Walsh. *Handbook of Satisfiability*, volume 185 of *Frontiers in Artificial Intelligence and Applications*. IOS Press, Amsterdam, The Netherlands, The Netherlands, 2009.
- [5] Hans Kleine Büning and Zhao Xishun. Satisfiable formulas closed under replacement. *Electronic Notes in Discrete Mathematics*, 9(0):48 – 58, 2001. {LICS} 2001 Workshop on Theory and Applications of Satisfiability Testing (SAT 2001).
- [6] Endre Boros, Peter L. Hammer, and Xiaorong Sun. Recognition of q-horn formulae in linear time. *Discrete Applied Mathematics*, 55(1):1 – 13, 1994.
- [7] Yacine Boufkhad, Olivier Dubois, Yannet Interian, and Bart Selman. Regular random k-sat: Properties of balanced formulas. *J. Autom. Reasoning*, 35(1-3):181–200, 2005.
- [8] Stephen A. Cook. The complexity of theorem-proving procedures. In *Proceedings of the Third Annual ACM Symposium on Theory of Computing*, STOC '71, pages 151–158, New York, NY, USA, 1971. ACM.
- [9] Y. Crama and P.L. Hammer. *Boolean Functions: Theory, Algorithms, and Applications*. Encyclopedia of Mathematics and Its Applications. Cambridge University Press, 2011.
- [10] James M. Crawford and Larry D. Auton. Experimental results on the crossover point in satisfiability problems. In *In Proceedings of the Eleventh National Conference on Artificial Intelligence*, pages 21–27, 1993.
- [11] Martin Davis, George Logemann, and Donald Loveland. A machine program for theorem-proving. *Commun. ACM*, 5(7):394–397, July 1962.
- [12] Martin Davis and Hilary Putnam. A computing procedure for quantification theory. *J. ACM*, 7(3):201–215, July 1960.
- [13] Reinhard Diestel. *Graph Theory*. Number 173 in Graduate Texts in Mathematics. Springer, 1997.
- [14] Yefim Dinitz. Theoretical computer science. chapter Dinitz' Algorithm: The Original Version and Even's Version, pages 218–240. Springer-Verlag, Berlin, Heidelberg, 2006.

- [15] William F. Dowling and Jean H. Gallier. Linear-time algorithms for testing the satisfiability of propositional horn formulae. *The Journal of Logic Programming*, 1(3):267 – 284, 1984.
- [16] Rodney G. Downey and M. R. Fellows. *Parameterized Complexity*. Springer Publishing Company, Incorporated, 2012.
- [17] S. Even, A. Itai, and A. Shamir. On the complexity of time table and multi-commodity flow problems. In *Foundations of Computer Science, 1975., 16th Annual Symposium on*, pages 184–193, Oct 1975.
- [18] Herbert Fleischner, Oliver Kullmann, and Stefan Szeider. Polynomial-time recognition of minimal unsatisfiable formulas with fixed clause-variable difference. *Theoretical Computer Science*, 289(1):503 – 516, 2002.
- [19] Herbert Fleischner, Egbert Mujuni, Daniel Paulusma, and Stefan Szeider. Covering graphs with few complete bipartite subgraphs. In V. Arvind and Sanjiva Prasad, editors, *FSTTCS 2007: Foundations of Software Technology and Theoretical Computer Science*, volume 4855 of *Lecture Notes in Computer Science*, pages 340–351. Springer Berlin Heidelberg, 2007.
- [20] J. Flum and M. Grohe. *Parameterized Complexity Theory (Texts in Theoretical Computer Science. An EATCS Series)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.
- [21] John Franco and Allen Van Gelder. A perspective on certain polynomial-time solvable classes of satisfiability. *Discrete Applied Mathematics*, 125(2–3):177 – 214, 2003.
- [22] Michael R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., New York, NY, USA, 1979.
- [23] Ian P. Gent and Toby Walsh. The sat phase transition. pages 105–109. John Wiley and Sons, 1994.
- [24] Jiong Guo and Rolf Niedermeier. Invitation to data reduction and problem kernelization. *SIGACT News*, 38(1):31–45, March 2007.
- [25] P. Hall. On representatives of subsets. *Journal of the London Mathematical Society*, s1-10(1):26–30, 1935.
- [26] M.H. Heydari, C.O. Shields Jr., L. Morales, and I.H. Sudborough. Computing cross associations for attack graphs and other applications. 2007. cited By 0.
- [27] John E Hopcroft and Richard M Karp. An $n^{5/2}$ algorithm for maximum matchings in bipartite graphs. *SIAM Journal on computing*, 2(4):225–231, 1973.
- [28] Harry R. Lewis. Renaming a set of clauses as a horn set. *J. ACM*, 25(1):134–135, January 1978.

- [29] L. Lovasz. *Matching Theory (North-Holland Mathematics Studies)*. Elsevier Science Ltd., Oxford, UK, UK, 1986.
- [30] Marc Mézard and Riccardo Zecchina. Random k -satisfiability problem: From an analytic solution to an efficient algorithm. *Phys. Rev. E*, 66:056126, Nov 2002.
- [31] Christos M. Papadimitriou. *Computational complexity*. Addison-Wesley, Reading, Massachusetts, 1994.
- [32] John S. Schlipf, Fred S. Annexstein, John V. Franco, and R.P. Swaminathan. On finding solutions for extended horn formulas. *Information Processing Letters*, 54(3):133 – 137, 1995.
- [33] Larry J. Stockmeyer. The polynomial-time hierarchy. *Theoretical Computer Science*, 3(1):1 – 22, 1976.
- [34] Stefan Szeider. Minimal unsatisfiable formulas with bounded clause-variable difference are fixed-parameter tractable. *Journal of Computer and System Sciences*, 69(4):656–674, 2004.
- [35] Stefan Szeider. Generalizations of matched CNF formulas. *Ann. Math. Artif. Intell.*, 43(1):223–238, 2005.
- [36] Craig A. Tovey. A simplified np-complete satisfiability problem. *Discrete Applied Mathematics*, 8(1):85 – 89, 1984.

Seznam použitých zkratek

KNF konjunktivně normální forma

SAT splnitelnost (satisfiability)

8. Přílohy

8.1 Uživatelská dokumentace

Pro replikaci experimentů je možné využít programů spolu s Makefilem přiloženým na CD. Ke spuštění je potřeba Microsoft Visual Studio 2012 nebo g++ překladač verze alespoň 5.1.1. U nižších verzí není zaručena funkčnost.

Jak spustit experimenty popíšeme pouze pro g++ pod operačním systémem Linux.

- (1) Zkopírujte si složku na pevný disk,
- (2) ve složce program spuštěním příkazu `make` spustíte všechny testy pro párování i pro biklikové pokrytí,
- (3) na standartním výstupu je výpis, který slouží k analýze a na chybovém výstupu jsou vypsány výsledky ve srozumitelné podobě.

Makefile obsahuje i další cíle. Cíle spustíme tak, že ve složce program spustíme příkaz `make cíl`, kde *cíl* je jeden z následujících.

Cíl `run_tests` je defaultní cíl, zkompiluje a spustí všechny testy,

`matching_tests` zkompiluje program k testům `matched` formulí,

`biclique_cover_tests` zkompiluje program k testům biklikového pokrytí,

`run_matching_tests` zkompiluje a spustí experiment `matched` formulí,

`run_biclique_cover_tests` zkompiluje a spustí experiment biklikového pokrytí,

`clear` vyčistí složku od všech souborů, které vznikly některým z cílů.

8.2 Hardware a software použitý pro experimenty

Experimenty proběhli na počítači s procesorem Intel i5 3500 a s 8GB 1333Hz paměti. Strojový čas, který zabraly všechny experimenty, které jsou zde popsány v kapitole 6, je více než týden.

Software použitý pro zhotovení a vyhodnocení experimentů je

Linux F22 kernel 4.0

g++ 5.1.1

gnuplot

Doxygen

Texlive

Microsoft Windows 8

Microsoft Visual Studio 2012

LibreCalc

8.3 Obsah CD

Složka `program` obsahuje program, který jsme použili k vyhodnocení všech experimentů použitých v diplomové práci.

Složka `vysledky` obsahuje `.csv` soubory, v nichž jsou všechny výsledky experimentů, které jsou prezentovány v diplomové práci.

Složka `texty` obsahuje `.pdf` soubory s diplomovou prací a s technickou dokumentací k programu ze složky `program`.