

# Domácí úloha 1

Experimentální vyhodnocení algoritmů gsat a probSAT

Leoš Tobolka

<b>Souhrn</b>	<b>3</b>
<b>Úvod – Specifikace zadání</b>	<b>3</b>
<b>Materiály</b>	<b>3</b>
Popis algoritmů	3
Vstupní data	4
Metriky	4
Provedení experimentu	4
<b>Výsledky – naměřená data</b>	<b>5</b>
Měření na sadách instancí	5
Sada uf20-91R	6
300 iterací, 1 opakování	7
300 iterací, 5 opakování	8
Sada uf50-218R	9
500 iterací, 10 opakování	10
3000 iterací, 3 opakování	11
Sada uf75-325	12
500 iterací, 10 opakování	13
5000 iterací, 3 opakování	14
<b>Diskuze</b>	<b>15</b>
Interpretace dat	15
Data tabulek	15
Grafy	16
Celkové vyhodnocení	16

# Souhrn

V tomto experimentu byly porovnány algoritmy GSAT a probSAT na instancích problému v rozsahu 20 – 75 proměnných a 75 – 315 klauzulí. Z naměřených dat z běhů algoritmů na sadách instancí byly vybrány základní metriky, jako je procentuální úspěšnost, počty kroků, počty kroků s důrazem na úspěšnost a podobně.

Po srovnání těchto naměřených metrik mezi algoritmy, vyšlo najevo, že algoritmus probSAT je ve všech měřených případech lepší než algoritmus GSAT.

## Úvod – Specifikace zadání

Jsou dány algoritmy GSAT a probSAT. Algoritmy slouží ke konstrukci řešení booleovských formulí v konjunktivní normální formě (SAT).

Úkolem je experimentálně porovnat tyto dva algoritmy na sadě instancí a určit, který algoritmus v menším počtu iterací dospěje k řešení 3-SAT instancí v rozsahu 20 – 75 proměnných.

Celé zadání lze najít na stránce

<https://courses.fit.cvut.cz/NI-KOP/homeworks/files/task1.html>.

## Materiály

### Popis algoritmů

Implementaci GSAT algoritmu jsem využil ze studijních materiálů

<https://courses.fit.cvut.cz/NI-KOP/download/index.html>. Algoritmus je dle zadání spouštěn s pevnými parametry  $p = 0.4$ . Parametr  $p$  určuje pravděpodobnost, že se v dané iteraci udělá náhodný krok.

Implementaci probSAT algoritmu jsem získal úpravou použitého GSAT algoritmu. Tento algoritmus v každé iteraci vybere z náhodné nesplněné klauzule náhodný literál. Pravděpodobnost výběru literálu se určuje na základě počtu nově splněných či nesplněných klauzulí po přehození hodnoty literálu. Algoritmus je spouštěn dle zadání s pevnými parametry  $cm = 0$  a  $cb = 2.3$ . Parametr  $cm$  ovlivňuje důležitost

počtu nově splněných klauzulí po přehození hodnoty literálu. Parametr *cb* naopak ovlivňuje důležitost počtu nově nesplněných klauzulí po přehození hodnoty literálu. Hodnota epsilon, která se přičítá k počtu nově nesplněných klauzulí po přehození hodnoty literálu, aby se zamezilo případné dělení nulou, je pevně nastavena na hodnotu 1.

## Vstupní data

Instance 3-SAT problému pro měření jsem získal ze studijních materiálů na stránce <https://courses.fit.cvut.cz/NI-KOP/download/index.html>. Jedná se o splnitelné instance z knihovny SATLIB. Data jsou v rozsahu 20 – 75 proměnných a 71 – 325 klauzulí. Konkrétně se jedná o sady uf20-71R, uf20-81R, uf20-91R, uf50-200R a uf75-325. Každá sada obsahuje 100 instancí.

## Metriky

Měřená metrika je počet kroků algoritmů, která je nezávislá na platformě.

Z naměřených dat jsem pro porovnání algoritmů využil následující:

- Průměrný počet kroků
- Průměrný počet kroků s penalizací při neúspěchu
- Procentuální úspěšnost
- Počet splněných instancí
- Počet nesplněných instancí

## Provedení experimentu

Oba algoritmy jsem pro každou sadu instancí spustil několikrát s různými vstupními parametry. Vstupní parametry jsem zvolil následující:

- -i 300 -T 1
- -i 300 -T 5
- -i 300 -T 10
- -i 300 -T 20
- -i 500 -T 1
- -i 500 -T 3
- -i 500 -T 5
- -i 500 -T 10
- -i 1000 -T 1
- -i 1000 -T 5
- -i 1000 -T 10

- -i 1000 -T 20
- -i 3000 -T 1
- -i 3000 -T 3
- -i 3000 -T 5
- -i 3000 -T 10
- -i 5000 -T 1
- -i 5000 -T 3
- -i 5000 -T 10

Výběr parametrů proběhl na základě zkušeností ze školních cvičení, kde jsem získal základní představu, jaké rozmezí parametrů bude efektivní pro vyhodnocení měření. Rozsah maximálního počtu iterací pro jeden běh je tedy  $300 \times 1 = 300$  až  $5000 \times 10 = 50000$ .

Výstup tisíce běhů obou algoritmů pro každou instanci ze sady jsem zpracoval skriptem do csv formátu, v kterém už lze přehledněji porovnat oba algoritmy pro jednu sadu a jednu kombinaci parametrů.

Z csv souborů vznikly grafy a tabulky. Všechny grafy lze najít v příloze tohoto dokumentu. V následující kapitole je jen malý výběr běhů na sadách instancí.

## Výsledky – naměřená data

Všechny grafy lze najít v příloze. Zde, pro tuto kapitolu, jsem vybral jen takové případy běhů s takovými parametry, které odpovídají těžkosti sad instancí.

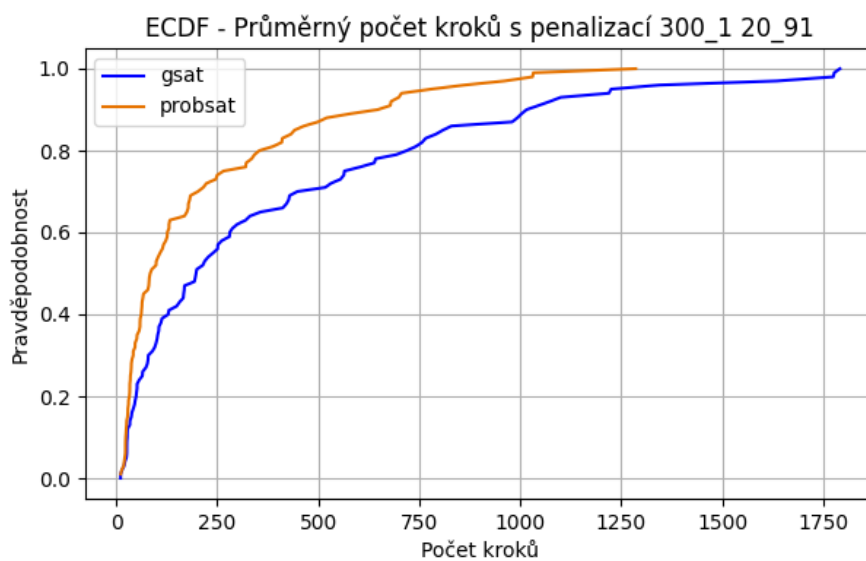
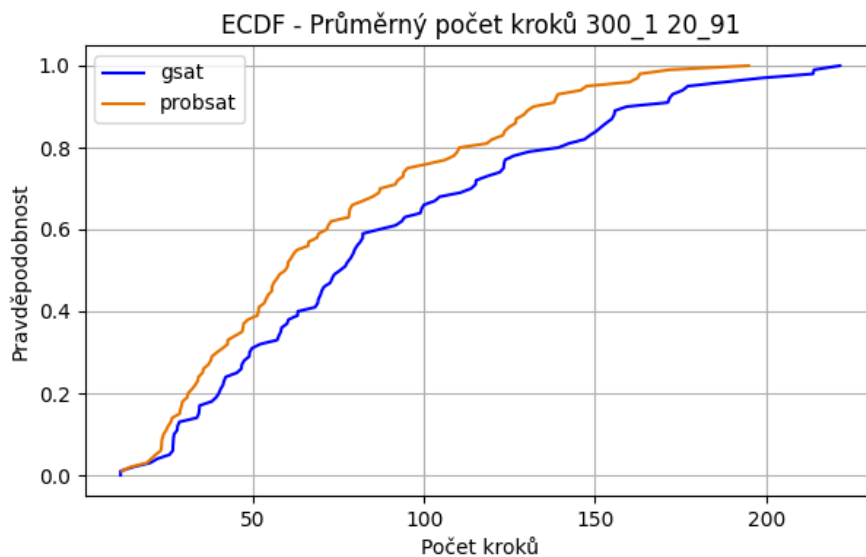
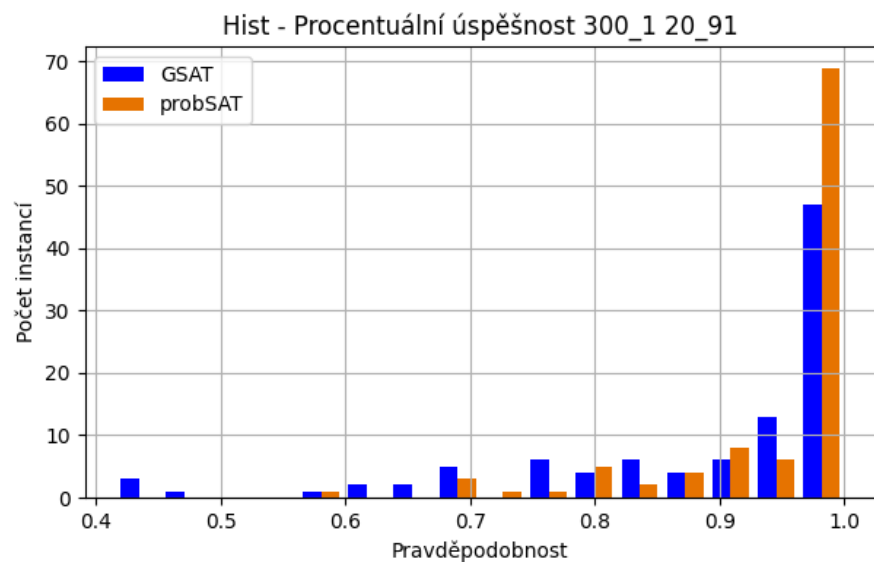
### Měření na sadách instancí

## Sada uf20-91R

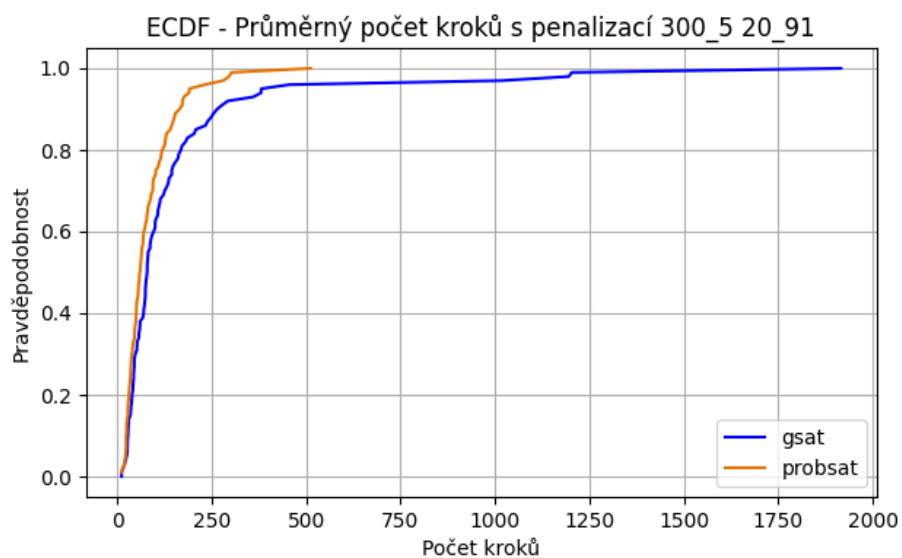
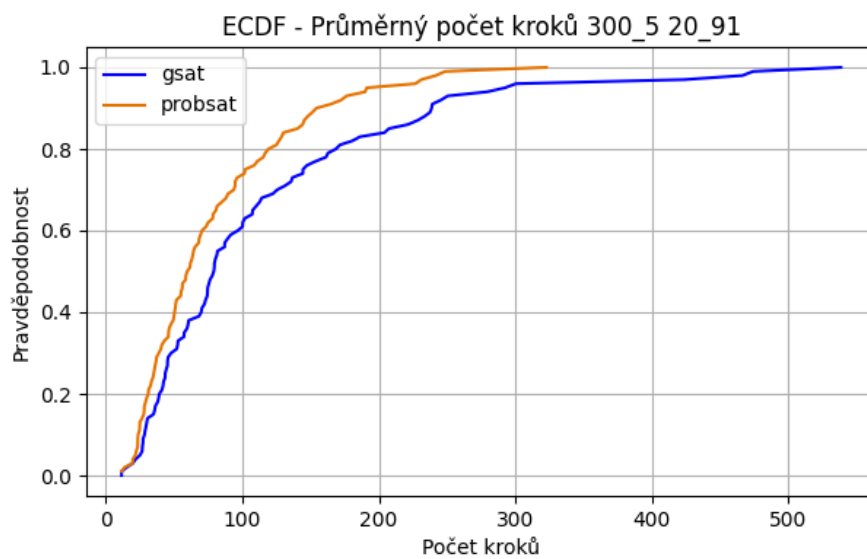
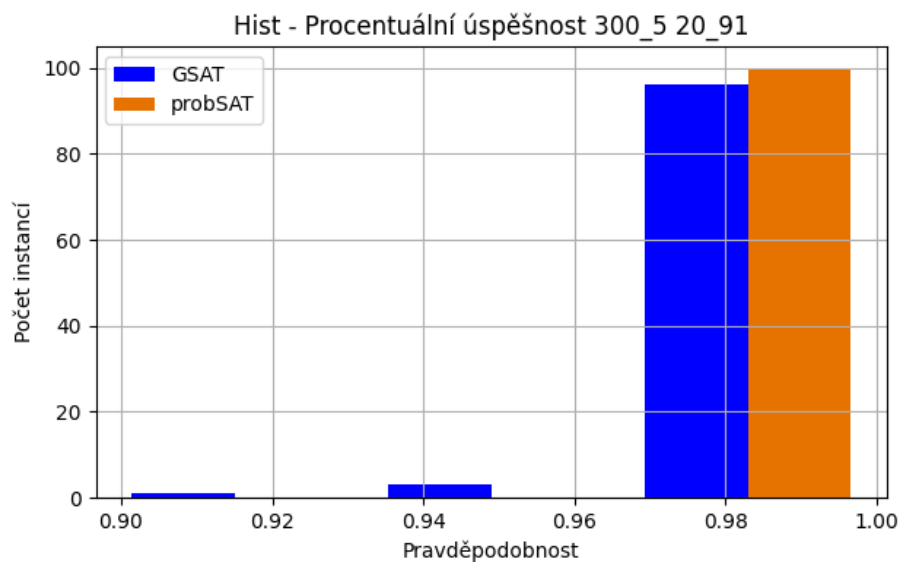
100 instancí - 20 proměnných, 91 klauzulí

uf20-91	Celkem kroků	Průměrný počet kroků	Celkem vážený počet kroků	Průměr vážený počet kroků	Celkem splněno instancí	Celkem NEsplněn o instancí	% úspěšnost
GSAT -i 300 -T 1	8780	88	38890	389	44424	5576	0.89
probSAT -i 300 -T 1	7122	71	21292	213	47376	2624	<b>0.95</b>
GSAT -i 300 -T 5	11414	114	15464	155	49850	150	1.0
probSAT -i 300 -T 5	7945	79	8323	83	49986	14	1.0
GSAT -i 300 -T 10	11484	115	12348	123	49984	16	1.0
probSAT -i 300 -T 10	8020	80	8020	80	50000	0	1.0
GSAT -i 500 -T 1	10369	104	36352	364	47113	2887	0.94
probSAT -i 500 -T 1	7743	77	16221	162	49058	942	<b>0.98</b>
GSAT -i 500 -T 3	12003	120	18672	187	49753	247	1.0
probSAT -i 500 -T 3	8191	82	8596	86	49985	15	1.0
GSAT -i 500 -T 5	12146	121	13991	140	49959	41	1.0
probSAT -i 500 -T 5	8099	81	8099	81	50000	0	1.0
GSAT -i 500 -T 10	12501	125	12681	127	49998	2	1.0
probSAT -i 500 -T 10	8110	81	8110	81	50000	0	1.0
GSAT -i 1000 -T 1	12104	121	29060	291	49058	942	0.98
probSAT -i 1000 -T 1	8159	82	10787	108	49854	146	<b>1.0</b>
GSAT -i 1000 -T 5	12997	130	13177	132	49998	2	1.0
probSAT -i 1000 -T 5	8235	82	8235	82	50000	0	1.0

300 iterací, 1 opakování



300 iterací, 5 opakování



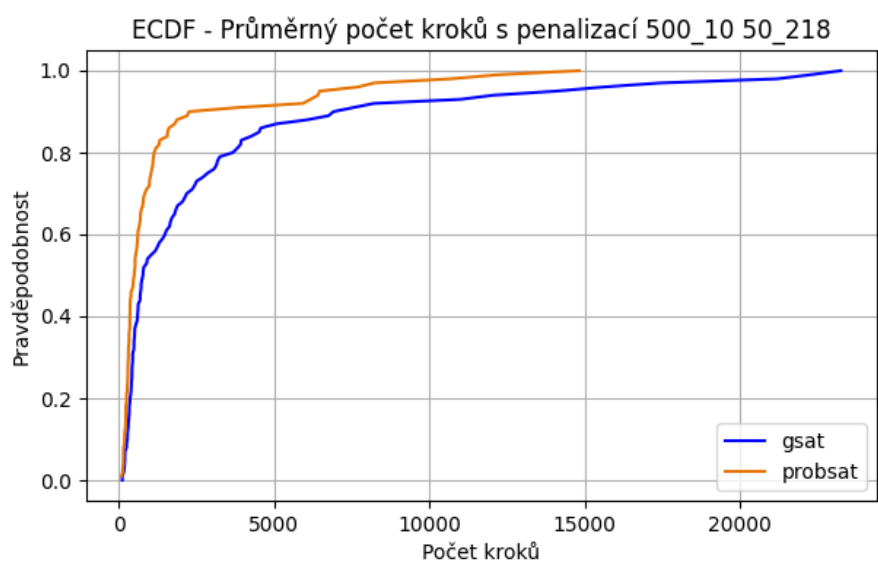
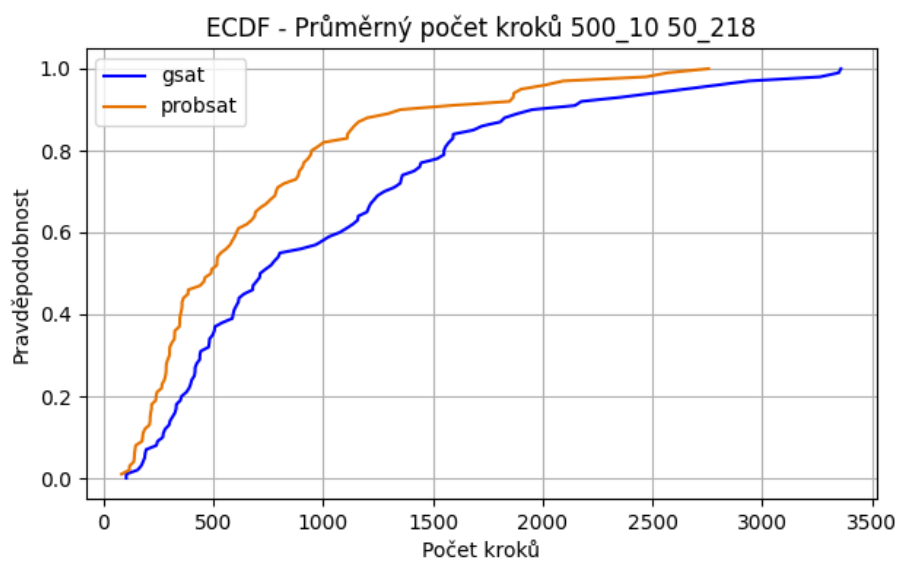
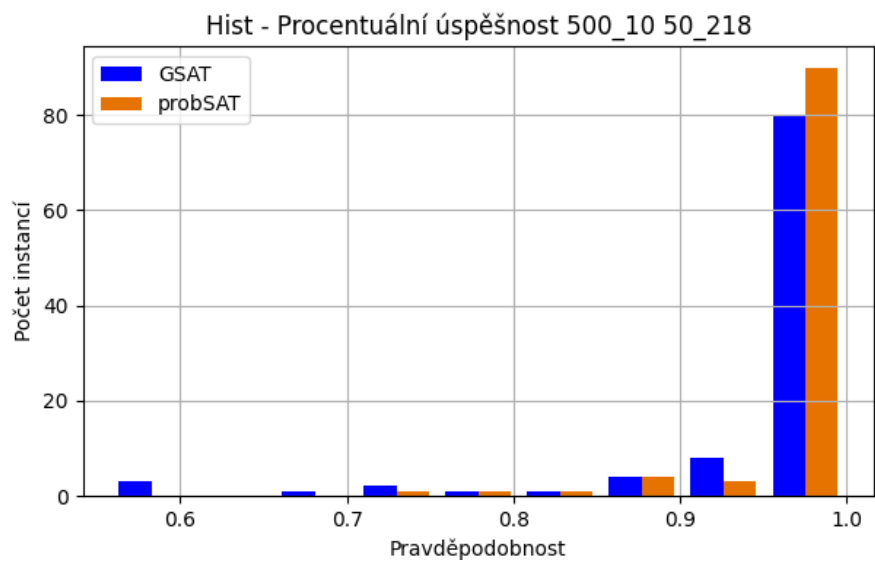


## Sada uf50-218R

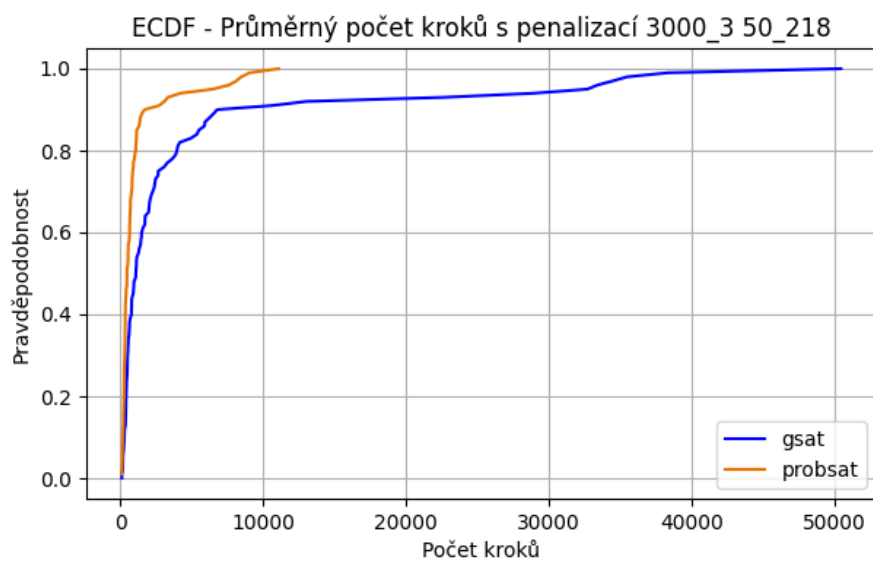
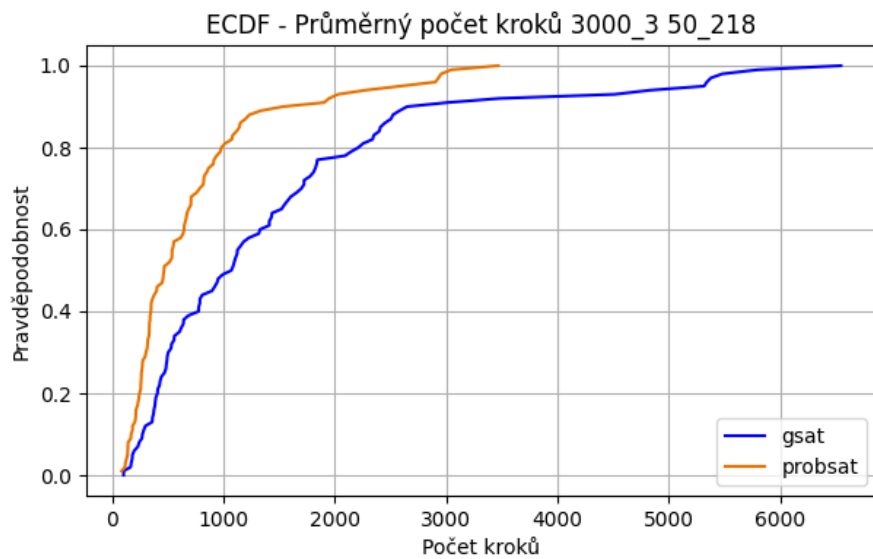
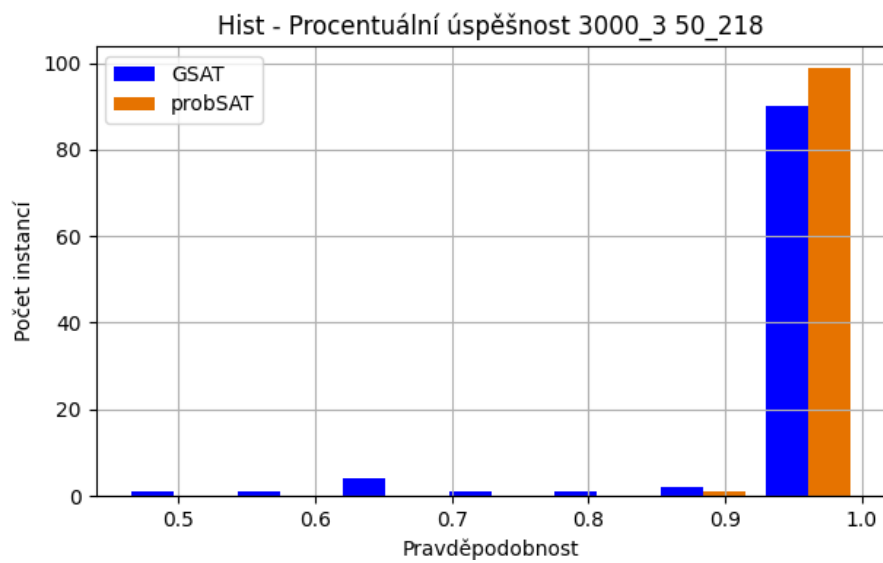
100 instancí - 50 proměnných, 218 klauzulí

uf50-218R	Celkem kroků	Průměrný počet kroků	Celkem vážený počet kroků	Průměr vážený počet kroků	Celkem splněno instancí	Celkem Nesplněno instancí	% úspěšnost
GSAT -i 300 -T 5	64994	650	329837	3298	40191	9809	0.8
probSAT -i 300 -T 5	52681	527	214897	2149	43992	6008	<b>0.88</b>
GSAT -i 300 -T 10	83885	839	301775	3018	45965	4035	0.92
probSAT -i 300 -T 10	63356	634	175946	1759	47915	2085	<b>0.96</b>
GSAT -i 500 -T 1	34552	346	261253	2613	24811	25189	0.5
probSAT -i 500 -T 1	30699	307	193383	1934	31924	18076	<b>0.64</b>
GSAT -i 500 -T 5	83105	831	353960	3540	43981	6019	0.88
probSAT -i 500 -T 5	60530	605	189365	1894	47137	2863	<b>0.94</b>
GSAT -i 1000 -T 1	55711	557	376867	3769	32158	17842	0.64
probSAT -i 1000 -T 1	44044	440	221128	2211	40162	9838	<b>0.8</b>
GSAT -i 1000 -T 5	110948	1109	360158	3602	47231	2769	0.94
probSAT -i 1000 -T 5	68827	688	143977	1440	49165	835	<b>0.98</b>
GSAT -i 3000 -T 1	102612	1026	535314	5353	41987	8013	0.84
probSAT -i 3000 -T 1	63665	637	192671	1927	47611	2389	<b>0.95</b>
GSAT -i 3000 -T 3	145614	1456	443532	4435	48161	1839	0.96
probSAT -i 3000 -T 3	73137	731	112341	1123	49758	242	<b>1.0</b>
GSAT -i 5000 -T 1	127320	1273	562290	5623	45167	4833	0.9
probSAT -i 5000 -T 1	69675	697	158235	1582	49016	984	<b>0.98</b>

500 iterací, 10 opakování



3000 iterací, 3 opakování

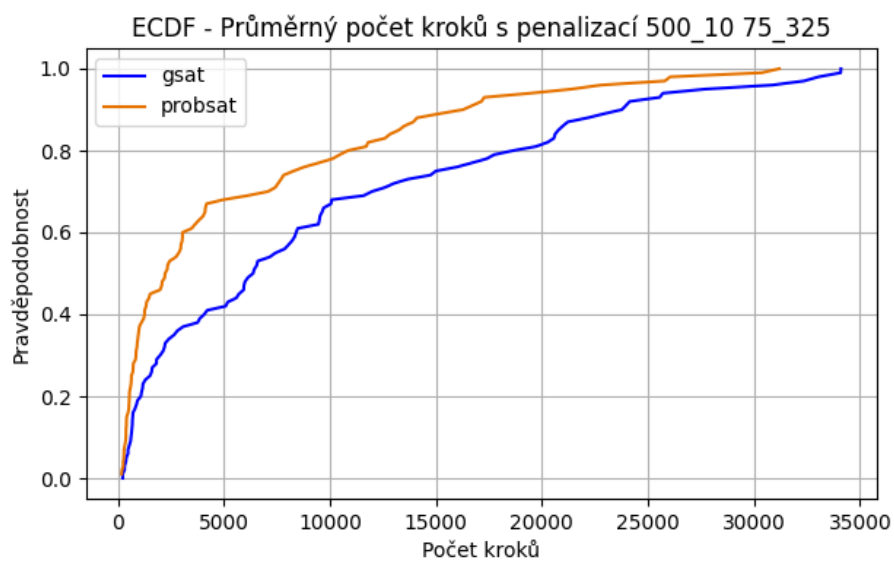
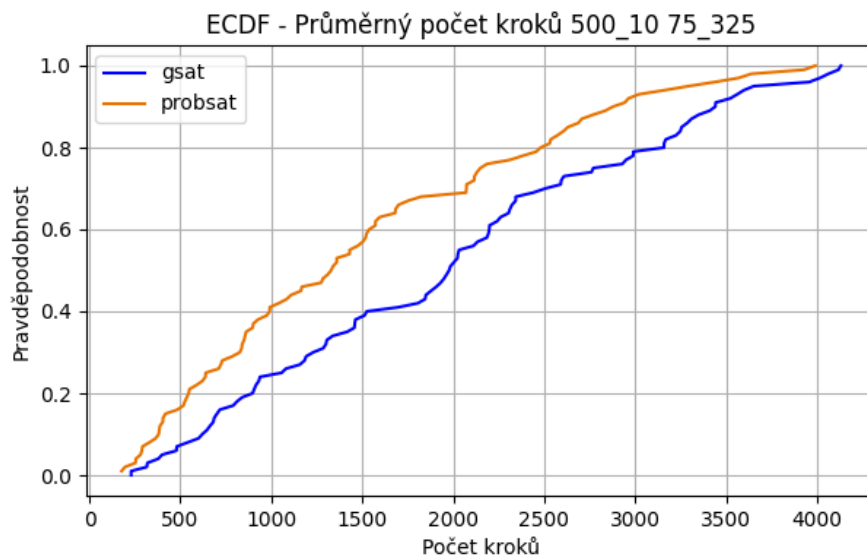
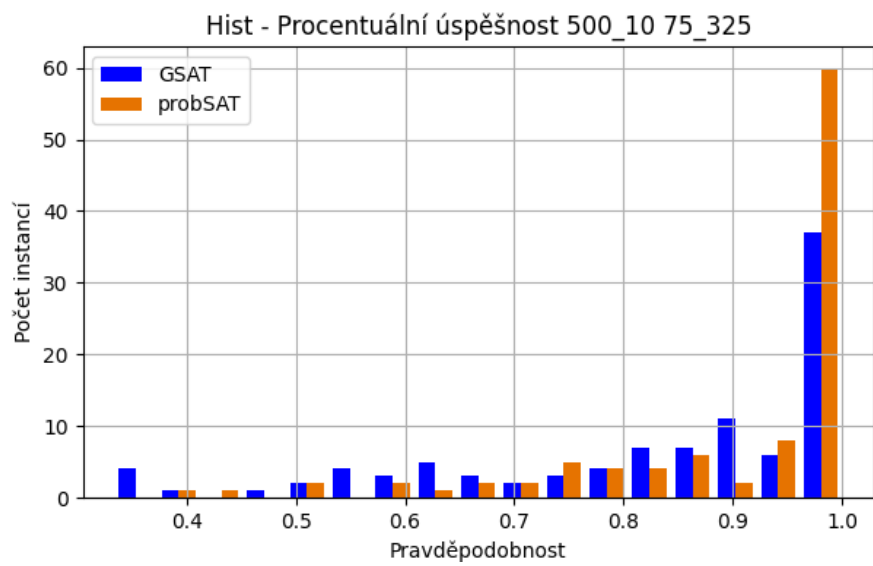


## Sada uf75-325

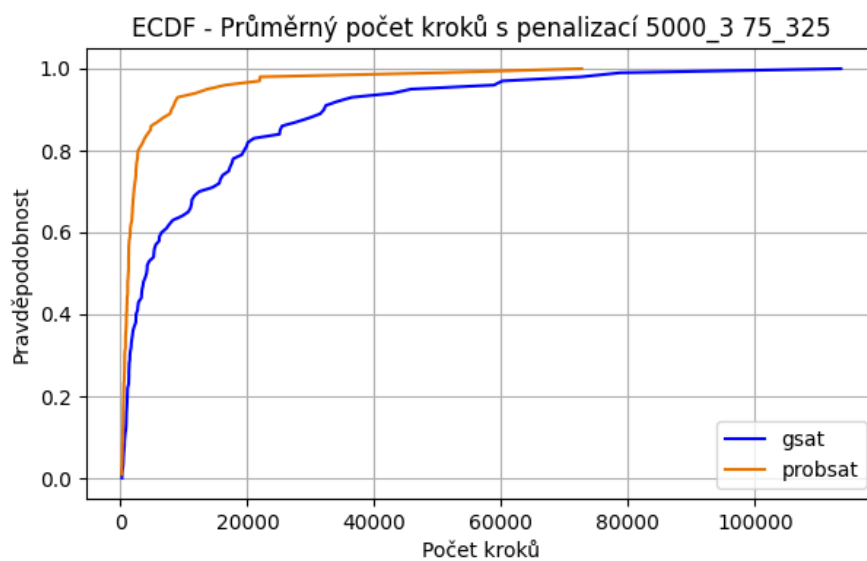
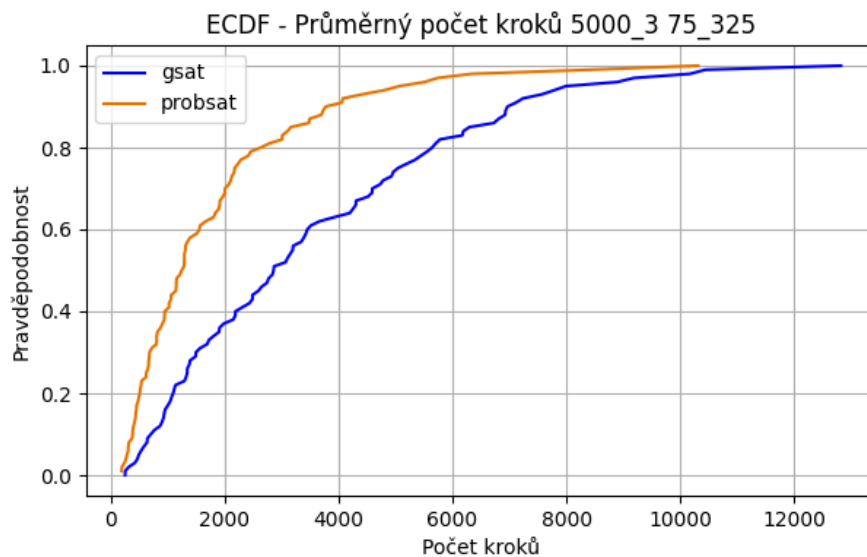
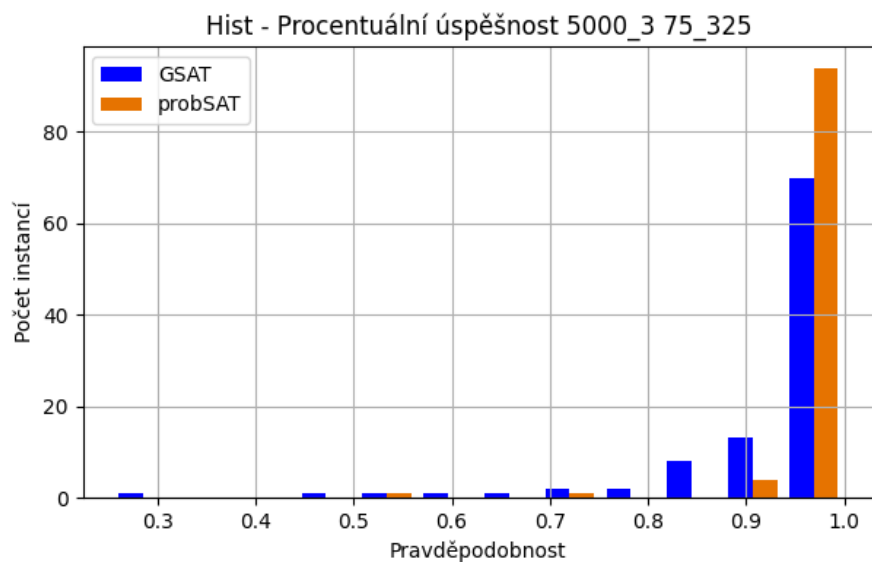
100 instancí - 75 proměnných, 325 klauzulí

uf75_325	Celkem kroků	Průměrný počet kroků	Celkem vážený počet kroků	Průměr vážený počet kroků	Celkem splněno instancí	Celkem Nesplněno instancí	% úspěšnost
GSAT -i 500 -T 5	140337	1403	870732	8707	33769	16231	0.68
probSAT -i 500 -T 5	114349	1143	595399	5954	39310	10690	<b>0.79</b>
GSAT -i 500 -T 10	198653	1987	947813	9478	41676	8324	0.83
probSAT -i 500 -T 10	150017	1500	567257	5673	45364	4636	<b>0.91</b>
GSAT -i 1000 -T 1	75607	756	604627	6046	20610	29390	0.41
probSAT -i 1000 -T 1	65990	660	445106	4451	28938	21062	<b>0.58</b>
GSAT -i 1000 -T 5	204253	2043	994003	9940	41225	8775	0.82
probSAT -i 1000 -T 5	143186	1432	518306	5183	45832	4168	<b>0.92</b>
GSAT -i 1000 -T 10	257235	2572	838995	8390	46768	3232	0.94
probSAT -i 1000 -T 10	167472	1675	394632	3946	48738	1262	<b>0.97</b>
GSAT -i 3000 -T 1	166128	1661	1143744	11437	31896	18104	0.64
probSAT -i 3000 -T 1	118345	1183	568975	5690	41655	8345	<b>0.83</b>
GSAT -i 3000 -T 3	283105	2831	1205371	12054	44307	5693	0.89
probSAT -i 3000 -T 3	165743	1657	443249	4432	48287	1713	<b>0.97</b>
GSAT -i 5000 -T 1	227884	2279	1406884	14069	36900	13100	0.74
probSAT -i 5000 -T 1	143342	1433	544472	5445	45543	4457	<b>0.91</b>
GSAT -i 5000 -T 3	352323	3523	1239543	12395	46714	3286	0.93
probSAT -i 5000 -T 3	179710	1797	378160	3782	49265	735	<b>0.99</b>

500 iterací, 10 opakování



5000 iterací, 3 opakování



# Diskuze

## Interpretace dat

### Data tabulek

Z tabulek pro každou sadu lze vidět následující metriky:

- Celkem kroků
- Průměrný počet kroků
- Celkem vážený počet kroků
- Průměr vážený počet kroků
- Celkem splněno instancí
- Celkem NESplněno instancí
- % úspěšnost

Metriky vychází ze všech běhů algoritmu s danými parametry pro danou sadu. Měnicí se parametry  $-i$  a  $-T$  určují maximální počet iterací vnitřního cyklu a maximální počet restartů.

V každé tabulce, pokud se zaměříme na dvojice řádků kde oba algoritmy byly spuštěny se stejnými parametry na stejné sadě, uvidíme výrazný rozdíl v jejich metrikách.

U méně těžších instancí, zejména v sadách s 20 literály, měli v některých případech algoritmy stejnou procentuální úspěšnost. To je zapříčiněno z toho důvodu, že v některých bězích byly parametry algoritmů nastaveny zbytečně vysoko. Proto i GSAT, který zde hodnotím jako horší, zvládl vyřešit všechny instance, stejně jako probSAT. Pokud se ale vezmou v potaz i ostatní metriky, tak i přes svoji stejnou úspěšnost, jeho ostatní metriky jsou oproti druhému algoritmu pozadu.

U těžších instancí, např. s 50 a 75 literály, už je výrazněji vidět rozdíl i v úspěšnosti algoritmů.

Z naměřených dat v tabulkách je vidět, že algoritmus probSAT je lepší než algoritmus GSAT. Statistiky běhů probSATu pro každou sadu instancí jsou ve všech případech lepší, nebo stejně dobré oproti GSATU.

## Grafy

Na grafech lze vidět empirické distribuční funkce průměrných počtů iterací a vážených počtu iterací algoritmů. Na histogramu lze vidět procentuální úspěšnost tisíce běhů na všech instancích sady.

Z grafů průměrných počtů iterací lze vypožorovat že probSATu průměrně stačí menší počet iterací pro nalezení řešení instance. Jeho křivka je vždy nad křivkou GSATu. Křivky se neprotínají, tudíž i v tomto kritériu hodnocení algoritmus probSAT je lepší než GSAT.

Distribuční funkce váženého počtu iterací, kde neúspěšný běh se počítá jako 10 násobek maximálního počtu iterací, také lépe vychází pro probSAT. Křivka tohoto algoritmu je také ve všech případech výše než GSATu. Křivka probSATu roste ze začátku rychleji, ale i potom co se jeho růst zpomalí, tak křivka zůstává nad druhou, což naznačuje že jeho počet neúspěšných běhů je menší a stále dosáhne řešení rychleji.

## Celkové vyhodnocení

Ze všech naměřených metrik, jak z tabulek či grafů, je vidět že algoritmus probSAT je v tomto experimentu lepší než algoritmus GSAT.

Zřejmě by šlo změřit více metrik z běhů algoritmů, které by mohly vypovědět další vlastnosti algoritmů. Tím je myšleno, že pravděpodobně existují i kritéria pro které je algoritmus GSAT lepší, ale ve valné většině, jak je z experimentu vidět, algoritmus probSAT je efektivnější a pro běžné využití, zejména na těžších instancích najde řešení v menším počtu kroků spolehlivěji.