KOP – Úloha 1. – Experimentální vyhodnocení algoritmů (gsat2 × probSAT)

Radek Cichra

October 30, 2024

1 Úvod

Tento ‘report‘ porovnává a hodnotí dvě různé implementace 3-SAT řešících algoritmů – **gsat2** a **probSAT**. Výstup vychází z výsledků měření na šesti různých datasetech 3-SAT formulí dostupných na stránkách předmětu. Formule jsou v rozsahu 20 až 75 proměnných a s poměrem klauzulí ku počtu proměnných ≥ 4.3 kromě `ruf75-320` (= jsou považovány za obtížné). Stěžejní metrikou je *počet kroků* algoritmu potřebných k nalezení řešení dané 3-SAT formule. Sledování této proměnné (oproti například měření času běhu) nám dovoluje abstrahovat a ignorovat jevy spjaté s implementací algoritmů / hardware konfigurací a platformou použitou k měření.

Předpokládaný výstup je potvrzení obecně převládajícího názoru, že probSAT algoritmus by měl být lepší.

1.1 vymezení (outline of the study)

V rámci úkolu byly algoritmy **gsat2** a **probSAT** testované na šesti setech formulí (1000 formulí/set):

1. `ruf20-91`
2. `uf20-91`
3. `ruf36-157`
4. `ruf50-218`
5. `uf50-218`

6. `ruf75-320`

kde dvojčíslí v názvu značí *#proměnných-#klauzulí*. Každý algoritmus byl na každou formuli spuštěn 1000krát, tedy pro každý set algoritmus vygeneroval 10^6 výstupů. Výstupy algoritmů ze stejných setů byly zhodnoceny, upraveny a vzájemně porovnány, včetně vizualizace sekundárních metrik (Histogram a CDF).

2 Metody & Metriky

2.1 Algoritmy

Porovnáváme dva algoritmy – **gsat2** a **probSAT**. Algoritmus gsat2 byl přímo převzat z hodin předmětu. Oproti tomu pro probSAT bylo nutné upravit implementaci nalezenou na Githubu. Upravenou implementaci lze najít v souboru `probSAT.c`. Hlavní úpravy jsou:

1. změna výstupu tak, aby se shodoval s výstupem algoritmu gsat2
2. úprava náhodné generace tak, aby docházelo častěji ke změnám (předtím se generace měnila pouze $\approx 1/\text{s}$)
3. změna hodnoty proměnné `cb` pro 3-SAT instance na 2.3 (hodnota specifikovaná v zadání úkolu)

probSAT byl postaven příkazem `gcc probSAT.c -lm -o probsat`.

2.2 Formát výstupu

Formát výstupu obou algoritmů byl pro jednoduchost sjednocen. Jeden záznam (jeden běh na jedné formuli = jeden řádek výstupu) má formát

`[A];run [B];[C];[D]`

kde `[A]` je soubor (`.cnf`) s danou formulí, `[B]` je číslo běhu algoritmu na daném souboru, `[C]` je příznak (ne)úspěchu (`S` nebo `F`) a `[D]` je počet kroků algoritmu na konci běhu. Úspěchem se myslí nalezení řešení při daném krokovém omezení. Reálný příklad záznamu je například

`ruf20-91-782.cnf;run 5;S;33`

Vedle tohoto výstupu je ještě generován `.raw` soubor s více informacemi na pozici `[D]` (tento výstup ale není v kontextu úkolu využit).

2.3 Parametry algoritmů

Parametry algoritmů byly během měření (v souladu se zadáním) pevně určené a neměnné. Co se týče nastavení maximálních celkových opakování (`MAX-TRIES`) a maximálních 'flipů' jednotlivých proměnných (`MAX-FLIPS`) pro každou formuli, bylo zvoleno

$$\texttt{MAX-TRIES} = 1,\ \texttt{MAX-FLIPS} = 10000$$

Tato konfigurace vede k tomu, že vysoké procento běhů (85%+) skončí úspěchem.

(*Pozn:* `MAX-FLIPS` = 10000 může u některých setů formulí zhoršit kvalitu vizualizace, pokud bychom chtěli zobrazit celou škálu 0 – 10000, proto bylo toto rozhodnutí doprovázené omezením vykresleného intervalu s pomocí mezikvartilového rozptylu IQR - viz 2.5)

2.4 Metriky – Celková úspěšnost algoritmů

První metrika, dle které lze hodnotit a porovnat algoritmy, je množství (ne)úspěšných běhů na daném setu (Tabulky 1 a 2). Tato informace může být příznak toho, který algoritmus je spolehlivější s daným omezením pokusů a 'flipů'. Jelikož obtížnost setů je podobná (protože poměr klauzulí ku proměnným je zhruba stejný), stává se hlavním faktorem ovlivňujícím počet kroků algoritmu přímo počet proměnných → více proměnných značí (zpravidla) více kroků pro nalezení řešení.

Je také důležité zmínit, že spíše než samotné počty (ne)úspěchů mají informační hodnotu jejich poměry vzhledem k celkovému počtu běhů a k druhému algoritmu – mohli bychom totiž klidně nastavit podmínky, kde oba algoritmy drasticky selžou (velmi nízký maximální počet 'flipů'). Toto je také důvod pro zvolení vysokého stropu `MAX-FLIPS` – aby oba algoritmy fungovali do jisté míry 'spolehlivě' a šlo lépe rozhodnout, který je spolehlivější.

Hodnoty lze zjistit (a ověřit) z výstupních dat bez scriptů, například příkazy

```
cat $DATA | grep S | wc -l

cat $DATA | grep F | wc -l
```

Z tabulek 1 a 2 lze vyvodit, že v rozmezí 0 – 10000 kroků probSAT algoritmus vyřešil větší procento běhů. V nejhorším případě probSAT skončilo $\approx 3.6\%$ běhů na jednom setu (`ruf75-320`) neúspěchem. Oproti tomu u gsat2 je to (opět u setu `ruf75-320`) až $\approx 14.1\%$.

sada	úspěšných běhů	neúspěšných běhů
ruf20-91	999947 ≈ 100.0%	53 ≈ 0.0%
uf20-91	999987 ≈ 100.0%	13 ≈ 0.0%
ruf36-157	992010 ≈ 99.2%	7990 ≈ 0.8%
ruf50-218	961579 ≈ 96.2%	38421 ≈ 3.8%
uf50-218	962393 ≈ 96.2%	37607 ≈ 3.8%
ruf75-320	859468 ≈ 85.9%	140532 ≈ 14.1%

Table 1: Tabulka s počtem úspěšných a neúspěšných běhů pro gsat2

sada	úspěšných běhů	neúspěšných běhů
ruf20-91	999998≈ 100.0%	2 ≈ 0.0%
uf20-91	999997 ≈ 100.0%	3 ≈ 0.0%
ruf36-157	999186 ≈ 100.0%	814 ≈ 0.0%
ruf50-218	993148 ≈ 99.3%	6852 ≈ 0.7%
uf50-218	995623 ≈ 99.6%	4377 ≈ 0.4%
ruf75-320	963937 ≈ 96.4%	36063 ≈ 3.6%

Table 2: Tabulka s počtem úspěšných a neúspěšných běhů pro probSAT

2.5 Metriky – Histogram & CDF

Dvě významné vizualizační metriky jsou **Histogram** a **CDF** funkce:

- Histogram ukazuje počet běhů (osa y) zakončených v nějakém počtu kroků (osa x). Nabízí tak ilustraci toho, v jaké oblasti (v intervalu kolika kroků) se koncentruje většina řešení pro daný algoritmus na daném setu. Také ukazuje distribuci řešení v daných mezích (0 – `MAX-FLIPS`). Za *lepší* algoritmus bychom považovali ten, který má užší distribuci blíž k levé mezi grafu (0) → takový algoritmus je konzistentní a končí v menším počtu kroků.

- CDF ukazuje, jak velká část (osa y) z celkového počtu běhů skončila v daném nebo menším počtu kroků (osa x). CDF lze brát jako alternativu k Histogramu, kde je ilustrovaná 'rychlost' algoritmu – prudká křivka ukazuje, že v daném počtu kroků skončilo velké množství běhů. Za *lepší* bychom považovali algoritmus, který má prudší křivu a rychleji se blíží k maximu osy y – 1. Alternativně lze také sledovat plochu pod CDF → *lepší* algoritmus bude mít větší plochu, neb dříve dosáhne 1.

Pro vykreslení byly použity scripty `make-hist.py` a `make-cdf.py`. Pro vykreslení porovnání pak `make-merged-cdfs.py` a `make-merged-hists.py`. Byl využit modul `matplotlib`. Každý set formulí byl zpracován zvlášť.

V rámci úprav dat před vizualizací došlo k následujícím akcím:

1. **Z dat se odstranily neúspěšné běhy** – úspěšnost obecně byla popsaná v předchozí kapitole 2.4, zde byl cíl vytvořit vizualizaci vypovídající o úspěšných bězích. Neúspěšné běhy mají počet kroků roven `MAX-TRIES`, tudíž by způsobily přírůstek běhů v tomto počtu kroků a snížili kvalitu vizualizace.

2. **Ohraničení hlavního intervalu s pomocí mezikvartilového rozptylu IQR** – při vykreslení celého intervalu 0 – `MAX-FLIPS` často docházelo k *nehezkým* a nepřehledným vizualizacím, obzvlášť pro formule s malým počtem proměnných. Omezení tohoto intervalu na jeho relevantní část metodou IQR vede k tomu, že relevantní informace zabírají většinu vizualizace a je proto snadnější se v ní vyznat.

   (*Pozn.:* Pro omezení nebyl použit běžný faktor $1.5 \times$ **IRQ**, ale $2 \times$ **IRQ**. Je to pro zaručení toho, aby se v grafu objevil i 90. percentil – viz. 2.6)

Pro každý set formulí tedy vzniklo 6 grafů - 2 histogramy, 2 CDF funkce, a 2 porovnávací grafy. Pro stručnost zde budou uvedené porovnávací grafy, ale všechny vizualizace samostatně lze najít ve složce `pdf-out/`.

Ve všech vizualizacích (1, 2, 3, 4, 5, 6) můžeme pozorovat následující trendy:

- V grafech CDF dosahuje křivka reprezentující probSAT dříve hodnoty 1.0 (100 % úspěšných běhů skončilo v daném kroku nebo dříve) $\rightarrow$ lze interpretovat jako příznak toho, že probSAT rychleji nalézá řešení. Rozdíl mezi probSAT a gsat2 se také zvětšuje s počtem proměnných.

- v Histogramech (více viditelné v sadách s větším počtem proměnných) má probSAT více prvků zastoupených v nižším počtu kroků (lze vidět i na dřívějším 50. a 90. percentilu). Zároveň s počtem kroků začíná převládat gsat2 $\rightarrow$ lze interpretovat jako příznak toho, že probSAT často rychleji nalezne řešení, a také v užším intervalu kroků.

2.6 Metriky – Porovnání percentilů

Další potenciální metrika, dle které lze algoritmy posuzovat, je porovnání percentilů. Percentilem se označuje počet kroků, ve kterém již nějaké specifické procento úspěšných běhů skončilo. *Lepší* algoritmus bude mít obecně nižší hodnotu jakéhokoliv percentilu.

Za vhodné jsme zvolili hodnoty 50 % a 90 %. Zároveň nám přišlo vhodné spojit tuto metriku s vizualizací Histogramu, než jen uvést hodnoty v tabulce. Takto si člověk může lépe představit 'objem' řešení před a za daným percentilem. Hodnoty percentilů jsou popsané v legendě grafu pro přehlednost. Stojí za to zmínit, že se dá zobrazit jak v Histogramu, tak v CDF.

V Histogramech z kapitoly 2.5 jsou vyobrazeny percentily pro oba algoritmy. Ve všech histogramech lze pozorovat následující:

- probSAT má jak 50., tak 90. percentil v menším počtu kroků, než gsat2 → příznak toho, že probSAT nachází více řešení v menším počtu kroků
- probSAT má značně menší rozdíl mezi počtem kroků 90. a 50. percentilu → vypovída o tom, že rozložení řešení probSAT je užší oproti gsat2

2.7 Další potenciální metriky

Pro úplnost je vhodné zmínit nějaké další metriky, kterými by mohl být tento 'report' obohacen – například vizualizace s pomocí box plotu s prvním a třetím kvartilem a mediánem. Takový box plot by mohl lépe ilustrovat *šířku* místa hlavní koncentrace řešení pro každý algoritmus – toto lze do jisté míry pozorovat i s pomocí Histogramu.

3 Výstup & Závěr

Výsledky měření na všech setech konzistentně poukazují na to, že algoritmus probSAT překonává gsat2. Analýza úspěšnosti, histogramů, CDF a percentilů naznačuje, že probSAT nalézá řešení rychleji (v méně krocích) a s větší konzistencí (poměr úspěšných × neúspěšných běhů a menší rozptyl percentilů).

Tento výsledek je v souladu s obecným předpokladem vysloveným v úvodu práce. Lze tedy s určitou jistotou tvrdit, že na daných setech formulí je algoritmus probSAT lepší, než gsat2.

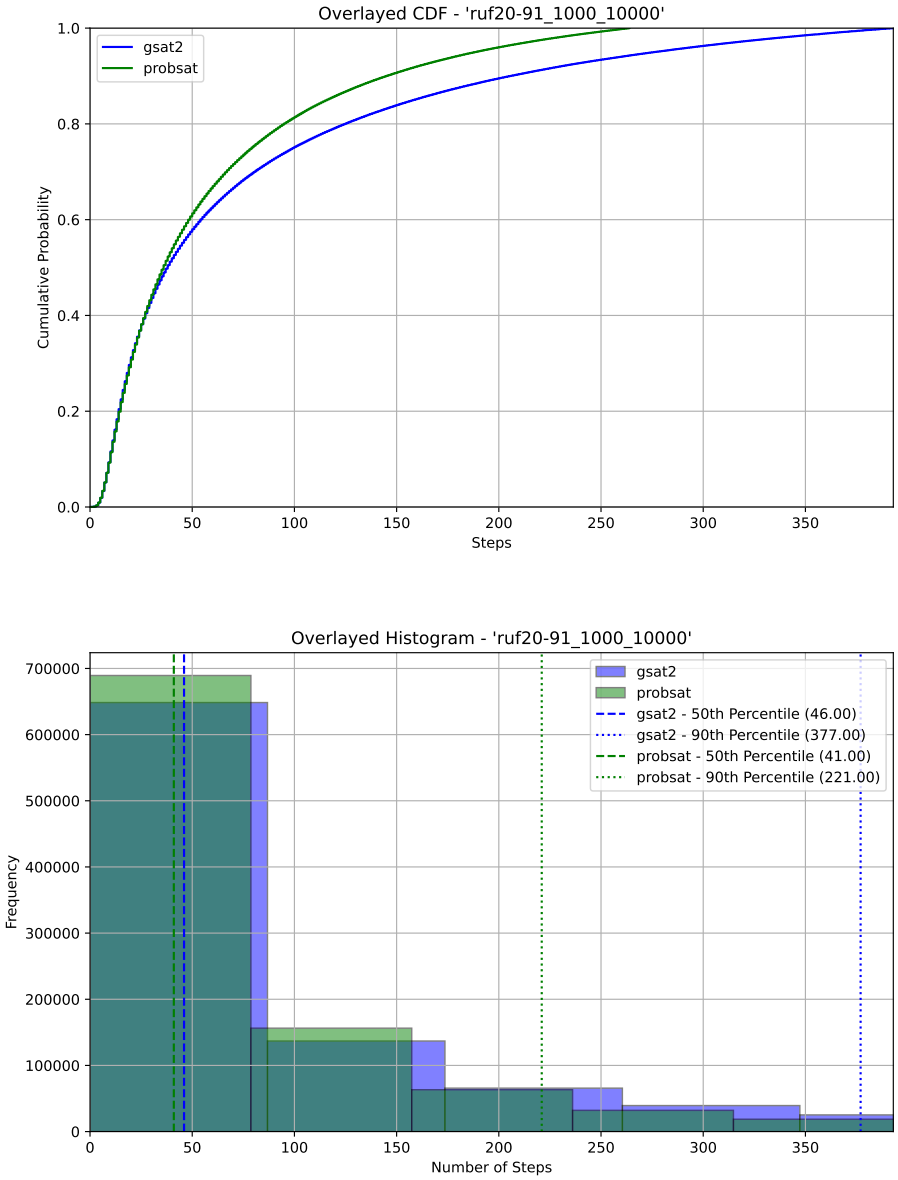

Figure 1: Porovnání Histogramů a CDF pro set `ruf20-91`

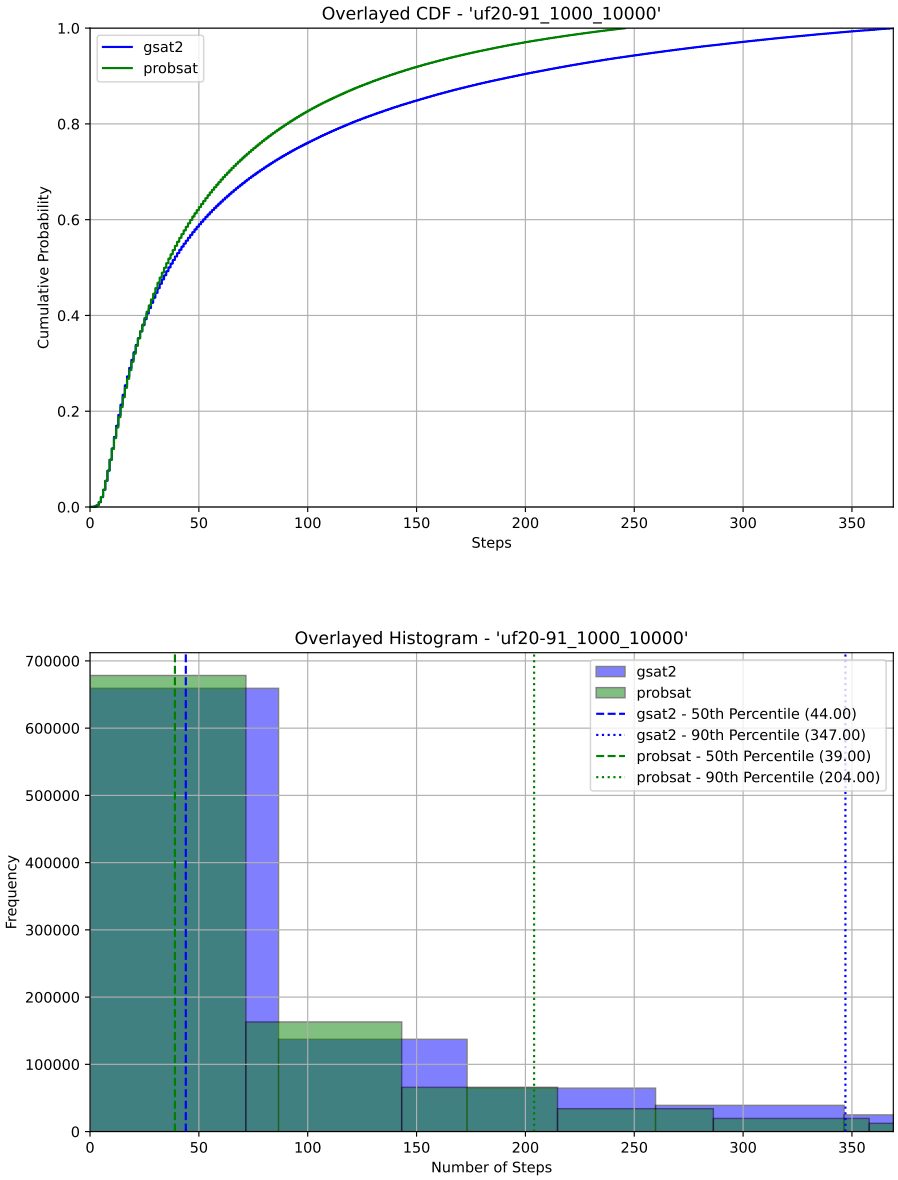

Figure 2: Porovnání Histogramů a CDF pro set `uf20-91`

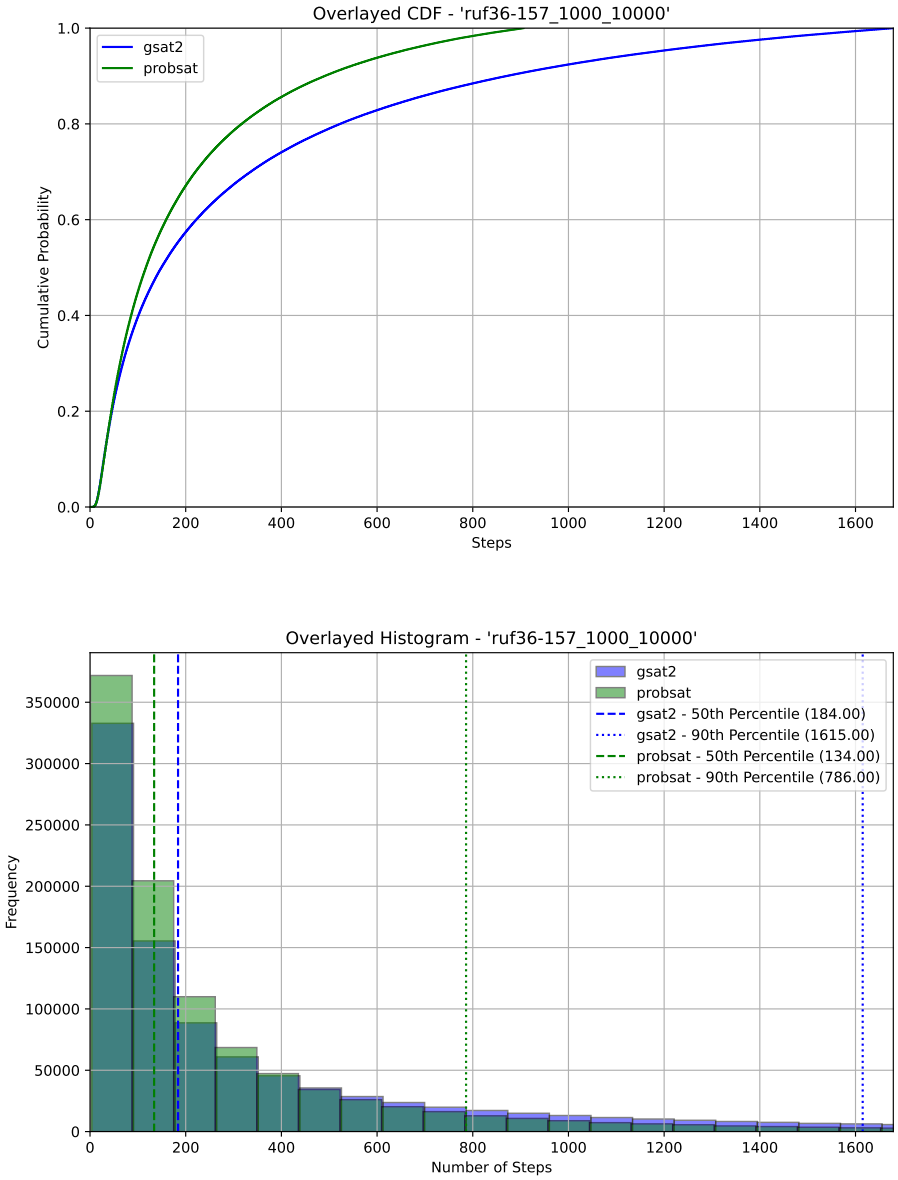

Figure 3: Porovnání Histogramů a CDF pro set `ruf36-157`

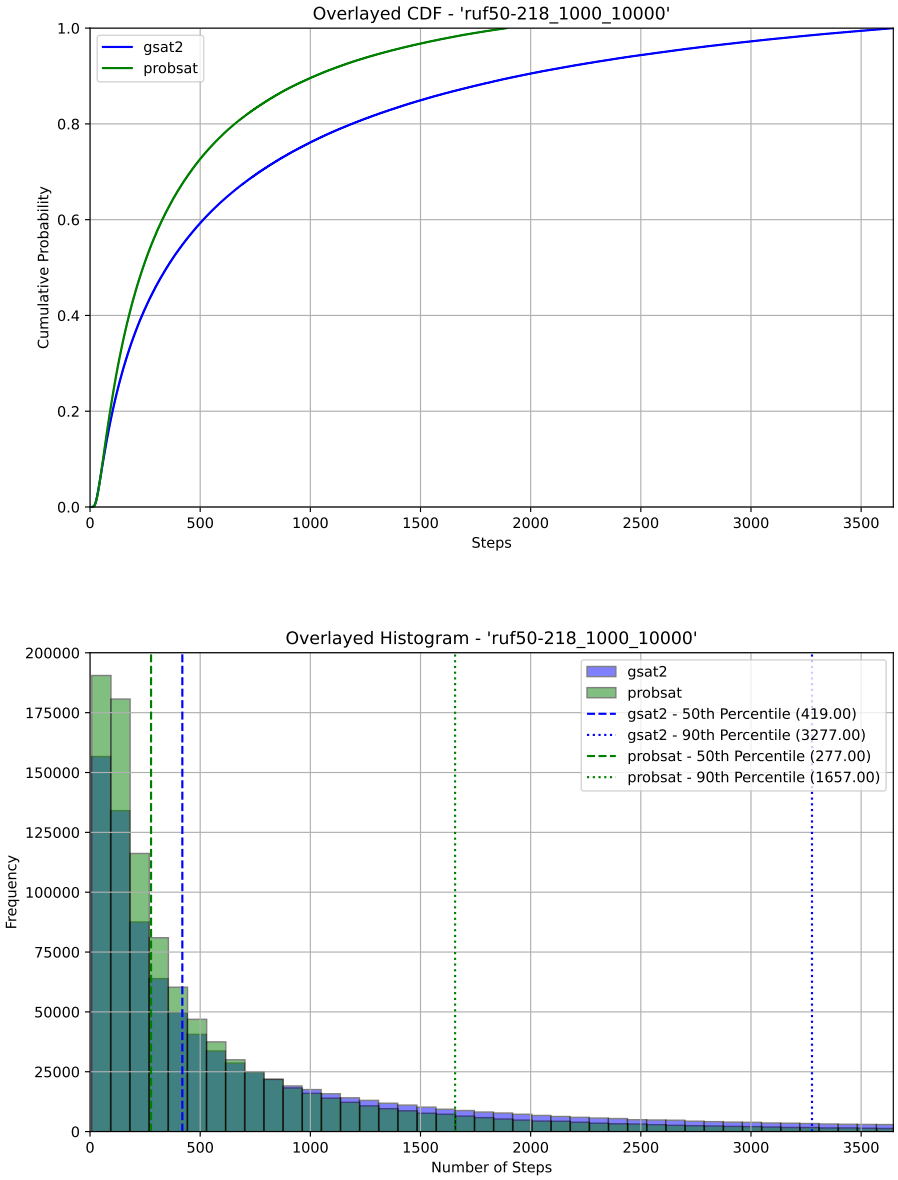

Figure 4: Porovnání Histogramů a CDF pro set `ruf50-218`

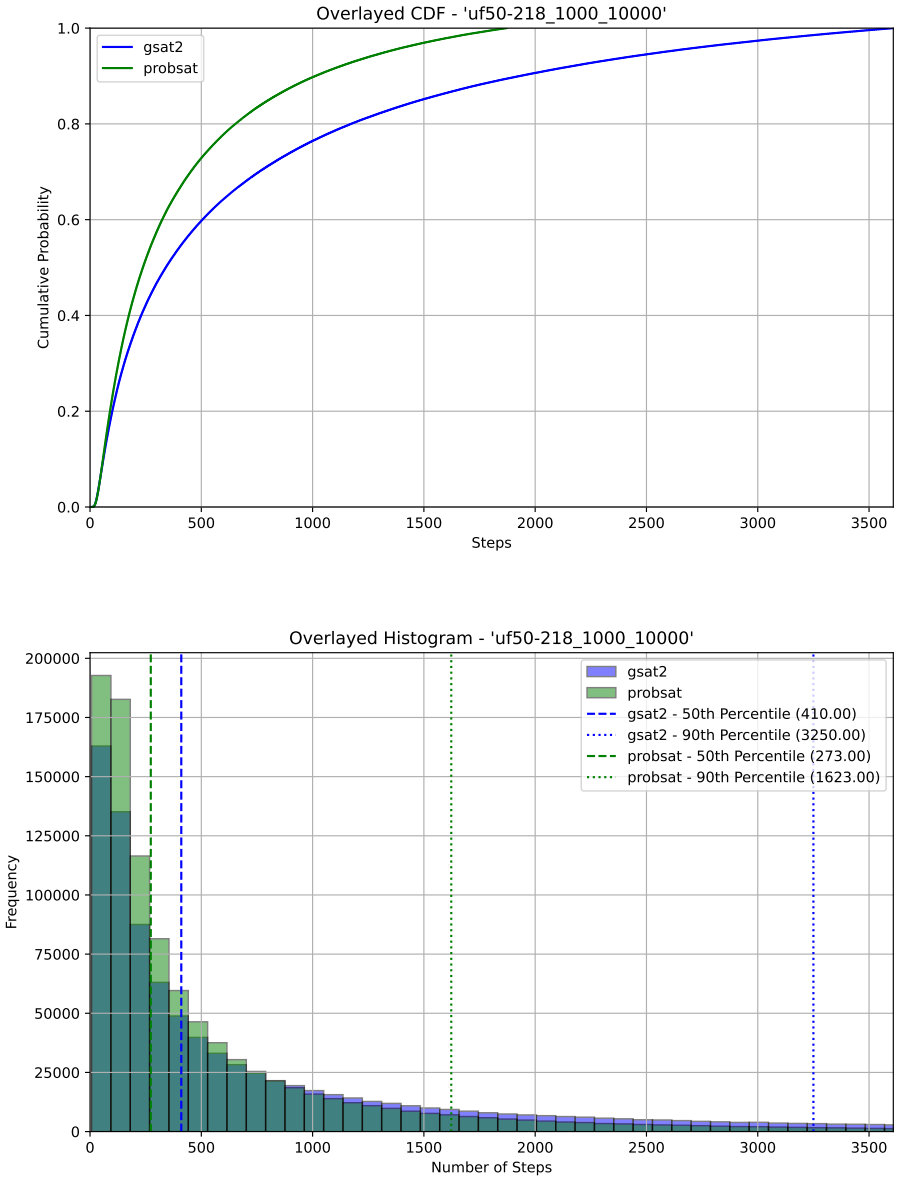

Figure 5: Porovnání Histogramů a CDF pro set `uf50-218`

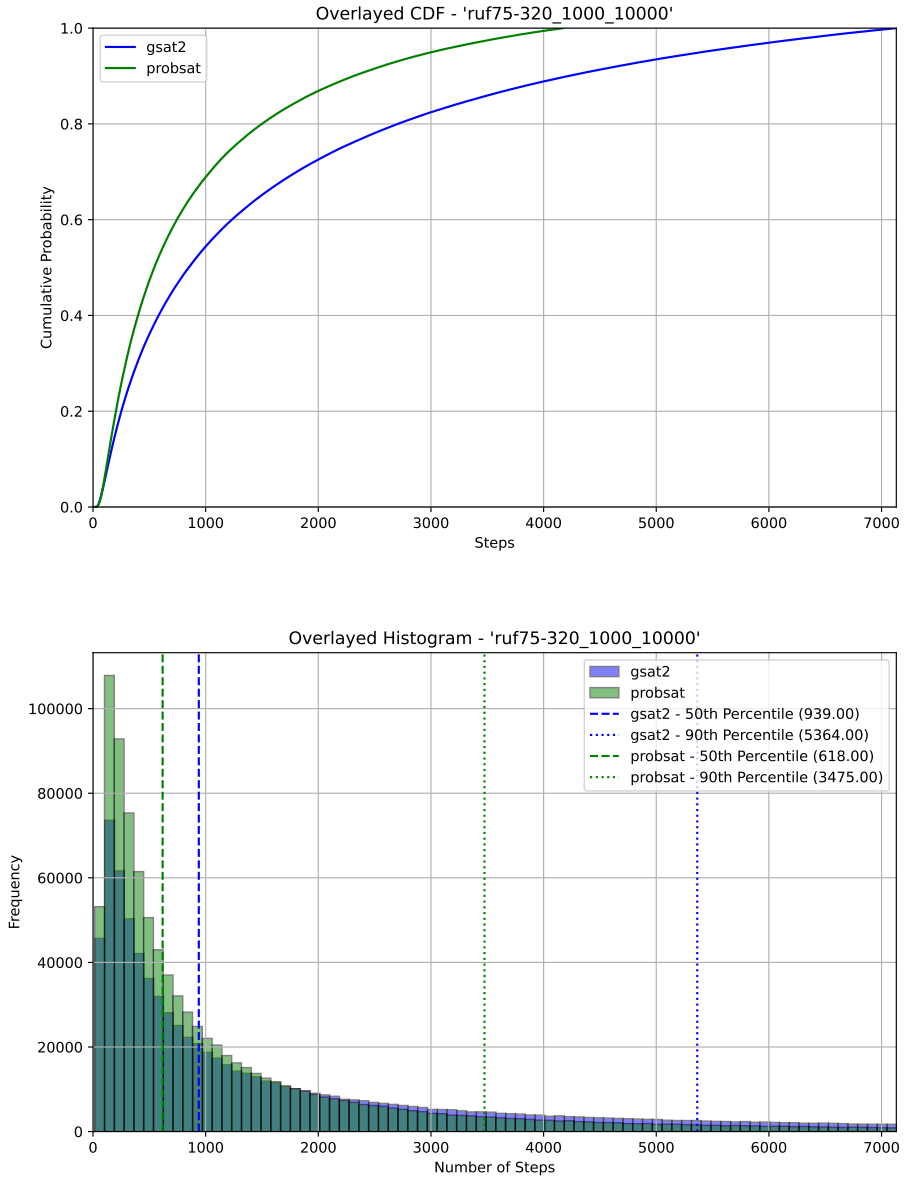

Figure 6: Porovnání Histogramů a CDF pro set `ruf75-320`