

Experimentální vyhodnocení algoritmů

Anna Kapitánová

November 19, 2022

1 Krátké shrnutí

Tato domácí úloha se zabývá experimentálním vyhodnocením algoritmů GSAT a ProbSAT. Jedná se o algoritmy řešící problém SAT (Boolean satisfiability problem). Oba algoritmy byly použity na třech sadách instancí, které obsahovaly instance s různým počtem proměnných a klauzulí. Následně byly provedeny šetření vybranými metrikami na naměřených datech, na základě kterých byla provedena diskuse o úspěšnosti obou algoritmů.

2 Zadání

Experimentálně srovnajte algoritmy GSAT a ProbSAT. Určete, který algoritmus dospěje rychleji (v menším počtu iterací) k řešení obtížných instancí 3-SAT v rozsahu 20-75 proměnných. Zdůvodněte použité metody a metriky, popište interpretaci dat.

- GSAT: $p = 0.4$,
- ProbSAT: $c_m = 0$, $c_b = 2.3$.

[Odkaz na zadání.](#)

3 Úvod

Experiment byl proveden na následujících sadách dat ([Odkaz na použité sady dat](#)):

- ruf20-91: sada obsahující obtížné instance – 20 proměnných a 91 klauzulí,
- ruf50-218: sada obsahující obtížné instance – 50 proměnných a 218 klauzulí,
- ruf75-320: sada obsahující obtížné instance – 75 proměnných a 320 klauzulí.

Maximální počet iterací byl nastaven na 3000, počet běhů na 1000. Nad každou instancí byl tedy program s daným algoritmem spuštěn 1000-krát. Následně byly nad těmito výsledky napočítány metriky a výsledky metrik byly uloženy jako jeden záznam příslušný dané instanci.

3.1 Algoritmus GSAT

Pseudokód algoritmu GSAT zobrazuje obrázek [1](#).

3.2 Algoritmus ProbSAT

Pseudokód algoritmu ProbSAT zobrazuje obrázek [2](#).

```

Procedure GSAT (formula F)
  for i := 1 to MAX-TRIES
    T := a randomly generated truth assignment
    for j := 1 to MAX-FLIPS
      if T satisfies F then return T
      with probability p do
        randomly pick an unsatisfied clause in F
        randomly pick a variable in that clause
      else
        randomly pick any variable in T
        whose value flip results in greatest
        decrease (can be 0 or negative)
        in the number of unsatisfied clauses
      flip the truth assignment of the chosen variable
    end for
  end for
  return "No satisfying assignment found"

```

Figure 1: Algoritmus GSAT

```

Procedure probSAT (formula F)
  for i := 1 to MAX-TRIES
    T := a randomly generated truth assignment
    for j := 1 to MAX-FLIPS
      if T satisfies F then return T
      randomly pick an unsatisfied clause in F
      randomly pick a variable in that clause,
        each variable x has the probability proportional to f(x,F)
      flip the truth assignment of the chosen variable
    end for
  end for
  return "No satisfying assignment found"

function f (variable x, formula F)
  make := number of clauses which become satisfied after the flip of x
  break := number of clauses which cease to be satisfied after the flip of x

  return make^cm / (eps+break)^cb

```

Figure 2: Algoritmus ProbSAT

4 Metriky

Jako metriky byly vybrány:

Průměrný počet kroků Důležitá metrika, pomocí které dokážeme lépe rozhodnout, který z algoritmů je průměrně rychlejší. Obsahuje průměrný počet iterací, než byl nalezen správný výsledek dané instance.

Vážený průměrný počet kroků Touto metrikou dokážeme lépe zohlednit, jak často byl daný algoritmus na jednotlivých instancích neúspěšný. Pokud totiž algoritmus nedokázal během nastavených 3000 iterací najít řešení, je tento běh brán jako neúspěšný a do váženého průměru se započítá počet proběhlých iterací pronásobený deseti.

Počet úspěšných běhů Metrika zohledňující v kolika bězích dokázal algoritmus nalézt řešení na dané instanci. Na každé instanci byl algoritmus spuštěn 1000-krát, pokud tedy nabývá tato metrika pro nějakou instanci hodnoty 1000, byl algoritmus úspěšných ve všech bězích.

5 Data

Byly vytvořeny dva datasety pro každou sadu dat, jeden dataset shromažďoval výsledky algoritmu GSAT, druhý algoritmu Probsat. Navíc byly vytvořeny datasety obsahující výsledky instancí ze všech datových sad. Následující výčet stručně shrnuje popis jednotlivých sloupců datasetů:

- inst: jméno instance,
- succ: počet běhů, kdy bylo nalezeno řešení algoritmem,
- steps: průměrný počet iterací algoritmu pro všechny běhy,
- max_iter: maximální povolený počet iterací,
- avg_fined: vážený počet iterací algoritmu, kde neúspěšný běh se počítá 10x (Penalized Average Runtime).

6 Výsledky

6.1 Srovnání průměrného počtu kroků algoritmů

Na histogramu 3 je vidět lognormální vzhled rozložení průměrného počtu kroků algoritmů. Modře je na grafech zobrazen algoritmus ProbSAT, žlutě potom algoritmus GSAT.

Z grafu je dobře viditelné, že algoritmus ProbSAT zde dominuje algoritmus GSAT. Medián průměrného počtu kroků odpovídá 672,91 krokům u algoritmu GSAT a 194,30 krokům pro ProbSAT. Minimum je pro GSAT 13,68 a ProbSAT 13,14 kroků. Dolní 25% kvantil u GSATu odpovídá hodnotě 167,46 kroků a horní 75% 1516,73. U ProbSAT je dolní kvantil 68,38 a horní 672,77 kroků.

6.2 Srovnání váženého průměru počtu kroků algoritmů

Srovnání vážených průměrů počtu kroků poskytují grafy 4 a 5.

Graf 4, ukazující výsledky na všech datech, zobrazuje dominanci algoritmu ProbSAT. Dolní 25% kvantil u GSATu odpovídá hodnotě 167,67 kroků a horní 75% 8491,68. U ProbSATu je dolní kvantil 68,38 a horní 1036,96 kroků. Medián průměrného počtu vážených kroků odpovídá 1543,13 krokům u algoritmu GSAT a 194,74 krokům pro ProbSAT.

Grafy na obrázku 5 obsahují histogramy váženého počtu kroků postupně pro 20, 50 a 75 proměnných. Z grafů je pěkně vidět, jak s rostoucím počtem proměnných roste i rozdíl v algoritmech. Na 75 proměnných je již rozdíl velmi výrazný ve prospěch algoritmu ProbSAT. Medián je u 75 proměnných roven 9306,36 pro GSAT a 2400,32 pro ProbSAT, směrodatná odchylka potom 7553,97 u GSAT a 6175,35 u ProbSAT.

6.3 Srovnání počtu úspěšných běhů

Na instancích s 20 proměnnými byl algoritmus ProbSAT úspěšný ve všech případech, GSAT nebyl úspěšný celkem ve 116 instancích. V instancích s více proměnnými již nebyl ani jeden z algoritmů úspěšný ve všech případech.

Graf 6 zobrazuje úspěšnost obou algoritmů na všech instancích, graf ale bohužel není moc dobře čitelný, pro lepší orientaci poslouží grafy na obrázku 7. Zde jsou seřazeny postupně grafy zobrazující úspěšnost obou algoritmů pro instance s 20, 50 a 75 proměnnými. Největší rozdíl v úspěšnosti algoritmů evidujeme na instancích se 75 proměnnými. Zde je viditelná jasná převaha algoritmu ProbSAT. Na instancích s 75 proměnnými byl algoritmus ProbSAT úspěšný průměrně ve 848 případech, GSAT pouze ve 672 případech.

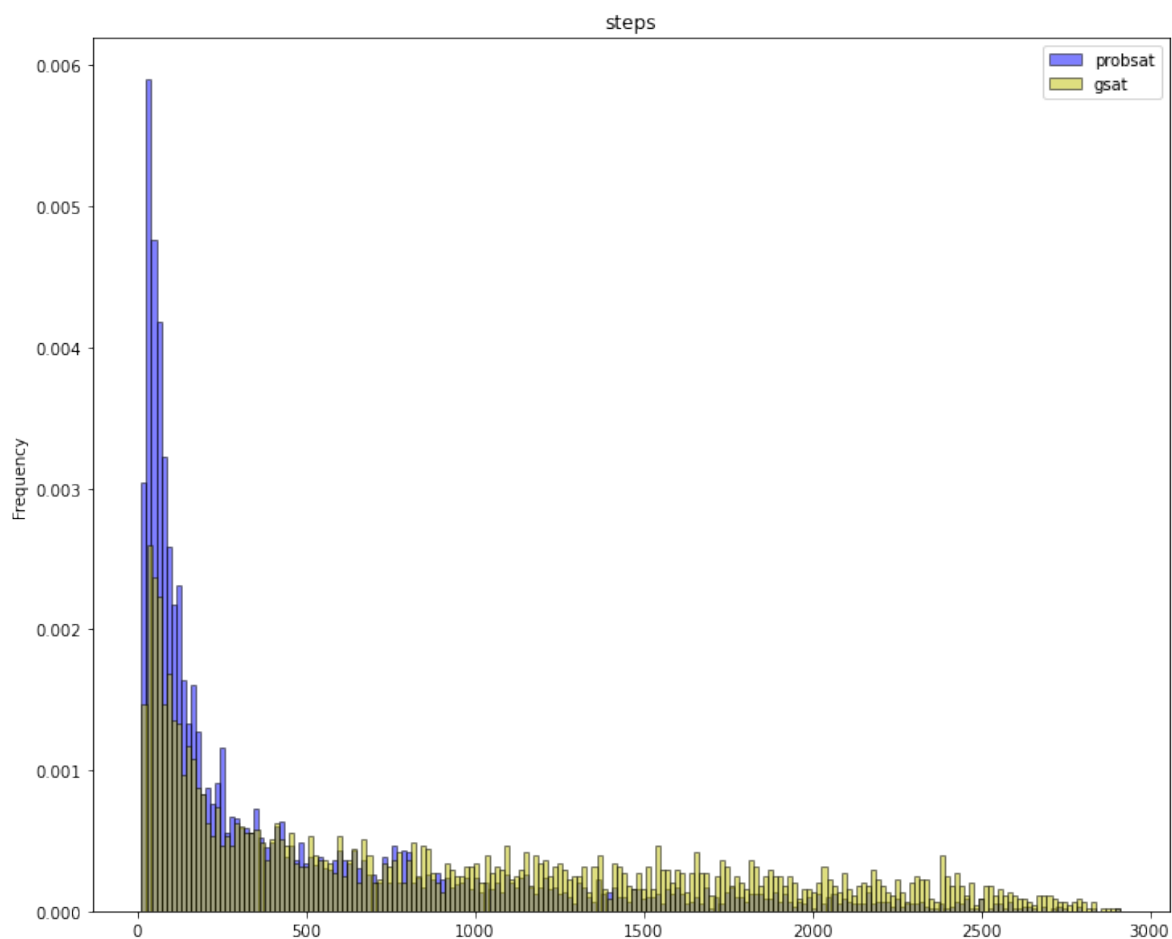


Figure 3: Srovnání průměrného počtu kroků algoritmů

7 Diskuse

Výsledky provedeného experimentu nasvědčují tomu, že algoritmus ProbSAT je z hlediska průměrného počtu kroků nalezení řešení rychlejší než-li algoritmus GSAT a to na všech sadách dat. Pokud vezmeme v potaz i neúspěšné běhy např. pomocí váženého počtu průměrného počtu kroků, vidíme opět převahu algoritmu ProbSAT oproti GSATu. Co se týče počtů úspěšných běhů, algoritmus ProbSAT i zde prokazuje svou dominanci a je úspěšnější an všech sadách dat.

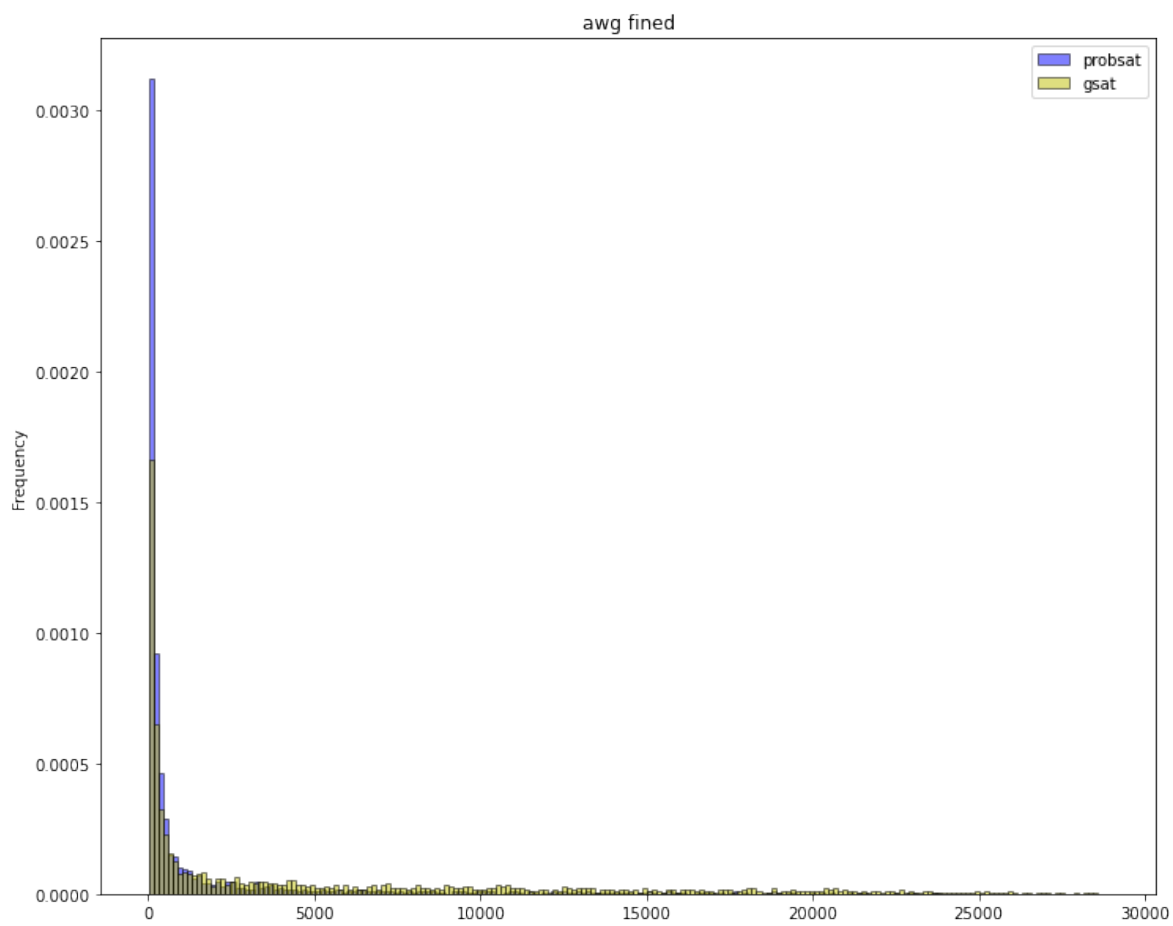


Figure 4: Srovnání váženého průměru počtu kroků algoritmů

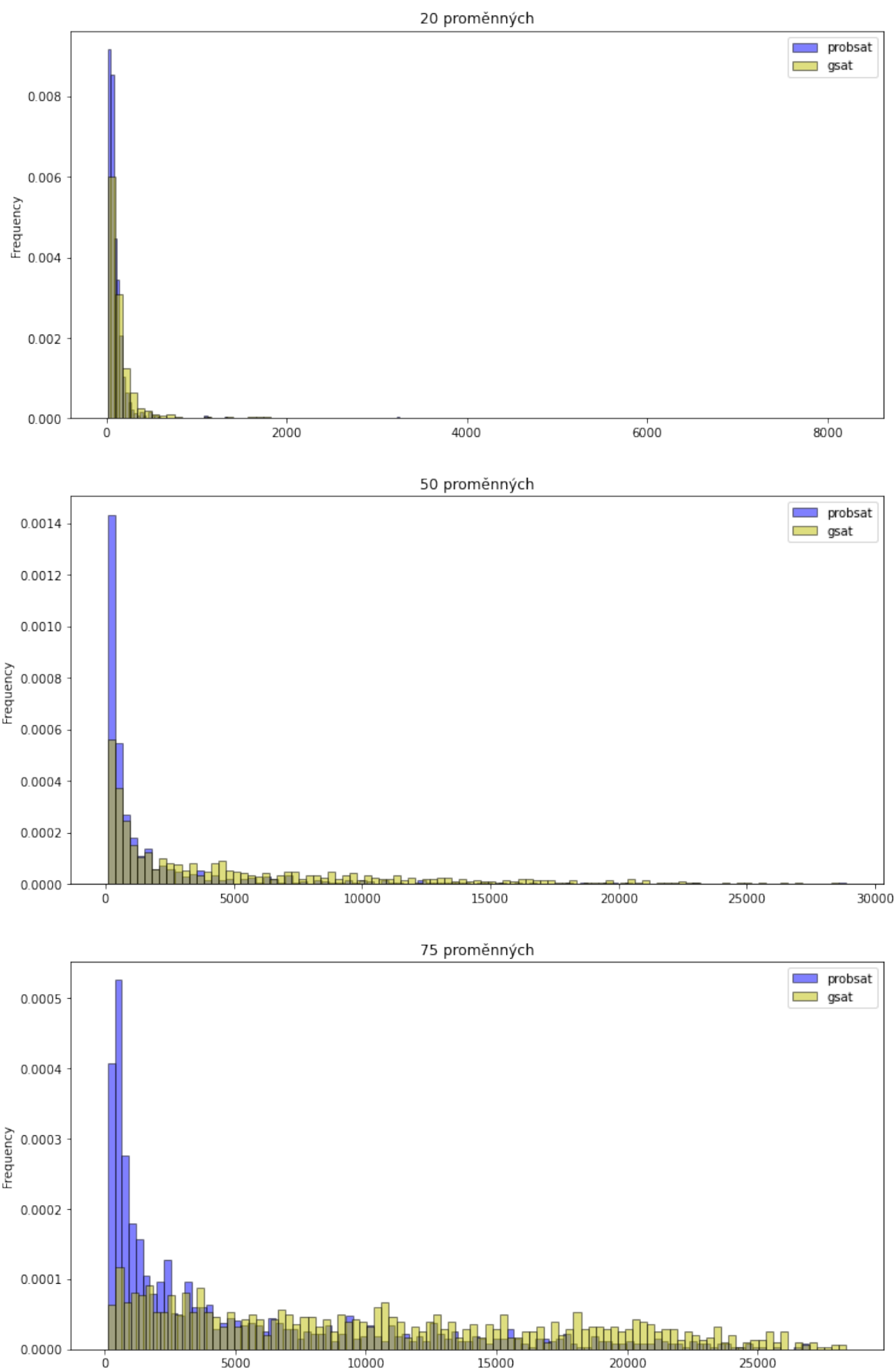


Figure 5: Srovnání váženého průměru počtu kroků algoritmů

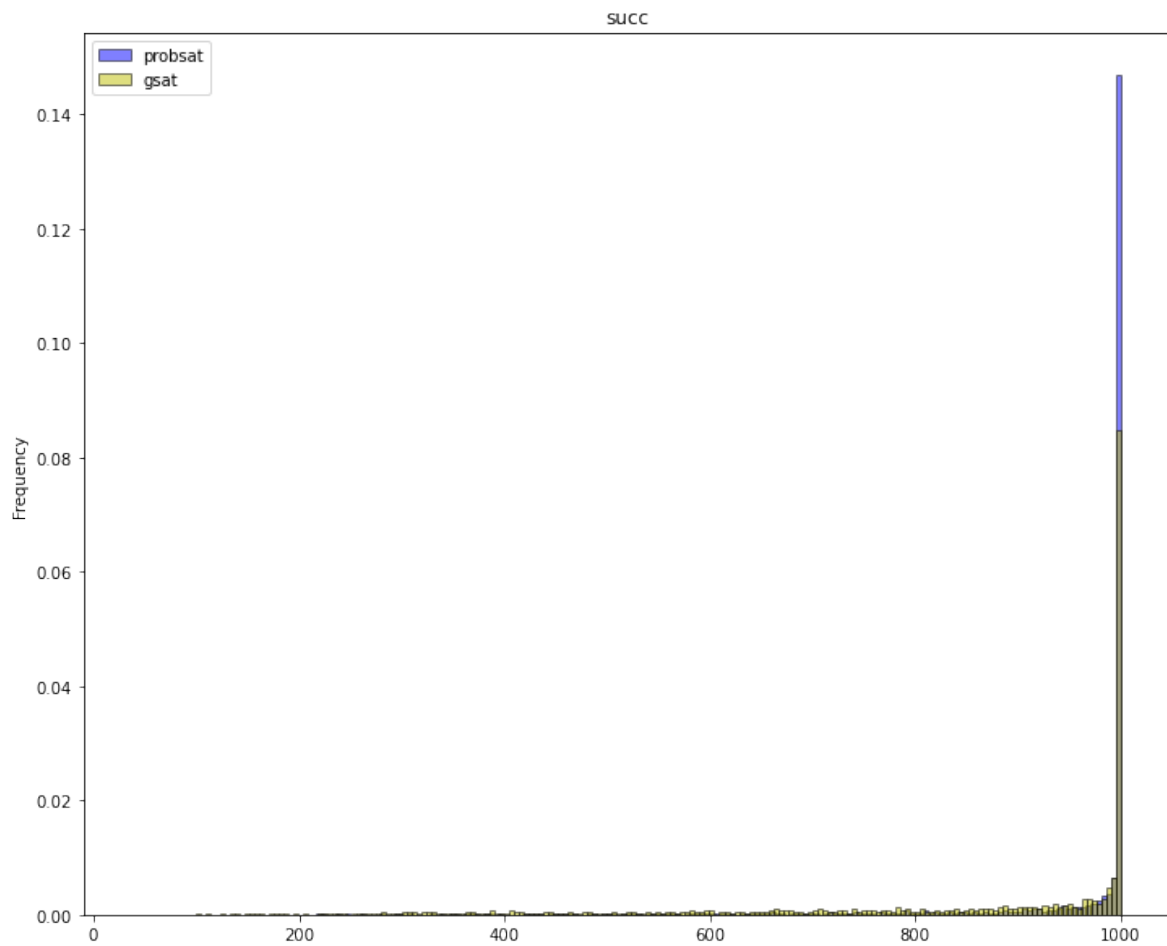


Figure 6: Srovnání počtu úspěšných běhů

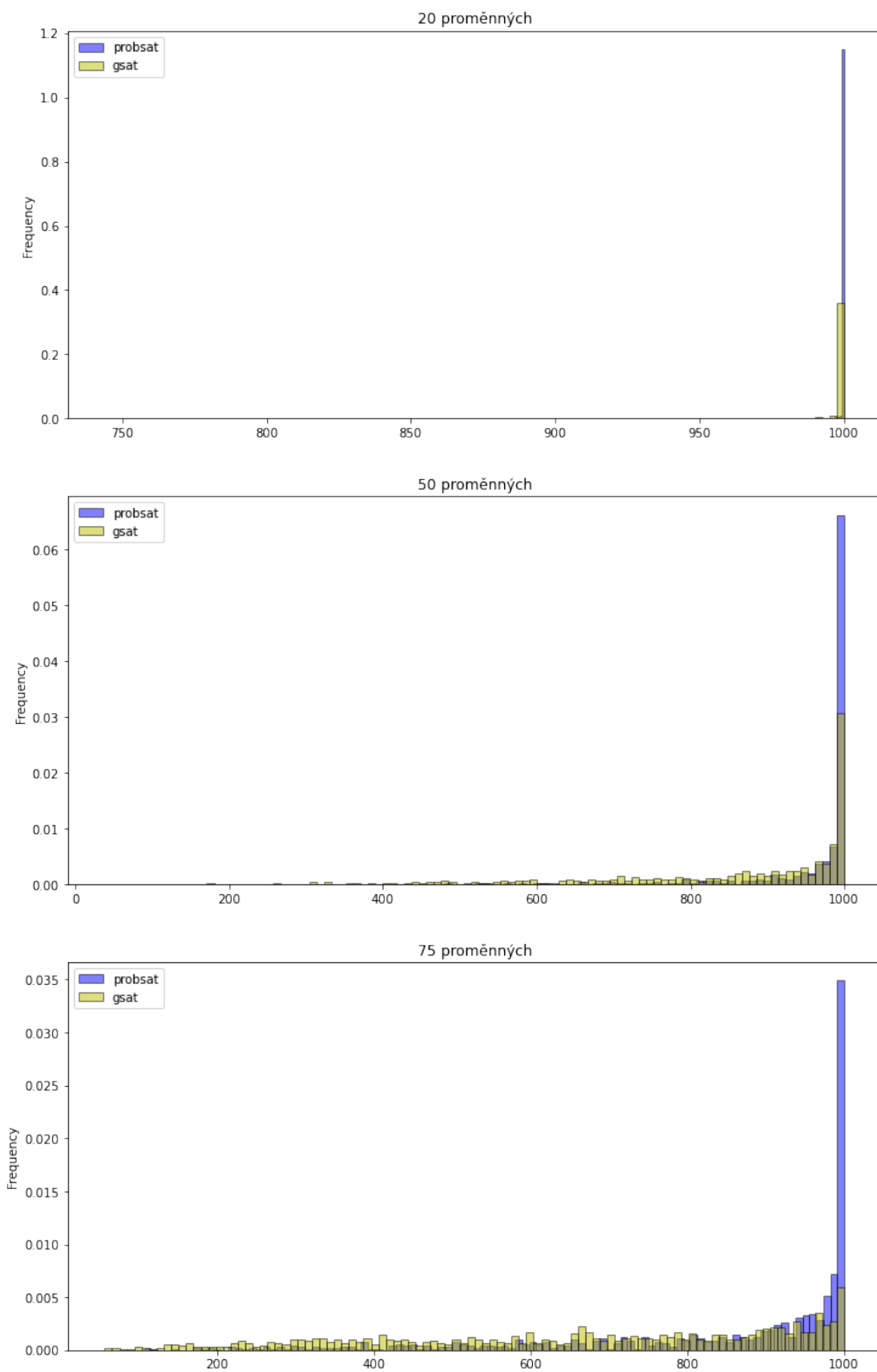


Figure 7: Srovnání počtu úspěšných běhů