Evidence for Long-Tails in SLS Algorithms

Florian Wörz Jan-Hendrik Lorenz

European Symposium on Algorithms (ESA)
September 6th, 2021

Motivation and Aim of This Talk

Motivation: Recent Developments in SLS Solvers

- SLS = successful paradigm for solving SAT
- Recent development [LW20]:
  - GapSAT solves not the original instance, but a modified, yet logically equivalent one
  - Empirically shown: on average, this improves the performance of state-of-the-art SLS solvers

Aim: Model and Analyze GapSAT

Caveat: Only shallow understanding of how runtimes are affected

AIM: Model this modification process and conduct an empirical analysis of the hardness of logically equivalent formulas

Overview of Our Results

1. A **lognormal** distribution perfectly characterizes the hardness
   $\Longrightarrow$ The hardness is **long-tailed**
2. Restarts are useful for long-tailed algorithms

Statistical Hardness Distribution with ALFA

ALFA (Adjusted logical formula algorithm)

1. Model the addition of a set of logically equivalent clauses L to a formula F and the subsequent solving of the amended formula $F^{(1)} := F \cup L$ by an SLS solver

Algorithm: Adjusted logical formula algorithm (ALFA)

Input: Boolean formula F, **Promise:** $F \in \mathrm{SAT}$

Generate **randomly** a set L of clauses such that $F \models L$
Call SLS$(F \cup L)$ for some SLS solver SLS

2. Use width-4-bounded resolution to generate L

Experimental Setup, Instance Types, and Solvers Used (1/2)

F

$F^{(1)}$

$F^{(2)}$

$\vdots$

$F^{(5000)}$

Experimental Setup, Instance Types, and Solvers Used (1/2)

solve 100 times with each solver S

F

$F^{(1)}$ flips$_S(F^{(1)}, s_1)$, $\ldots$, flips$_S(F^{(1)}, s_{100})$

$F^{(2)}$

$\vdots$

$F^{(5000)}$

Experimental Setup, Instance Types, and Solvers Used (1/2)

solve 100 times with each solver S

$$
F \quad
\begin{array}{c|ccc}
F^{(1)} & \text{flips}_S(F^{(1)}, s_1), & \ldots, & \text{flips}_S(F^{(1)}, s_{100}) \\
F^{(2)} & \text{flips}_S(F^{(2)}, s_1), & \ldots, & \text{flips}_S(F^{(2)}, s_{100}) \\
\vdots & & & \\
F^{(5000)} & & &
\end{array}
$$

Experimental Setup, Instance Types, and Solvers Used (1/2)

solve 100 times with each solver S

$$
F \quad
\begin{array}{llll}
F^{(1)} & \text{flips}_S(F^{(1)}, s_1), & \dots, & \text{flips}_S(F^{(1)}, s_{100}) \\
F^{(2)} & \text{flips}_S(F^{(2)}, s_1), & \dots, & \text{flips}_S(F^{(2)}, s_{100}) \\
\vdots & & & \\
F^{(5000)} & \text{flips}_S(F^{(5000)}, s_1), & \dots, & \text{flips}_S(F^{(5000)}, s_{100})
\end{array}
$$

Experimental Setup, Instance Types, and Solvers Used (1/2)

solve 100 times with each solver S

F	$F^{(1)}$	$\text{flips}_S(F^{(1)}, s_1)$,	$\ldots$,	$\text{flips}_S(F^{(1)}, s_{100})$	$\text{mean}_S(F^{(1)}) =: x_1$
	$F^{(2)}$	$\text{flips}_S(F^{(2)}, s_1)$,	$\ldots$,	$\text{flips}_S(F^{(2)}, s_{100})$	
	$\vdots$				
	$F^{(5000)}$	$\text{flips}_S(F^{(5000)}, s_1)$,	$\ldots$,	$\text{flips}_S(F^{(5000)}, s_{100})$	

Experimental Setup, Instance Types, and Solvers Used (1/2)

solve 100 times with each solver S

$$
F \quad
\begin{array}{lllll}
F^{(1)} & \text{flips}_S(F^{(1)}, s_1), & \dots, & \text{flips}_S(F^{(1)}, s_{100}) & \text{mean}_S(F^{(1)}) =: x_1 \\
F^{(2)} & \text{flips}_S(F^{(2)}, s_1), & \dots, & \text{flips}_S(F^{(2)}, s_{100}) & \text{mean}_S(F^{(2)}) =: x_2 \\
\vdots & & & & \\
F^{(5000)} & \text{flips}_S(F^{(5000)}, s_1), & \dots, & \text{flips}_S(F^{(5000)}, s_{100}) &
\end{array}
$$

Experimental Setup, Instance Types, and Solvers Used (1/2)

solve 100 times with each solver S

$$
F \quad
\begin{array}{lllll}
F^{(1)} & \text{flips}_S(F^{(1)}, s_1), & \ldots, & \text{flips}_S(F^{(1)}, s_{100}) & \text{mean}_S(F^{(1)}) =: x_1 \\
F^{(2)} & \text{flips}_S(F^{(2)}, s_1), & \ldots, & \text{flips}_S(F^{(2)}, s_{100}) & \text{mean}_S(F^{(2)}) =: x_2 \\
\vdots & & & & \\
F^{(5000)} & \text{flips}_S(F^{(5000)}, s_1), & \ldots, & \text{flips}_S(F^{(5000)}, s_{100}) & \text{mean}_S(F^{(5000)}) =: x_{5000}
\end{array}
$$

Experimental Setup, Instance Types, and Solvers Used (1/2)

solve 100 times with each solver S

$$
F \quad
\begin{array}{llll}
F^{(1)} & \mathrm{flips}_S(F^{(1)}, s_1), \quad \dots, & \mathrm{flips}_S(F^{(1)}, s_{100}) & \mathrm{mean}_S(F^{(1)}) =: x_1 \\
F^{(2)} & \mathrm{flips}_S(F^{(2)}, s_1), \quad \dots, & \mathrm{flips}_S(F^{(2)}, s_{100}) & \mathrm{mean}_S(F^{(2)}) =: x_2 \\
\vdots & & & \\
F^{(5000)} & \mathrm{flips}_S(F^{(5000)}, s_1), \quad \dots, & \mathrm{flips}_S(F^{(5000)}, s_{100}) & \mathrm{mean}_S(F^{(5000)}) =: x_{5000}
\end{array}
$$

Given such a sample $(x_1, \dots, x_{5000})$, we will plot the empirical distribution function

$$\hat{F}_{5000}(t) := \frac{1}{5000} \sum_{i=1}^{5000} \mathbb{1}_{\{x_i \leqslant t\}}, \quad t \in \mathbb{R}.$$

Experimental Setup, Instance Types, and Solvers Used (2/2)

Instance Types:

1. Hidden Solution (with different parameters)
2. Uniform Random
3. Factoring
4. Coloring

Experimental Setup, Instance Types, and Solvers Used (2/2)

Instance Types:

1. Hidden Solution (with different parameters)
2. Uniform Random
3. Factoring
4. Coloring

Used Solvers:

1. Schöning's Random Walk Algorithm SRWA [Sch02] — all instance types
2. PROBSAT solver family[1] [BS12] — 55 hidden solution instances with $n \in \{50, 100, 150, 200, 300, 800\}$
3. YALSAT[2] [Bie14] — 10 hidden solution instances with $n = 300$

---

[1] One of these solvers won the random track of the SAT competition 2013
[2] Won the random track of the SAT competition 2017

Experimental Setup, Instance Types, and Solvers Used (2/2)

Instance Types:

1. Hidden Solution (with different parameters)
2. Uniform Random
3. Factoring
4. Coloring

Used Solvers:

1. Schöning's Random Walk Algorithm SRWA [Sch02] — all instance types
2. PROBSAT solver family[1] [BS12] — 55 hidden solution instances with $n \in \{50, 100, 150, 200, 300, 800\}$
3. YALSAT[2] [Bie14] — 10 hidden solution instances with $n = 300$

Total CPU time: 80 years

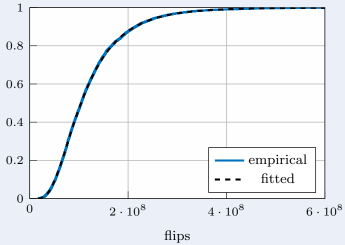
---

[1] One of these solvers won the random track of the SAT competition 2013

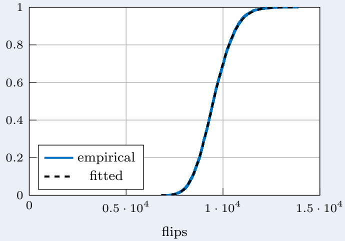
[2] Won the random track of the SAT competition 2017

Experimental Results and Statistical Evaluation (1/3)

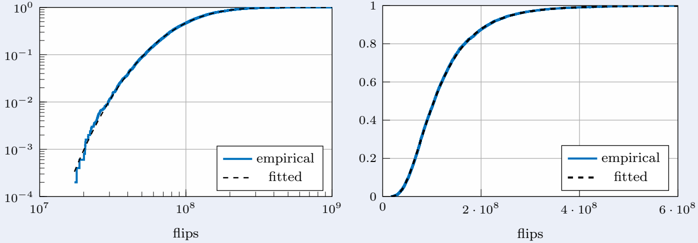
Instance A (Factoring solved with SRWA)



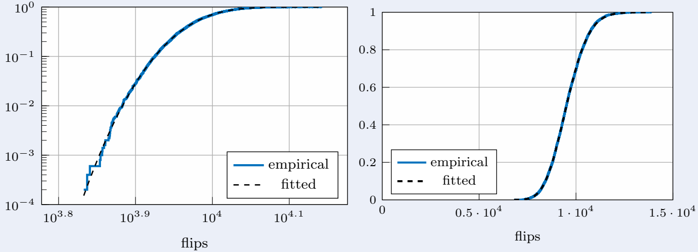
Instance B (Hidden Solution solved with probSAT)



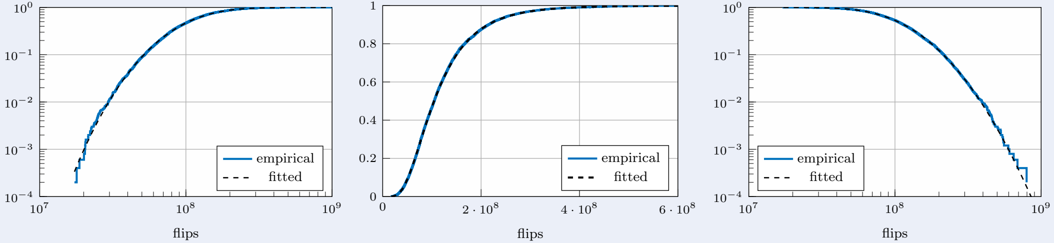
Instance A (Factoring solved with SRWA)



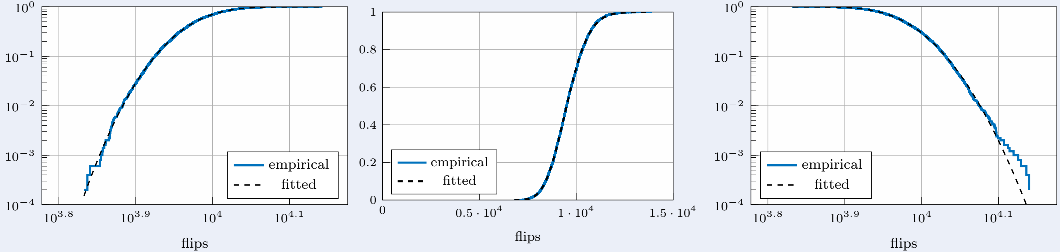
Instance B (Hidden Solution solved with probSAT)



Instance A (Factoring solved with SRWA)



Instance B (Hidden Solution solved with probSAT)



Experimental Results and Statistical Evaluation (2/3)

Résumé: LogN seems to be well suited

↳ Concretize this through χ^2 goodness-of-fit test (and additional bootstrap test)

Example

Instance A:

$$p_{\chi^2} \approx 0.783 \gg 0.05$$
$$p_{\text{👣}} \approx 0.76 \gg 0.05$$

Instance B:

$$p_{\chi^2} \approx 0.008$$
$$p_{\text{👣}} \approx 0.013$$

Experimental Results and Statistical Evaluation (3/3)

Table 1: Statistical goodness-of-fit results for ALFA+SRWA

	hidden	different chances	uniform	factoring	coloring	total
rejected	0	2	1	2	0	5
# of instances	20	120	25	33	32	230

Experimental Results and Statistical Evaluation (3/3)

Table 1: Statistical goodness-of-fit results for ALFA+SRWA

	hidden	different chances	uniform	factoring	coloring	total
rejected	0	2	1	2	0	5
# of instances	20	120	25	33	32	230

Table 2: Goodness-of-fit results for ALFA+PROBSAT

number of variables	50	100	150	200	300	800	total
rejected	2	2	1	0	2	0	7
# of instances	10	10	10	10	10	5	55

Experimental Results and Statistical Evaluation (3/3)

Table 1: Statistical goodness-of-fit results for ALFA+SRWA

	hidden	different chances	uniform	factoring	coloring	total
rejected	0	2	1	2	0	5
# of instances	20	120	25	33	32	230

Table 2: Goodness-of-fit results for ALFA+PROBSAT

number of variables	50	100	150	200	300	800	total
rejected	2	2	1	0	2	0	7
# of instances	10	10	10	10	10	5	55

Table 3: Goodness-of-fit results for ALFA+YALSAT

	χ^2	
rejected	2	0
# of instances	10	10

Conjectures

Conjecture (Strong Conjecture)

The runtime of ALFA *with* SLS $\in$ {SRWA, PROBSAT, YALSAT} *follows a lognormal distribution.*

Conjectures

Conjecture (Strong Conjecture)

The runtime of ALFA *with* SLS $\in$ {SRWA, PROBSAT, YALSAT} *follows a lognormal distribution.*

Definition ([FKZ11])

A positive, real-valued random variable X is *long-tailed*, if and only if

$$\forall x \in \mathbb{R}^+ : \Pr[X > x] > 0 \qquad \text{and} \qquad \forall y \in \mathbb{R}^+ : \lim_{x \to \infty} \frac{\Pr[X > x + y]}{\Pr[X > x]} = 1.$$

Conjecture (Weak Conjecture)

The runtime of ALFA *with* SLS $\in$ {SRWA, PROBSAT, YALSAT} *follows a long-tailed distribution.*

Restarts Are Useful For Long-Tailed Distributions

Extend Known Result for Lognormal Distributions

Definition

Restarts are *useful* if there is a $t > 0$ such that

$$\mathrm{E}[X_t] < \mathrm{E}[X].$$

A condition for the usefulness of restart was proven in [Lor18]:

Theorem ([Lor18])

Strong Conjecture (runtimes are lognormally distributed) $\Longrightarrow$ *Restarts are useful*

Our paper extends this result and mathematically proves that restarts are useful even if only the Weak Conjecture holds (i. e., restarts are useful for long-tailed distributions).

Extended Result

Theorem

Consider a positive, long-tailed random variable X with continuous pdf f, cdf F, and hazard rate function $r(t) := \frac{f(t)}{1-F(t)}$.
Also assume that

- *either $\mathrm{E}[X] = \infty$ holds;*
- *or the limits $\lim_{t\to\infty} r(t)$ and $\lim_{t\to\infty} t^2 \cdot f(t)$ both exist.*

In both cases, restarts are useful for X.

The conditions of this theorem are not restrictive since all naturally occurring long-tail distributions satisfy these conditions (see [NWZ20]).

Conjecture (Corollary of the Weak Conjecture)

Restarts are useful for ALFA *with* SLS $\in$ {SRWA, PROBSAT, YALSAT}.

Summary and Outlook

Conclusion

Have provided compelling evidence that the runtime of Alfa follows a long-tailed or lognormal distribution.

According to [SMP11]: super-linear speedups by parallelization $\Longrightarrow$ usefulness of restarts

Question 1: Can super-linear speedups be obtained by parallelizing Alfa-type algorithms?

Question 2: Can some of the Conjectures be theoretically proven?

Question 3: What can be said about CDCL solvers?