CPSC 532D - MODULE 13:

RANDOMISED TREE SEARCH

Holger H. Hoos

Department of Computer Science
University of British Columbia
Canada

Learning Goals

- Understand motivation and concepts of randomised systematic search (RSS) and stochastic tree search (STS).
- Understand randomisation and restart mechanisms for RSS.
- Know about characteristic RSS behaviour, in particular "heavy-tailed" run-time and search cost distributions.

Motivation & Background

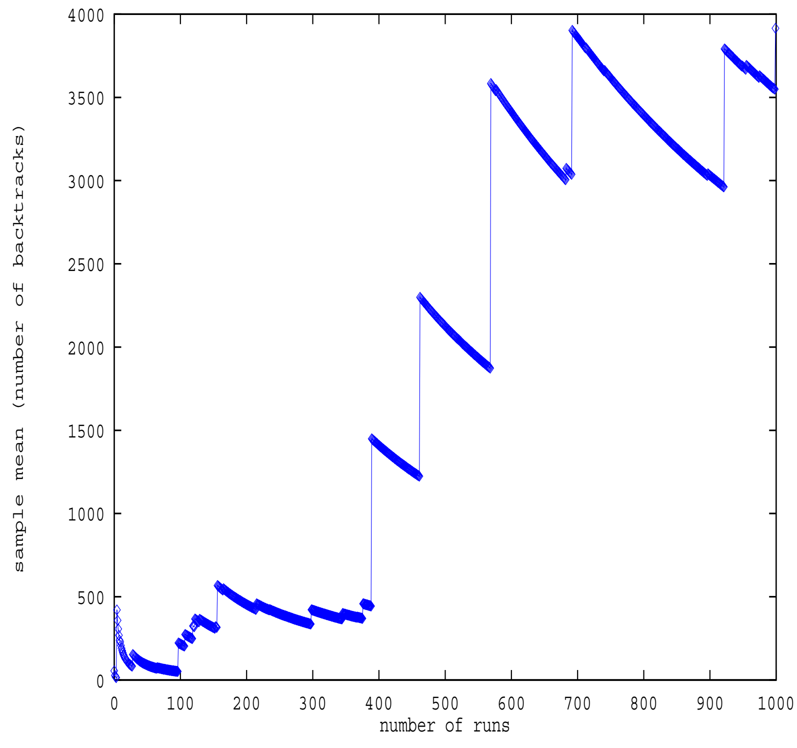
Observation:

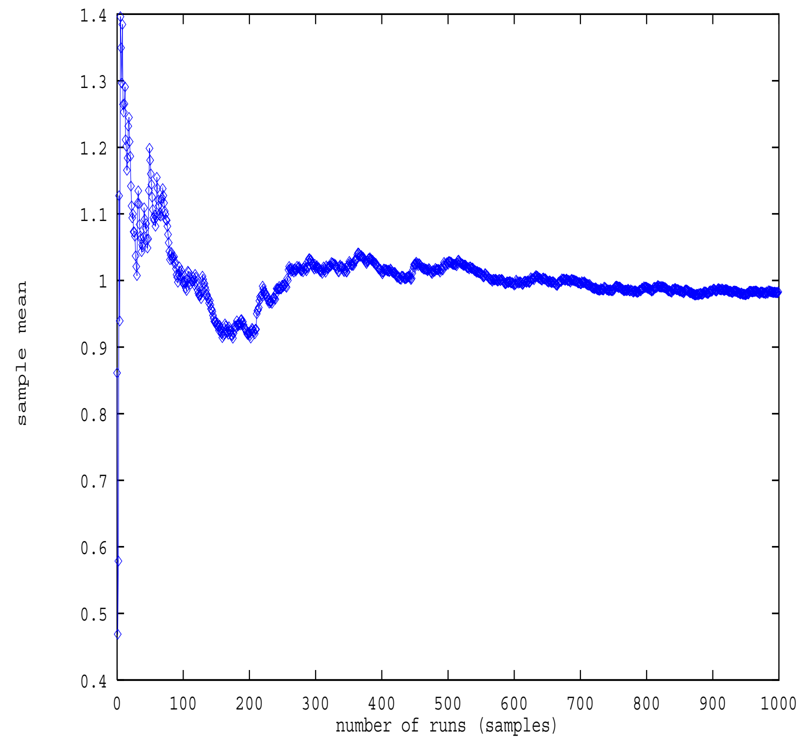
Typical deterministic systematic search algorithms perform abysmally bad on certain problem instances.

Intuitive Explanation:

Incorrect heuristic choices early in the search process can force search process to fully explore large parts of the search tree.

Erradic (left) *vs.* Stable (right) Estimation of Mean Run-Time





Randomisation & Restart:

- *Randomisation of heuristic choices* allows correct choices to be made against (incorrect) heuristic guidance.
- *Restart mechanism* helps to overcome stagnation (similar to restart in SLS)

Stochastic Tree Search (STS)

Key ideas:

- Modify systematic search algorithm using randomisation and restart.
- Restart replaces backtracking.

Note:

- Resulting algorithms are probabilistically approximately complete (PAC), but *not* complete. (Why?)
- Idea is closely related to Iterated Construction Search (ICS)

Example: Isamp [Crawford & Baker, 1994]

- STS algorithm for SAT, derived from (high-performance) David Putnam (DP) variant
- restarts search whenever assignment cannot be further expanded (contradiction)
- random choice of variable and value to assign at each step
- uses unit propagation (like DP)
- shown to perform well (compared to high-performance SLS / systematic search algorithms) on certain types of SAT encoded-scheduling problems (with many solutions)

Variants of STS / STS Algorithms:

- Greedy Adaptive Randomised Search Procedures (GRASP)
- Heuristic-Biased Stochastic Sampling (HSBS) [Bresina, 1996]
- Adaptive Probing [Ruml, 2001]
- ...

Randomised Systematic Search (RSS)

Key ideas:

- Modify systematic search algorithm using randomisation and restart.
- Use backtracking and iteratively increasing restart cutoff to maintain completeness.

(First proposed and investigated by Carla Gomes *et al.*)

Example: Randomised Davis-Putnam (DP) Algorithm for SAT

- *randomise* selection of variable to be instantiated next and/or order of instantiations (with truth values)
- *restart* search (from root of search tree) after fixed number θ of choices/backtracks

Preserve completeness by ...

- keeping track of previous choices along search path ensures complete exploration of tree for sufficienty high θ
- iteratively increasing search cutoff θ allows full tree search after fixed, instance-dependent number of iterations

Randomisation of Heuristic Choices

Note: Most systematic search alg extend partial candidate solutions based on heuristic function; ties are broken deterministically.

Key idea: Randomise tie-breaking

Problem: Good heuristics rarely produce ties.

Solution: Randomise over *heuristically equivalent choices*; two choices are heuristically equivalent iff their scores are within $H\%$ of the highest score (over all choices); parameter H controls degree of randomisation.

Characteristic Behaviour of RSS

Empirical Observations:

- Distribution of search cost for deterministic systematic search over certain sets of randomly generated problem instances has very high variance, erradic mean.
  (Due to rare outlier instances with extremely high search cost.)
- Same type of “heavy-tailed” distribution is encountered when measuring RTDs for RSS on individual instances.

(Hypothesised) Reason:

Outliers in search cost and run-time distributions are caused by incorrect heuristic choices early in the search
(often depending on syntactic aspects of problem instances, such as order of variable appearance in a CNF formula)

Consequence:

Using restart mechanism reduces variance in run-time of RSS, and decreases mean (by eliminating extremely long runs) for individual instances as well as random instance distributions
$\leadsto$ increased efficiency and robustness

(This result can be analytically proven for any situation in which the RTD of a given algorithm shows search stagnation, *i.e.*, falls below an exponential distribution fitted from the left.)

Polynomial Decay in the Right Tail

Definition: A probability distribution with CDF F(x) shows polynomial decay in the right tail iff

$$\lim_{x\to\infty} (1 - F(x))/Cx^{-\alpha} = 1, \quad x > 0$$

for some constants $C > 0, 2 > \alpha > 0$.

Equivalently:

$$1 - F(x) \sim Cx^{-\alpha}, \quad x > 0$$

for some constants $C > 0, 2 > \alpha > 0$.

These distributions are often called "heavy-tailed".

Graphical Characterisation:

In log-log plot of $1 - F(x)$, right tail asymptotically approaches a straight line for $x \to \infty$.

(The slope of that line provides estimate for α.)

Note:

For RTD with cdf $F(x)$, $1 - F(x) = Pr\{RT > x\}$
(failure probability for cutoff x).

Distribution types that *don't* show polynomial decay in right tail:

- Normal (Gaussian) distribution
- Exponential distribution
- Weibull distribution
- ...

(In fact, all of these show expontial decay in the right tail.)

Distribution types that *do* show polynomial decay in right tail:

- Pareto distribution, CDF:

$$F(x) = 1 - 1/x^{\alpha}$$

- Cauchy distribution, PDF:

$$f(x) = 1/\pi \cdot \gamma/(\gamma^2 + (x - \delta)^2)$$

- Lèvy distribution, PDF:

$$f(x) = \sqrt{\gamma/(2\pi)} \cdot (x - \delta)^{-3/2} \cdot e^{-\gamma/(2(x-\delta))}$$

Such “heavy-tailed” distributions have been used for empirically modelling a range of phenomenae, including certain properties of random walks and traffic in communication networks.

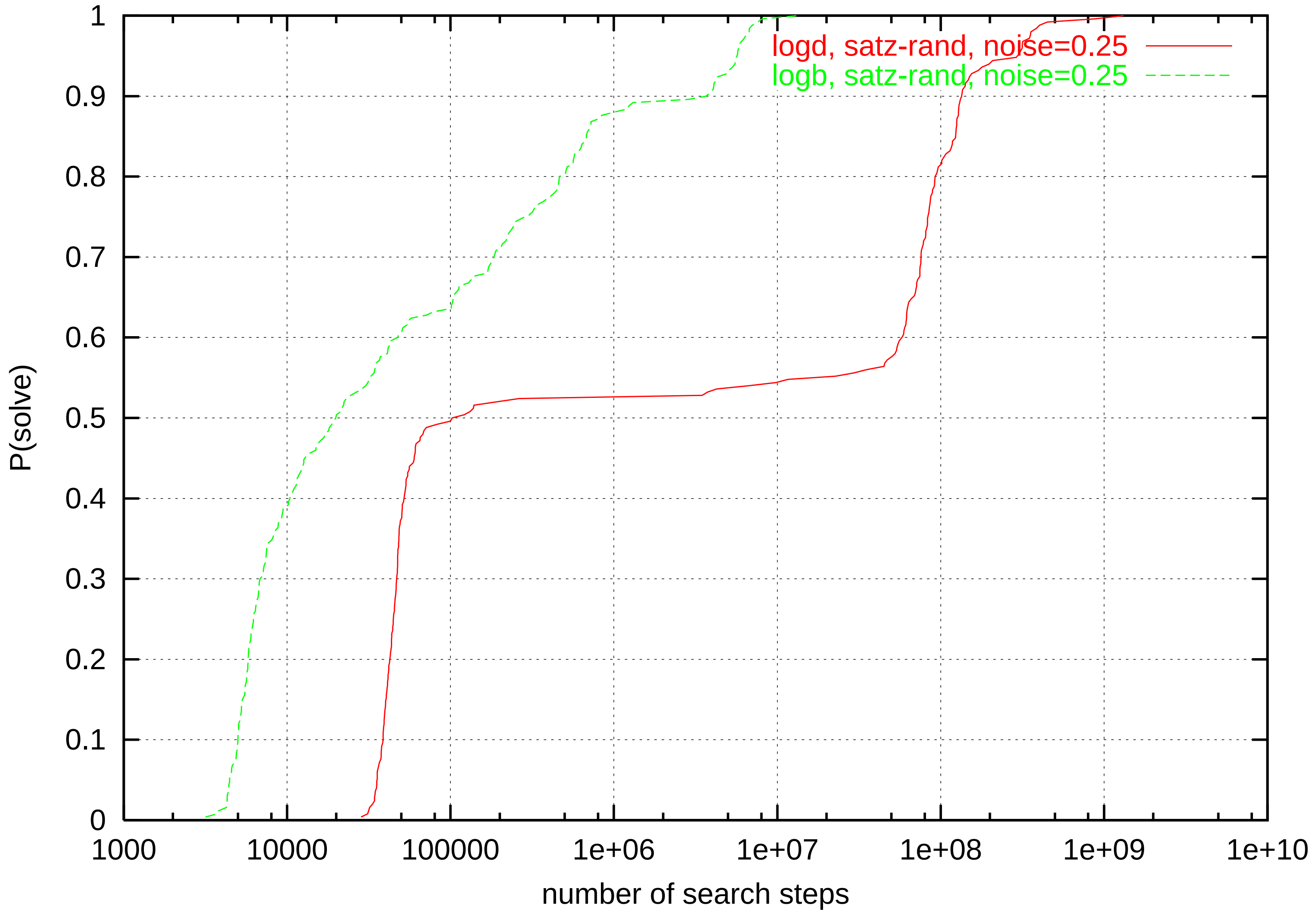
Some Properties of Distributions with "Heavy" Right Tails

- $2 > \alpha > 1$: finite mean, infinite variance
- $1 \geq \alpha > 0$: infinite mean, infinte variance
  (*e.g.*, Cauchy, Lèvy distributions)

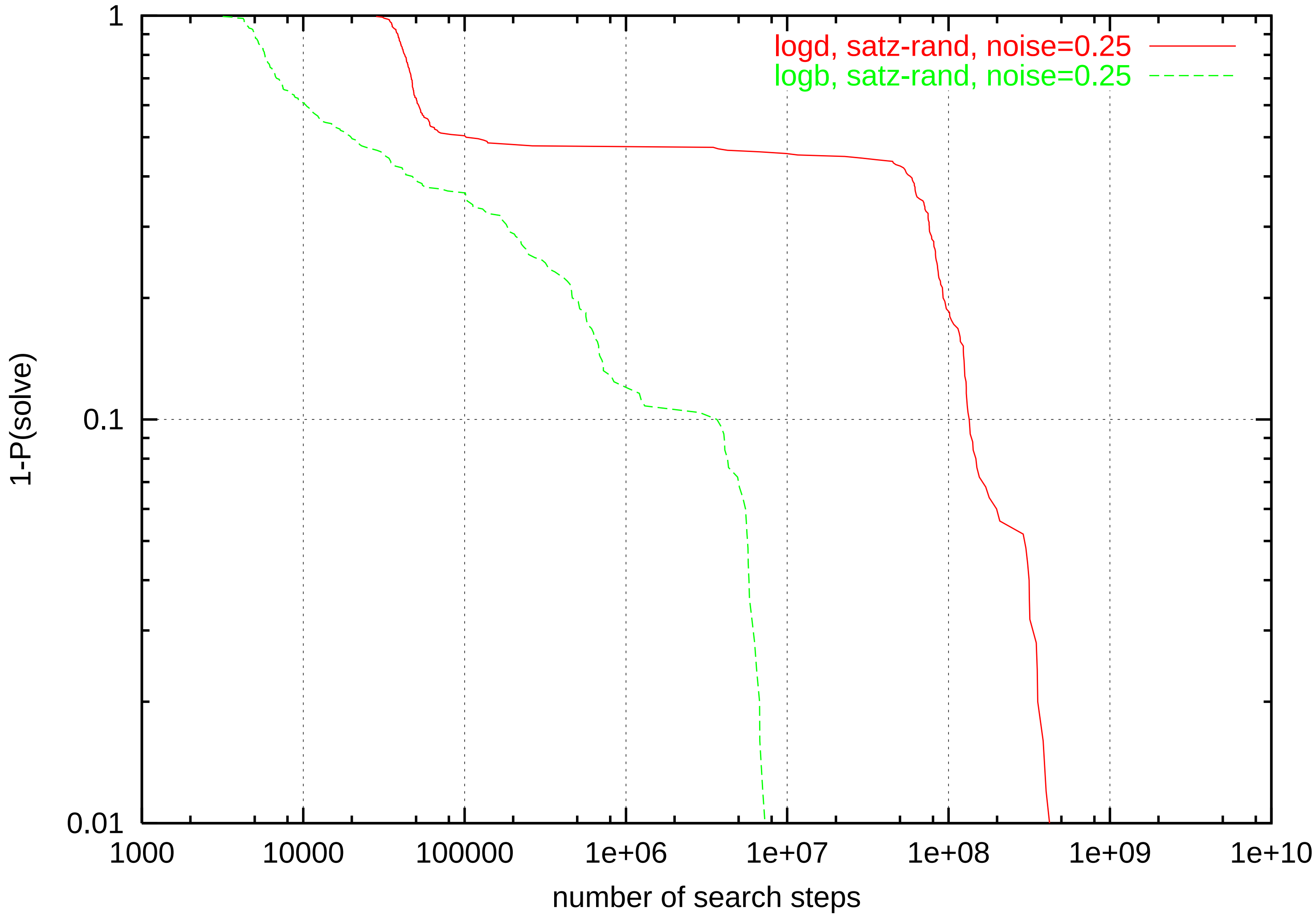
Parameter α is also called *index of stability*.

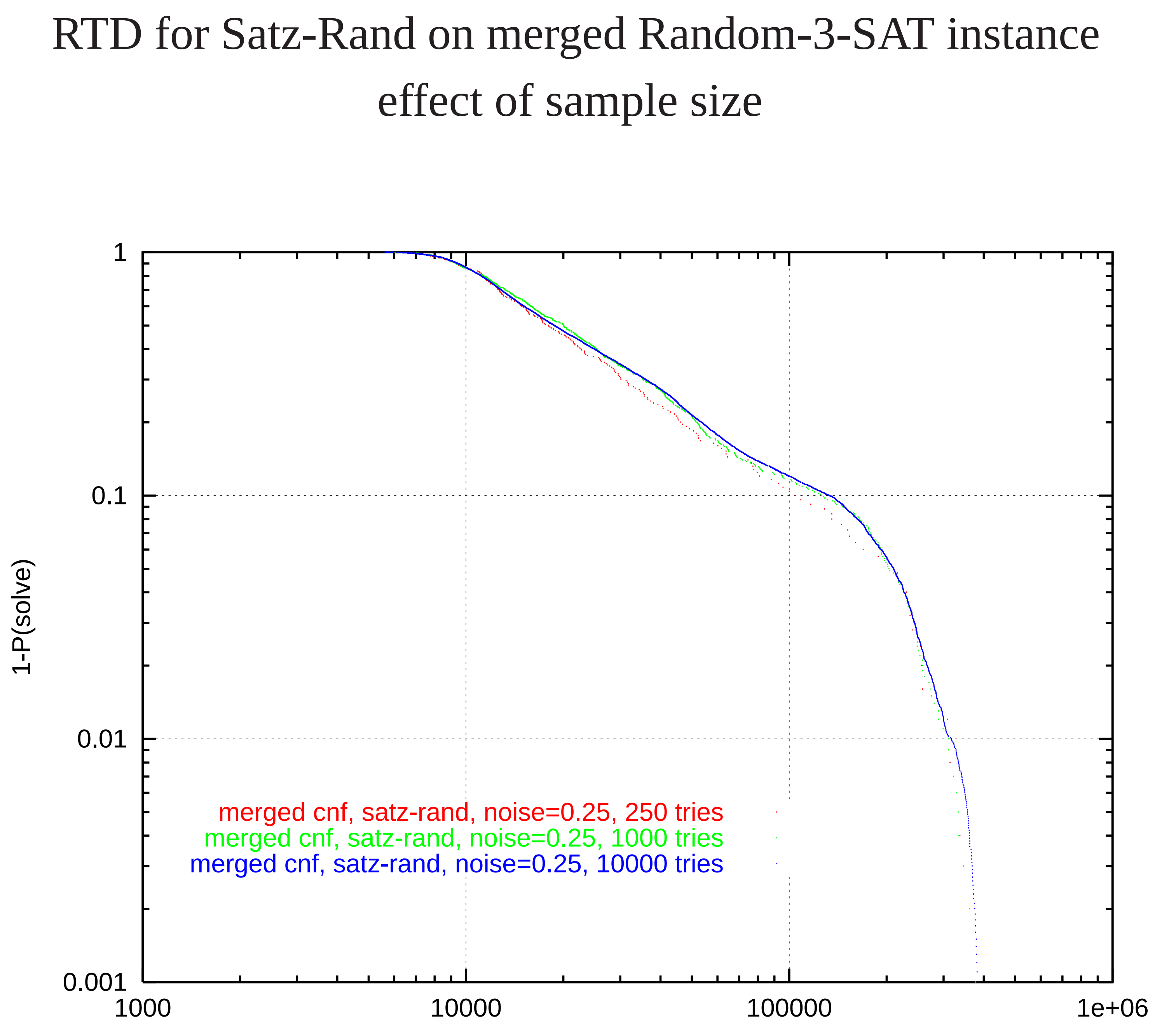
Note: Actual RTDs allways have finite mean and variance. (Why?)

RTDs of Satz-Rand on two SAT-encoded Logistics Planning instances



RTDs of Satz-Rand on two SAT-encoded Logistics Planning instances (right tails)




RTD for Satz-Rand on merged Random-3-SAT instance
effect of sample size
1-P(solve)
1
0.1
0.01
0.001
1000
10000
100000
1e+06
merged cnf, satz-rand, noise=0.25, 250 tries
merged cnf, satz-rand, noise=0.25, 1000 tries
merged cnf, satz-rand, noise=0.25, 10000 tries

Polynomial Decay in the Left Tail

(Analogous to polynomial decay in the right tail)

Definition: A probability distribution with CDF F(x) shows polynomial decay in the left tail iff

$$\lim_{x \to 0} F(x)/Cx^{\alpha} = 1, \quad x > 0$$

for some constants $C > 0, \alpha > 0$.

Graphical Characterisation: In log-log plot, left tail asymptotically approaches a straight line for $x \to 0$.

Weibull Distributions

Generalisation of exponential distribution.

Cumulative distribution function (CDF):

$$wd[m, \beta](x) = W(x, m, \beta) = 1 - 2^{-(x/m)^{\beta}}$$

Parameters:

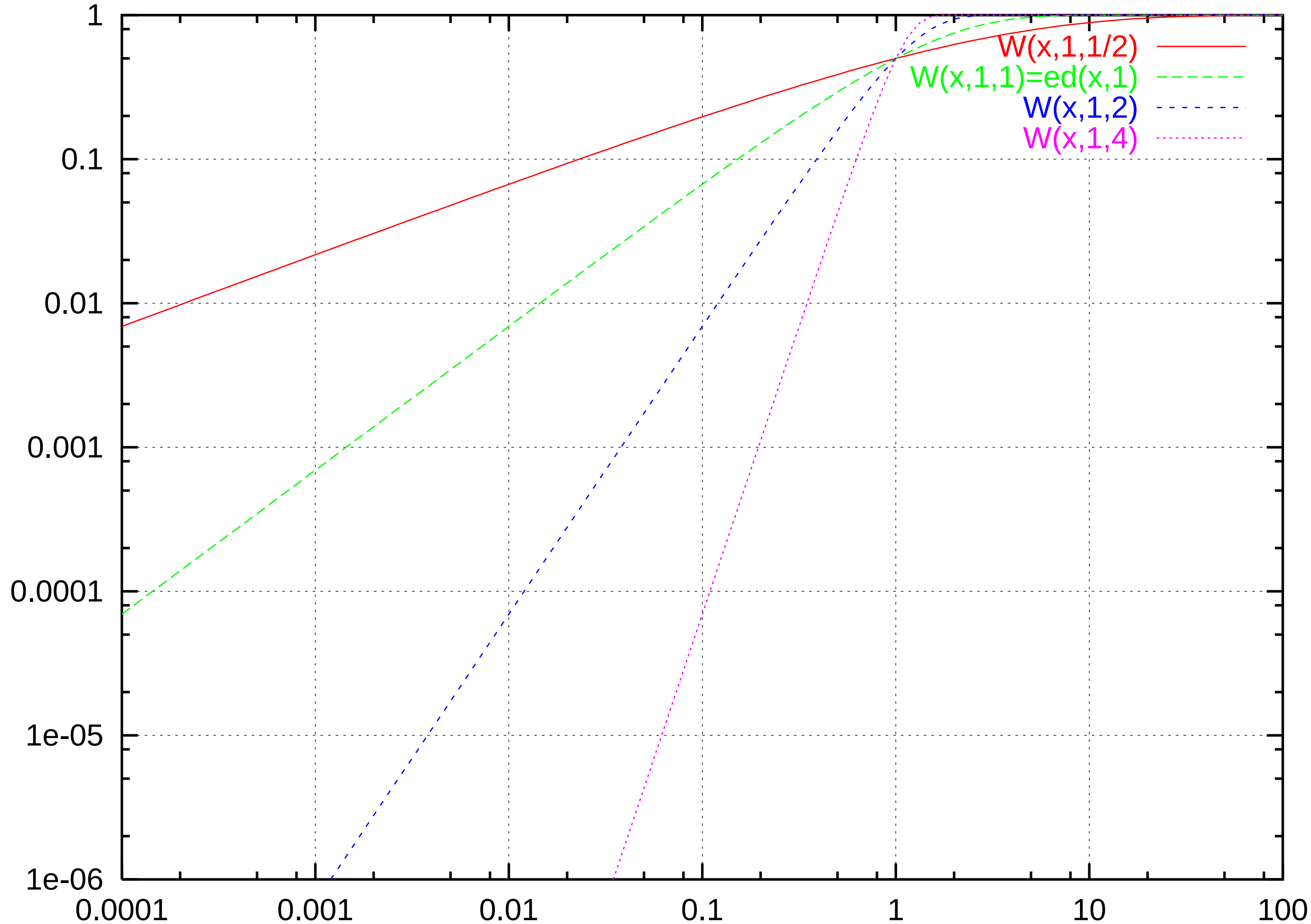
- m: median
- β: controls the variation coefficient (stdddev/mean).

Fact (provable):

All Weibull distributions have polynomial decay in the left tail.

Left tails of Weibull distributions $W(x, m, \beta)$

for different values of shape parameter β



GED Mixtures Characterise RSS Behaviour

Generalised Exponential Distributions (GEDs)

Generalisation of exponential distribution, originally developped for characterising typical RTDs of SLS algorithms.

Cumulative distribution function (CDF):

$$ged[m, \gamma, \delta](x) = wd[m, 1 + (\gamma/x)^{\delta}](x) = 1 - 2^{-(x/m)^{1+(\gamma/x)^{\delta}}}$$

Facts (provable):

- The right tail of any GED asymptotically approaches that of an exponential distribution.
- The left tail of any GED with $\gamma > 0$ does *not* show polynomial decay.

Mixtures of Generalised Exponential Distributions

Cumulative distribution function (CDF):

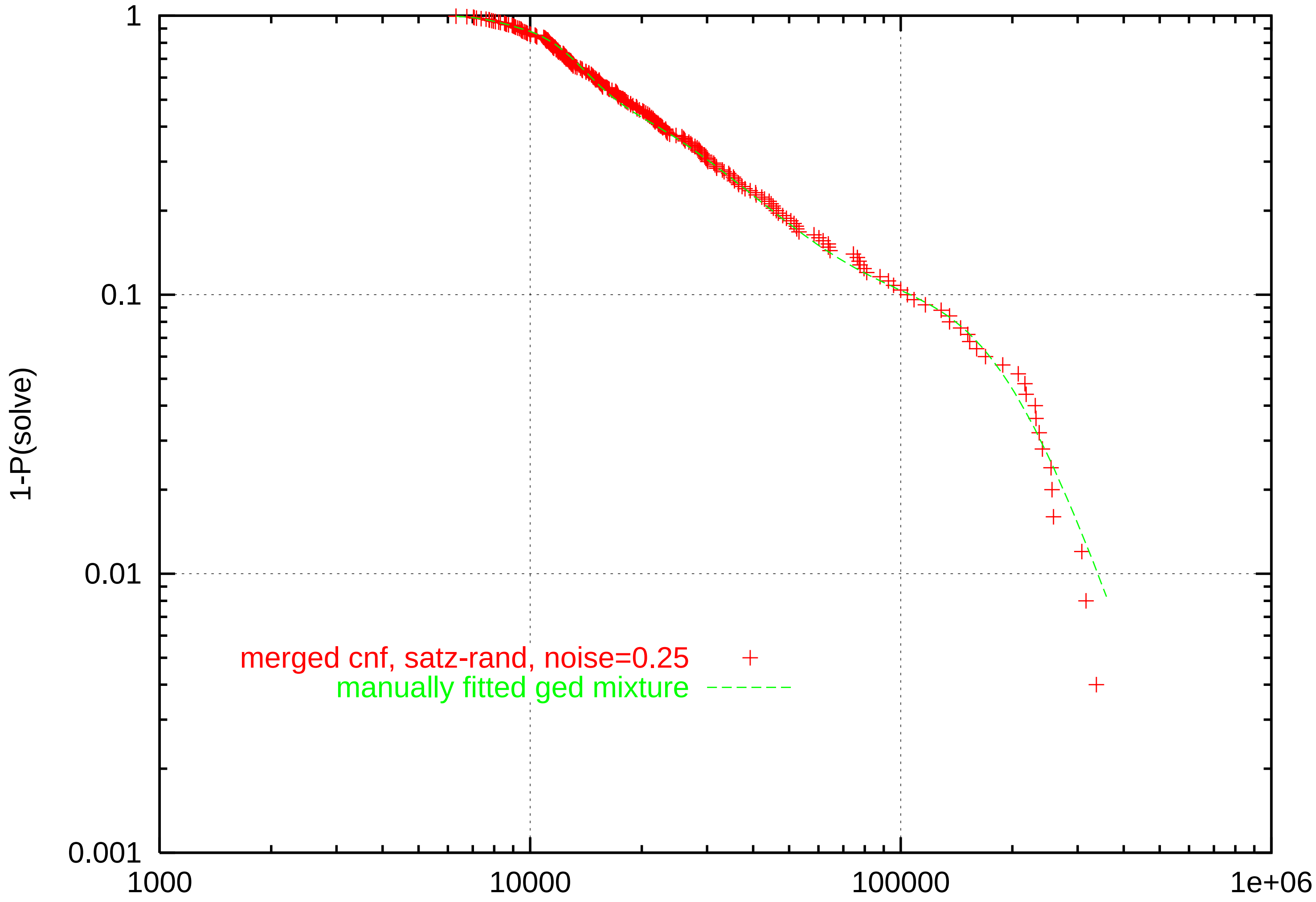
$$\sum_{i=1}^{\nu} c_i \cdot ged[m_i, \gamma_i, \delta_i](x)$$

(Developped and used for characterisation of irregular RTDs for SLS algorithms.)

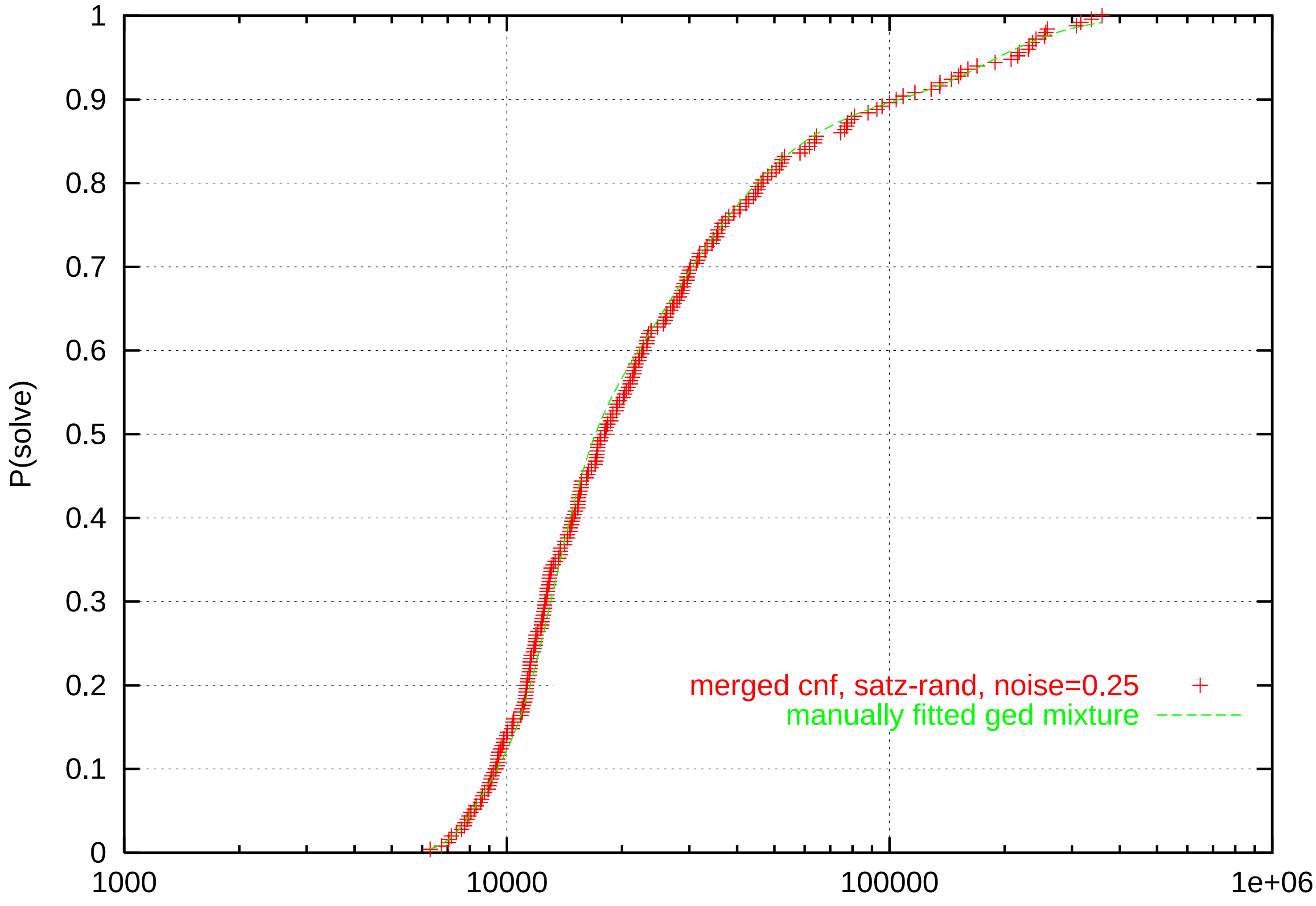
Facts (provable):

- The right tail of any finite GED mixture asymptotically approaches that of an exponential distribution.
- The left tail of any GED with $\gamma > 0$ does *not* show polynomial decay.
- GED mixtures with an *infinite* number of components can have polynomial decay in their right tails.

RTD for Satz-Rand on merged Random-3-SAT instance
appproximation with GED mixture (right tail)



RTD for Satz-Rand on merged Random-3-SAT instance appproximation with GED mixture (entire distribution)



Empirical Results:

- GED mixtures with a small number of components yield very good approximations of the RTDs observed for Randomised Systematic Search algorithms.
- Different from previously used “heavy-tailed” distributions (such as Pareto or Lèvy), these approximations capture the entire distribution.
- GED mixtures appear to provide a unified model for characterising the run-time behaviour (RTDs) of RSS and SLS algorithms.
- Results on the effectiveness of restart still apply.

Pros and Cons of RSS Algorithms

Pros:

- increased robustness, in particular when using suitably tuned noise and restart strategies
- simple, generic extension of systematic search
- resulting algorithms typically still complete
- potential for easy parallelisation

Cons:

- highly stochastic behaviour
- difficult to analyse theoretically / empirically
- parameter tuning often difficult,
  but critical for obtaining good performance

Summary

- Stochastic tree search and randomised systematic search are two relatively new and little studied classes of stochastic search algorithms.
- There is limited evidence that randomisation and restart techniques can improve the robustness of systematic search behaviour.
- An increasing number of state-of-the-art systematic search algorithms (especially for SAT) use randomisation & restart.
- Many issues surrounding stochastic tree search, randomised systematic search, and “heavy-tailed” behaviour are not fully understood and need further research.

Important Concepts:

- stochastic tree search (STS)
- randomised systematic search (RSS)
- heuristic equivalence
- polynomial decay (“heavy-tailed”) distributions
- completeness preserving restart strategies for RSS
- mixtures of (generalised) exponential distributions

Further Readings

- J.M. Crawford and A.B. Baker: Experimental Results on the Application of Satisfiability Algorithms to Scheduling Problems. Proc. of the AAAI-94, pp. 1092–1097, 1994.
- J.L. Bresina. Heuristic-biased stochastic sampling. Proc. of AAAI-96, pp. 271-278, 1996.
- Work by Carla Gomes *et al.*, in particular:
  - C. Gomes, B. Selman, H. Kautz: Boosting combinatorial search through randomization. Proc. AAAI-98, pp. 431–437, 1998.
  - C. Gomes, B. Selman, N. Crato, H. Kautz: Heavy-Tailed Phenomena in Satisfiability and Constraint Satisfaction Problems. Journal of Automated Reasoning, 2000.

- H.H. Hoos: Heavy-Tailed Behaviour in Randomised Systematic Search Algorithms for SAT? Technical Report TR-99-16. Department of Computer Science, University of British Columbia, 1999.

- Work by Wheeler Ruml, in particular:
  - W. Ruml: Stochastic Tree Search: Where to Put the Randomness? Proc. of IJCAI-01 Workshop on Stochastic Search Algorithms, 2001.
  - W. Ruml: Incomplete tree search using adaptive probing. Proc. of IJCAI-01, 2001.