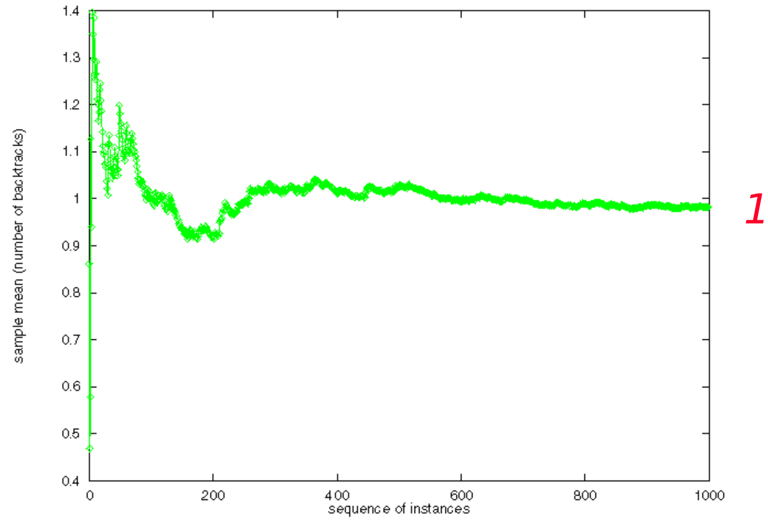
Effects of randomization

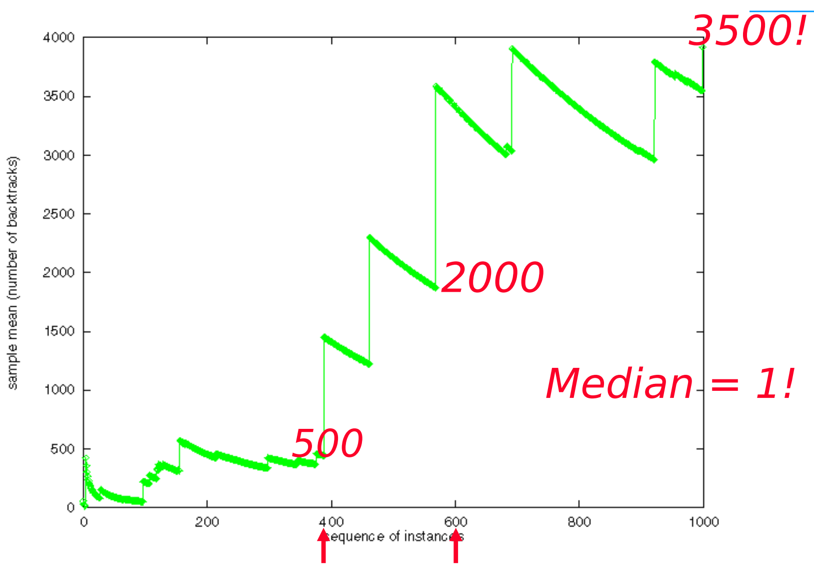
- *Studi su randomizzazione algoritmi di Carla Gomes, Bart Selman e colleghi*
- *Inserire una scelta casuale nella search*
  - *Selezione variabile, selezione valore o entrambi*
- *L'effetto casuale può essere anche piccolo*
  - *Es: uso un'euristica di selezione della variabile non random (first-fail, max-constrained, …), ma in caso di parità scelgo a caso*

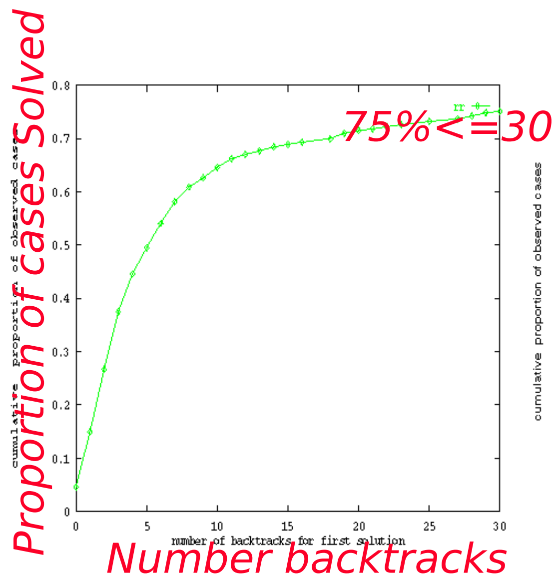
Eseguendo molte prove ci si aspetterebbe …

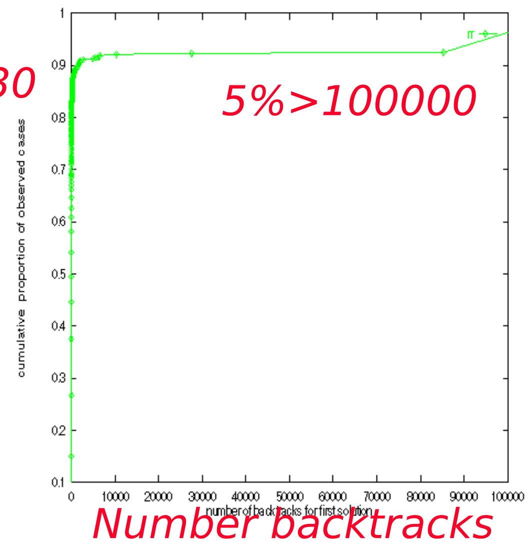


Standard Mean Cost Behavior (Gamma)

Erratic Behavior of Search Cost Quasigroup Completion Problem







Heavy-Tailed Behavior

Heavy-Tailed Distributions

... infinite variance ... infinite mean

Introduced by Pareto in the 1920's

--- "probabilistic curiosity."

Mandelbrot established the use of heavy-tailed distributions to model real-world fractal phenomena.

Examples: stock-market, earth-quakes, weather,...

Decay of Distributions

Standard --- Exponential Decay

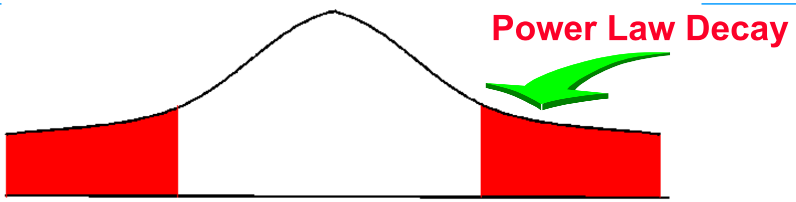
e.g. Normal:

$$\Pr[X>x] \approx Ce^{-x^2},\ \textit{for some } C>0, x>1$$

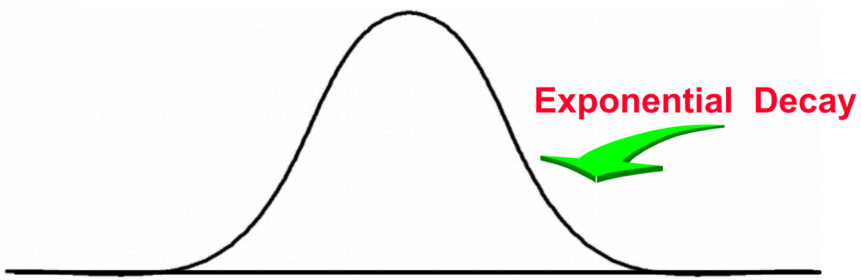
Heavy-Tailed --- Power Law Decay

e.g. Pareto-Levy:

$$\Pr[X>x] = Cx^{-\alpha},\ x>0$$

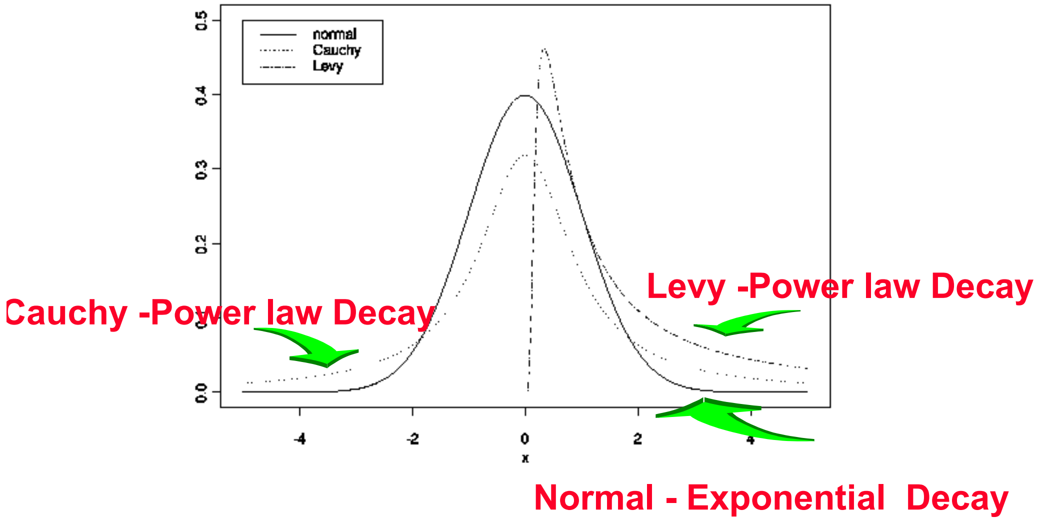


HEAVY TAILED DISTRIBUTION
(infinite mean & variance)



Standard Distribution
(finite mean & variance)

Normal, Cauchy, and Levy



How to Check for "Heavy Tails"?

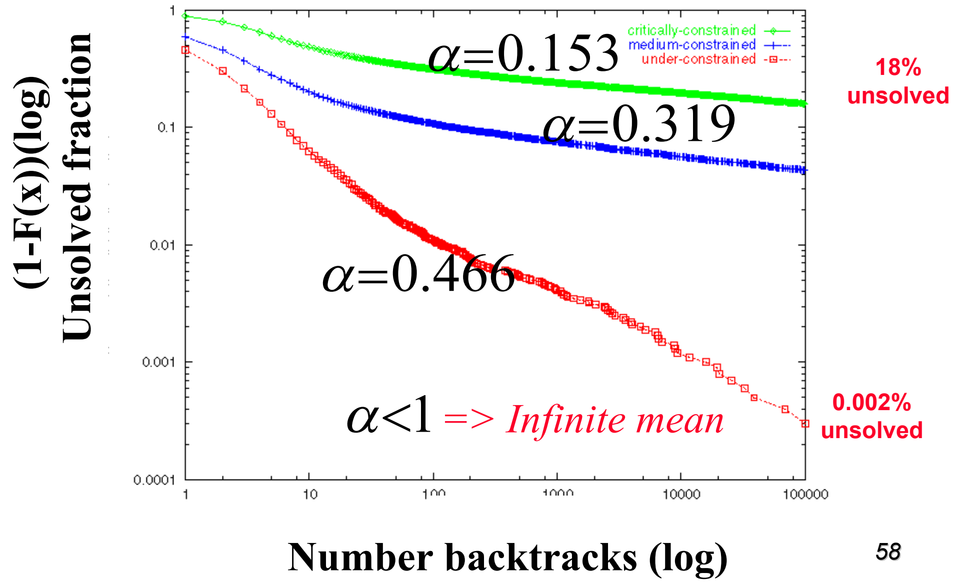
Log-Log plot of tail of distribution should be approximately linear.

Slope gives value of α

$\alpha < 1$ ***infinite mean and infinite variance***

$1 \leq \alpha < 2$ ***infinite variance***

Heavy-Tailed Behavior in QCP Domain



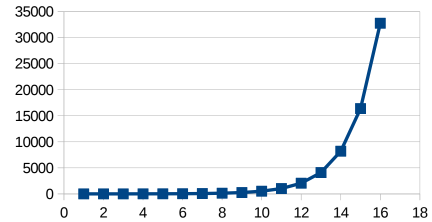
Completezza

- *Facendo ripartire la ricerca, come garantire la completezza?*
- *Ad es, se il problema non ha soluzioni, quando si smette di fare restarts?*
- *Una soluzione* parziale *è il clause learning: nei restart non si cancellano le clausole apprese, quindi non si ri-esplora lo stesso spazio di ricerca*
  - *Però se si mantengono tutte le clausole, l'occupazione di memoria diventa eccessiva: ogni tanto devo comunque dimenticare qualche clausola appresa*
- *Un'altra soluzione è aumentare via via il periodo dei restart, in modo che prima o poi in un unico restart si esplori tutto lo spazio di ricerca*

Dopo quanto tempo fare restart?

- *Progressione geometrica*

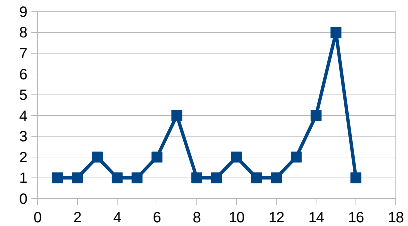
$$t_i = b^i$$



- *Luby et al. [1993]*

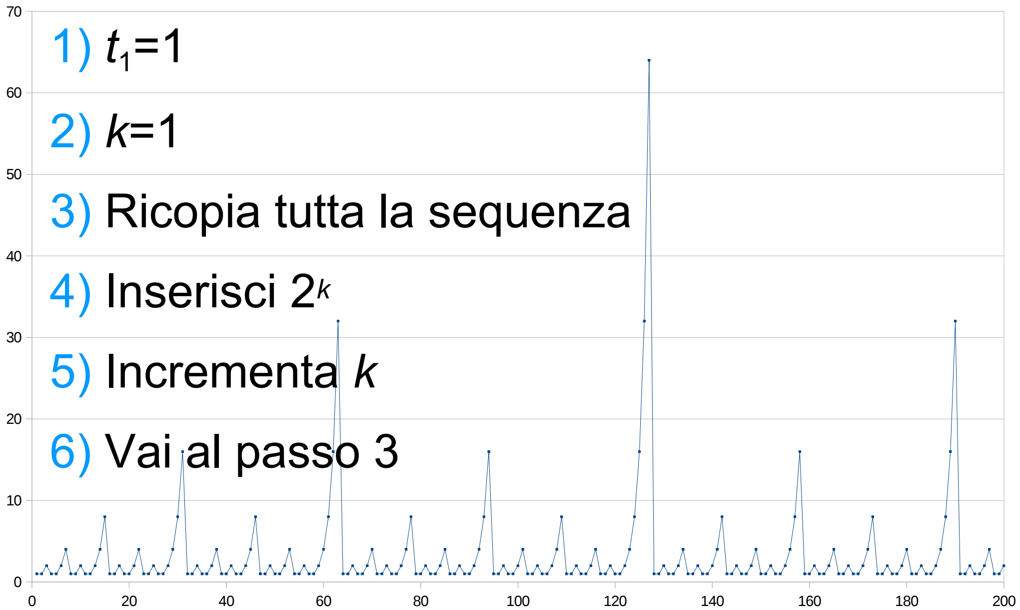
$$t_i = \begin{cases} 2^{k-1} & \text{se } i = 2^k - 1 \\ t_{i-2^{k-1}+1} & \text{se } 2^{k-1} \leq i < 2^k - 1 \end{cases}$$

Ottima quando la distribuzione è ignota



Come generare la sequenza Luby et al.

1) t_1=1
2) k=1
3) Ricopia tutta la sequenza
4) Inserisci 2^k
5) Incrementa k
6) Vai al passo 3



Encodings CSP → SAT

- *Advantages*
  - *Efficient SAT solvers*
  - *Common interface (DIMACS file)*
  - *After conversion, the problem can be solved with new solvers, adopting new technologies*
  - *No need to re-implement in CSP complex search strategies, heuristics, … (intelligent backtracking, learning, restarts, …)*
- *Problems*
  - *Structure of the problem is typically lost*

Direct Encoding

- *A SAT variable for each domain element*

X0,X1,X2 – Y0,Y1,Y2

- *at-least-one*

  X0 V X1 V X2

- *at-most-one*

  - X0 V -X1

  -X0 V -X2

  -X1 V -X2

- *conflict* *-X1 V -Y1*

- *X, Y :: 0..2, X>Y*

	Y=0	Y=1	Y=2
X=0	0	0	0
X=1	1	0	0
X=2	1	1	0

Direct Encoding in generale

- Dato un CSP con n variabili, ciascuna con un dominio di dimensione d
- Per ogni variabile X
  - d variabili SAT: X_1, X_2, … X_d
  - Una clausola *"at least one"* di lunghezza d

    $$X_1 \vee X_2 \vee \ldots \vee X_d$$

  - $d(d-1)/2$ clausole "at most one" di lunghezza 2
  - Per ogni i,j da 1 a d: $-X_i \vee -X_j$
- Per ogni vincolo binario $c(X,Y)$
  - per ogni coppia di assegnamenti $X \leftarrow v$, $Y \leftarrow w$ che non soddisfa il vincolo

    $$-X_v \vee -Y_w$$

Dicevamo ...

- Per ogni variabile X

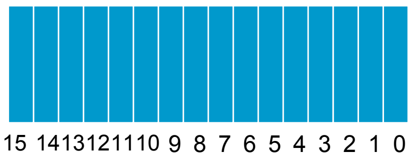
  d variabili SAT: $X_1, X_2, \ldots X_d$

- Però per descrivere il problema (CSP), devo dire che le variabili hanno dominio da 1 a d
- Per scrivere il numero d servono $\log_2 d$ bit
- Quindi con questa codifica si usa un numero esponenziale di variabili SAT rispetto al numero di bit che servono per descrivere il problema!

Direct Encoding

- *Conflict*

  $-X_5 \vee -Y_3$



if Y_3=true → X_5 must be false

if Y is assigned value 3 → you can remove value 5 from X's domain

DPLL performs pruning of Forward Checking

Support Encoding (Gent)

- *A SAT variable for each domain element*

  $X_0, X_1, X_2 - Y_0, Y_1, Y_2$

- *at-least-one*

  $X_0 \vee X_1 \vee X_2$

- *at-most-one*

  $-X_0 \vee -X_1$

  $-X_0 \vee -X_2$

  $-X_1 \vee -X_2$

- *Support* $X_2 \rightarrow Y_0 \vee Y_1$

- *X, Y :: 0..2, X>Y*

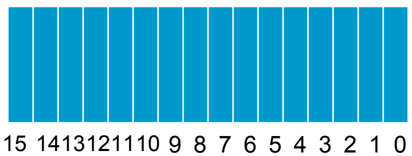
	Y=0	Y=1	Y=2
X=0	0	0	0
X=1	1	0	0
X=2	1	1	0

DPLL gives pruning of Maintaining Arc-Consistency on the original CSP

Support Encoding

- *Support*

  $-X_5 \vee Y_3 \vee Y_4 \vee Y_5$



- *if Y_3=false, Y_4=false, Y_5=false → X_5 must be false*
- *if all the values compatible with X=5 are deleted → you can remove value 5 from X's domain*

DPLL performs pruning of (Maintaining) Arc-Consistency

Support encoding: in generale

- Dato CSP con *n* variabili, ciascuna con un dominio di dimensione *d*
- Per ogni variabile *X*
  - *d* variabili SAT: $X_1, X_2, \ldots X_d$
  - Una clausola *"at least one"* di lunghezza *d*

    $X_1 \lor X_2 \lor \ldots \lor X_d$
  - $d(d-1)/2$ clausole *"at most one"* di lunghezza 2

    Per ogni *i,j* da 1 a *d*: $-X_i \lor -X_j$
- Per ogni vincolo binario c(*X*,*Y*)
  - per ogni valore $X \leftarrow v$,
    se i valori consistenti con *v* nel dominio di *Y* sono $w_1, w_2, \ldots w_k$

    $-X_v \lor Y_{w1} \lor Y_{w2} \lor \ldots \lor Y_{wk}$
  - Analogamente per ogni valore $Y \leftarrow w$

log-Encoding

- *Variables encoded with binary code*

$X_1, X_0 - Y_1, Y_0$

- ~~*at-least-one*~~
- ~~*at-most-one*~~
- *Prohibited value*

  $-X_0 \lor -X_1$
- *Conflict*

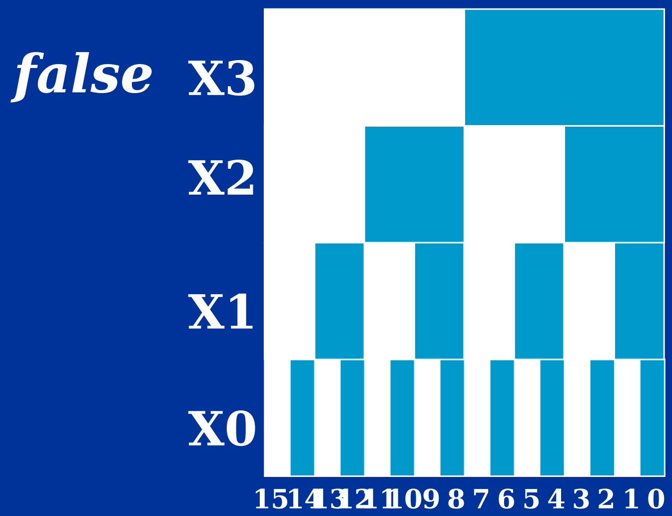
  $X_0 \lor X_1 \lor Y_0 \lor Y_1$

- *X, Y :: 0..2, X>Y*

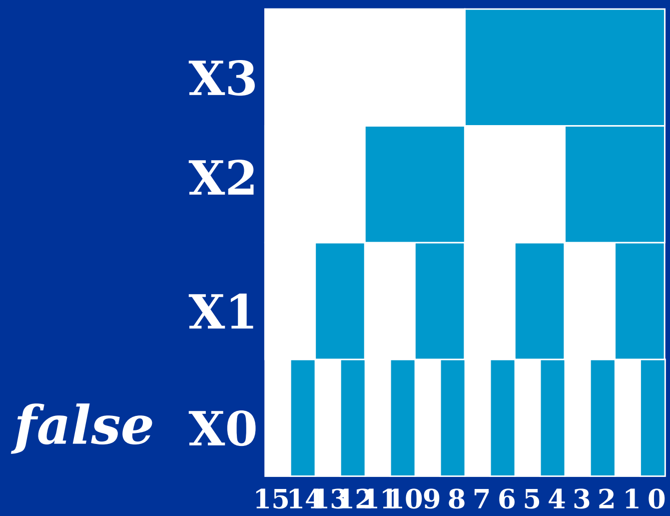
	Y=00	Y=01	Y=10
X=00	0	0	0
X=01	1	0	0
X=10	1	1	0

Pruning???

log-Encoding



log-Encoding



Log-encoding, in generale

- Dato un CSP con n variabili, ciascuna con un dominio di dimensione d. Sia $m=\lceil log_2 d \rceil$, (arrotondato per eccesso)
- Per ogni variabile X
  - m variabili SAT: $X_1, X_2, \ldots X_m$
  - Per ogni valore v (rappresentato in binario con $v_0 \ldots v_{m-1}$, ovvero $v=\sum_b 2^b v_b$) non ammesso nel dominio di X (es. Se m non è una potenza di 2, per cui $d<2^m$) una clausola *"prohibited value"* di lunghezza m

$$\bigvee_b v_b \; XOR \; X_b$$

- Per ogni vincolo binario $c(X,Y)$
  - per ogni coppia di assegnamenti inconsistenti $X \leftarrow v,\ Y \leftarrow w$

$$(\bigvee_b v_b \; XOR \; X_b) \vee (\bigvee_i w_i \; XOR \; Y_i)$$