The analysis of BP guided decimation algorithm

Federico Ricci-Tersenghi
Physics Department
Sapienza University, Roma

FRT, G. Semerjian, JSTAT (2009) P09001
A. Montanari, FRT and G. Semerjian, Proc. Allerton (2007) 352

Motivations

- Solving algorithms are of primary relevance in combinatorial optimization
  -> provide lower bounds
  -> their behavior is related to problem hardness

- Analytical description of the dynamics of solving algorithms is difficult

- Can we link it to properties of the solution space ?

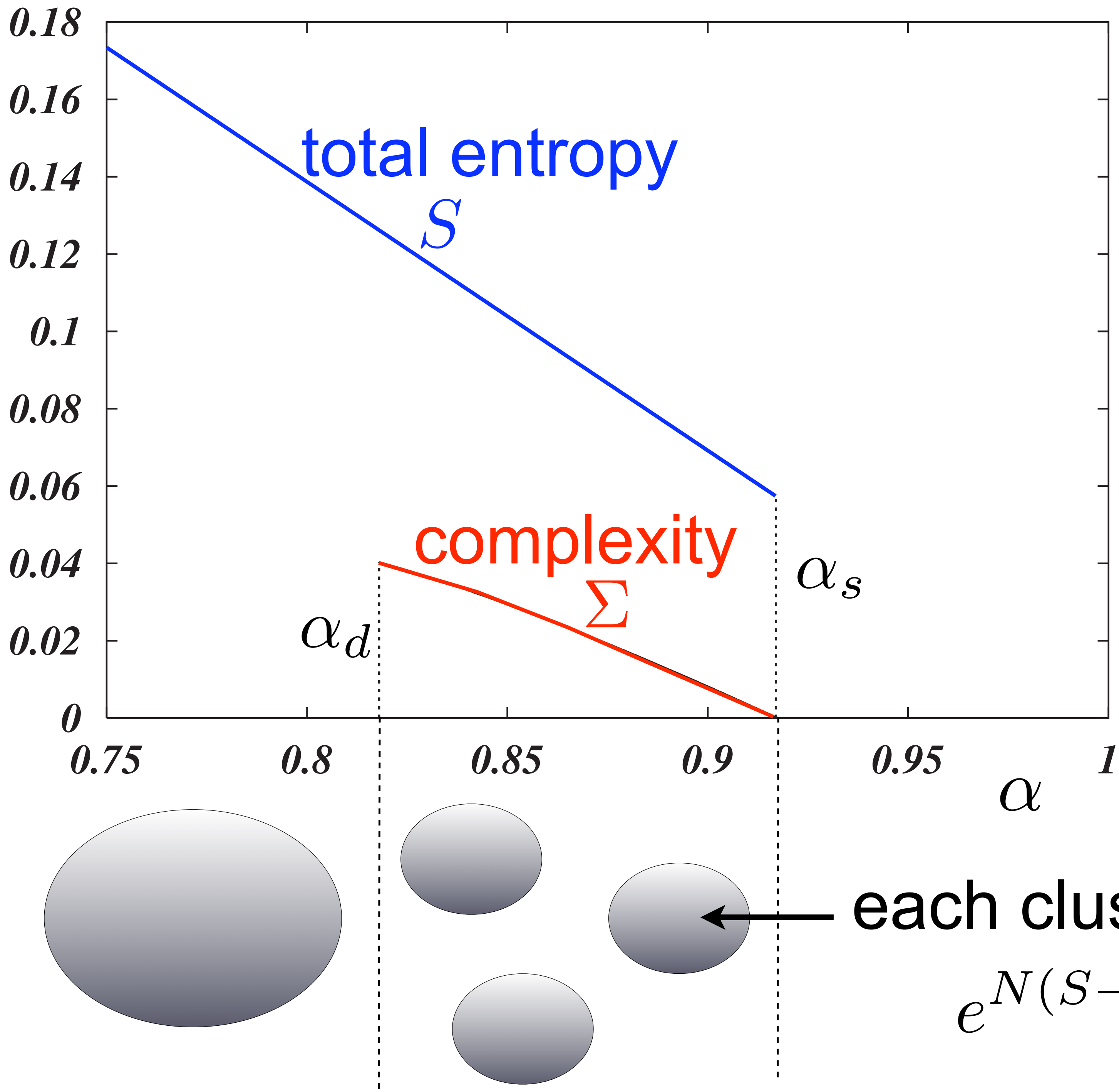
- Is there a threshold unbeatable by any algorithm ? (kind of first principles limitation...)

Models and notation

- Random k-XORSAT (k=3)
- Random k-SAT (k=4)

- Notation:
  - N variables, M clauses
  - Clause to variables ratio $\alpha = M/N$

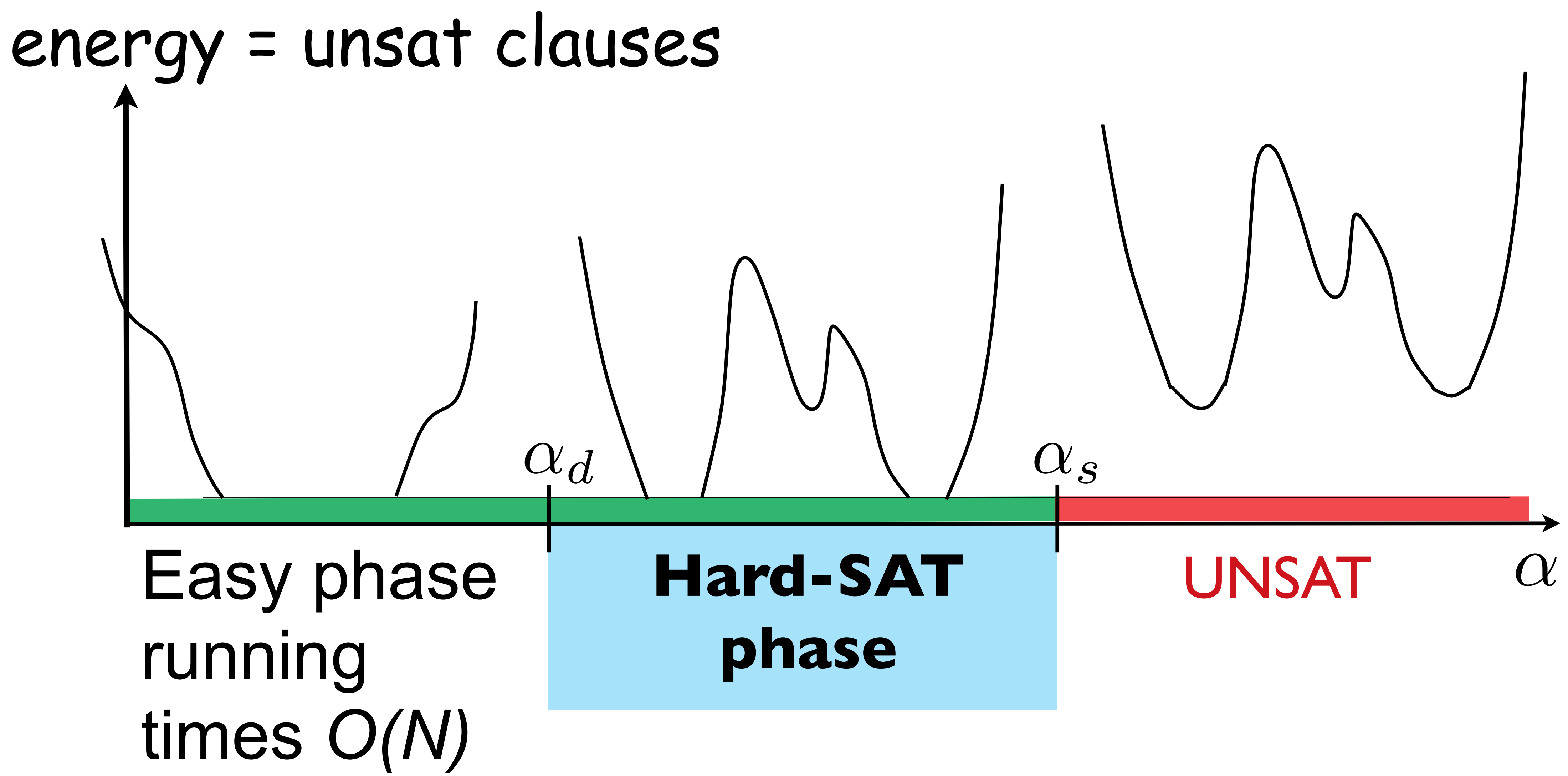
Phase transitions in random CSP



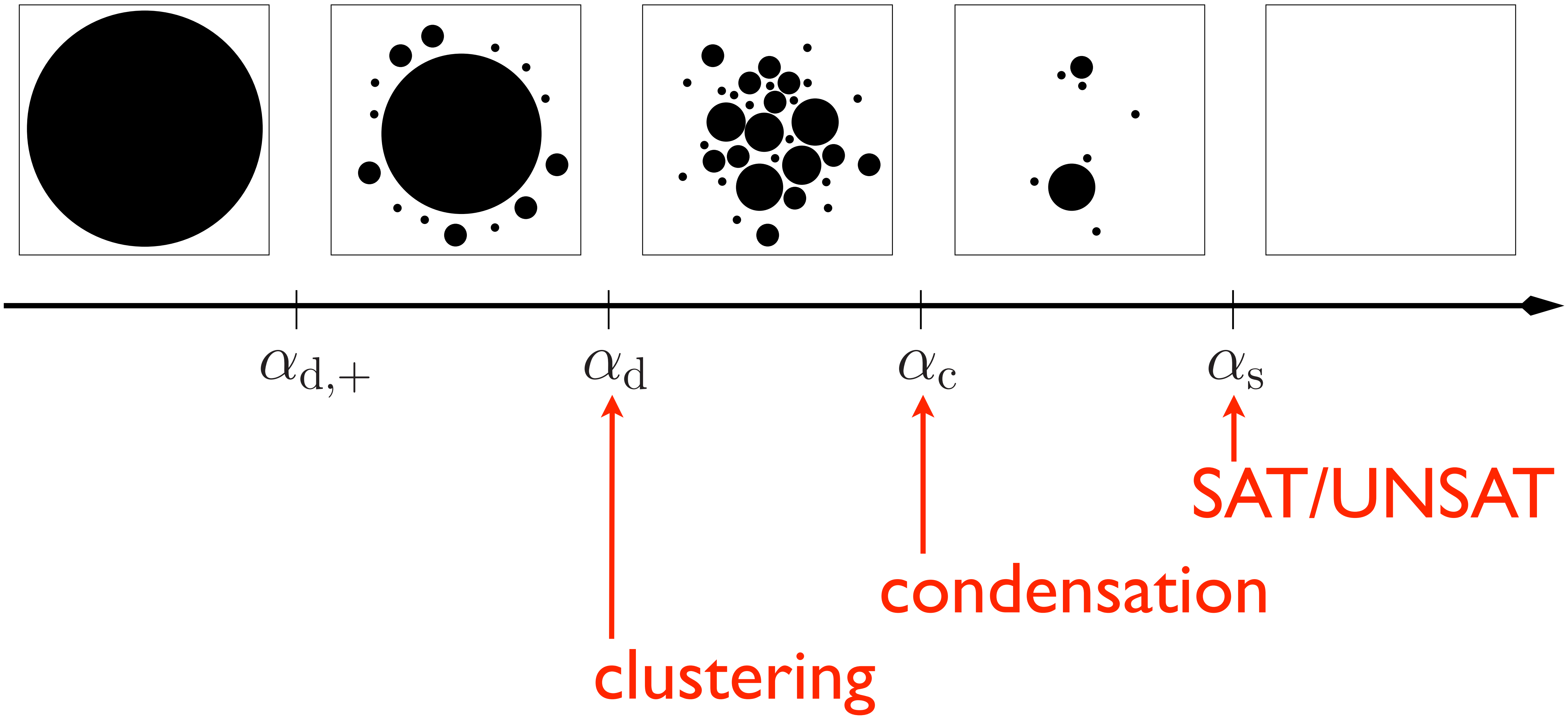
each cluster contains $e^{N(S-\Sigma)}$ solutions

clustering phase transition

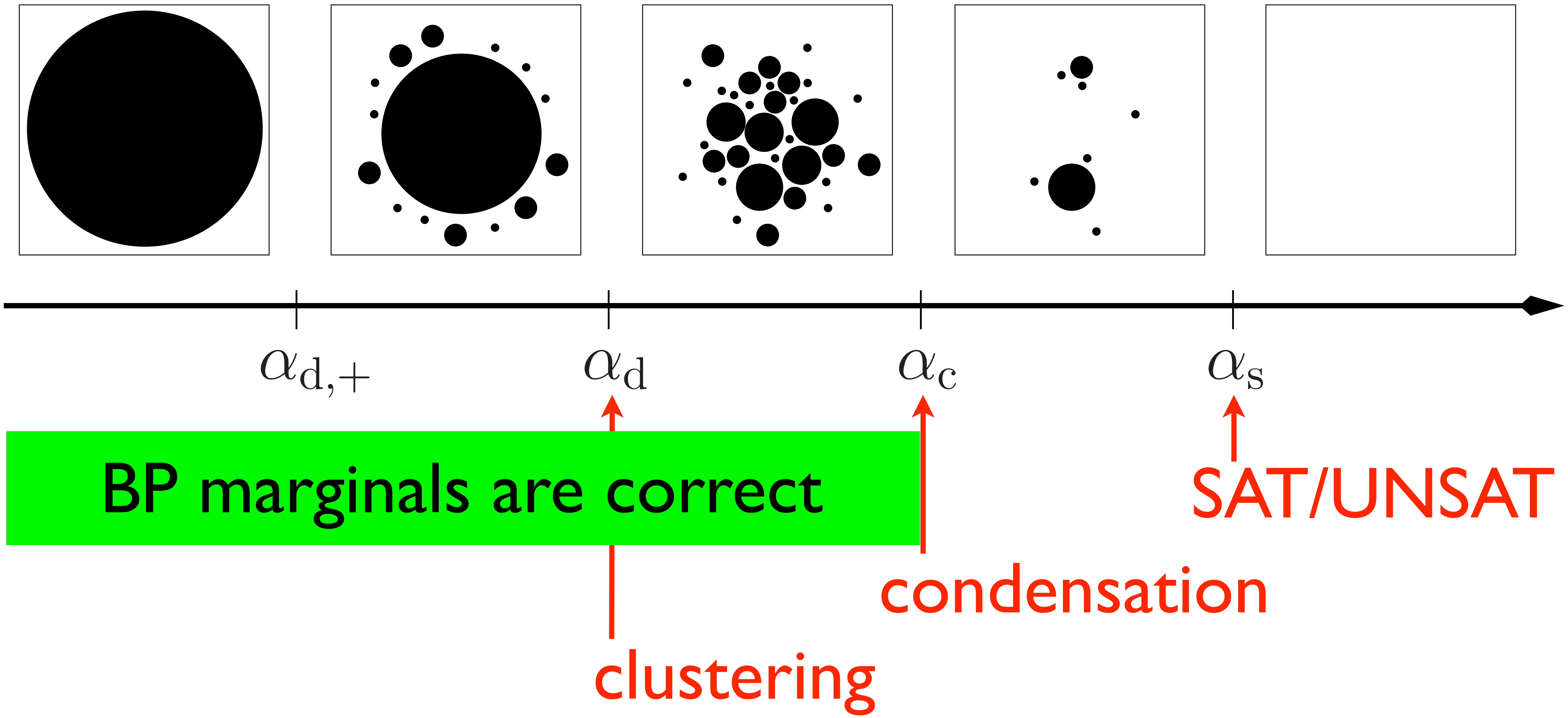
Standard picture



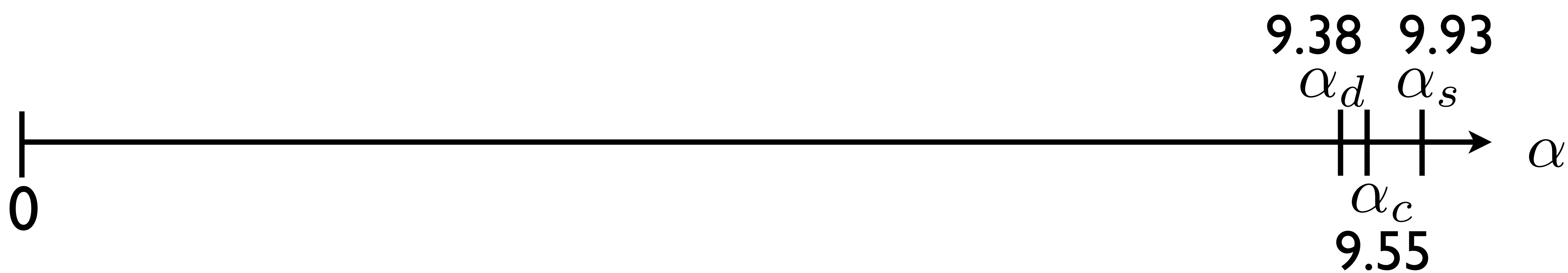
More phase transitions in random k-SAT (k > 3)



More phase transitions in random k-SAT (k > 3)

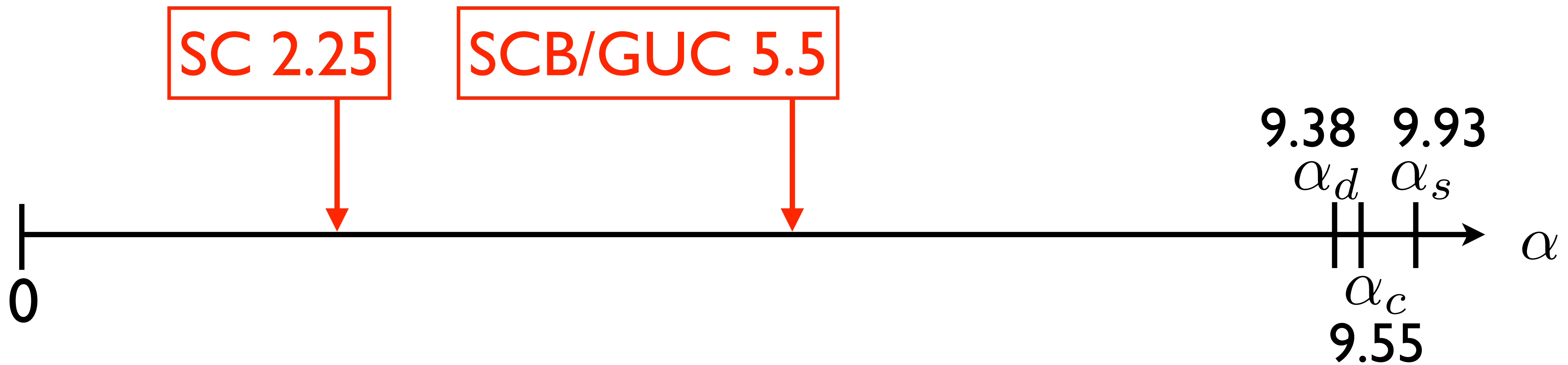


Performance of algorithms for random 4-SAT

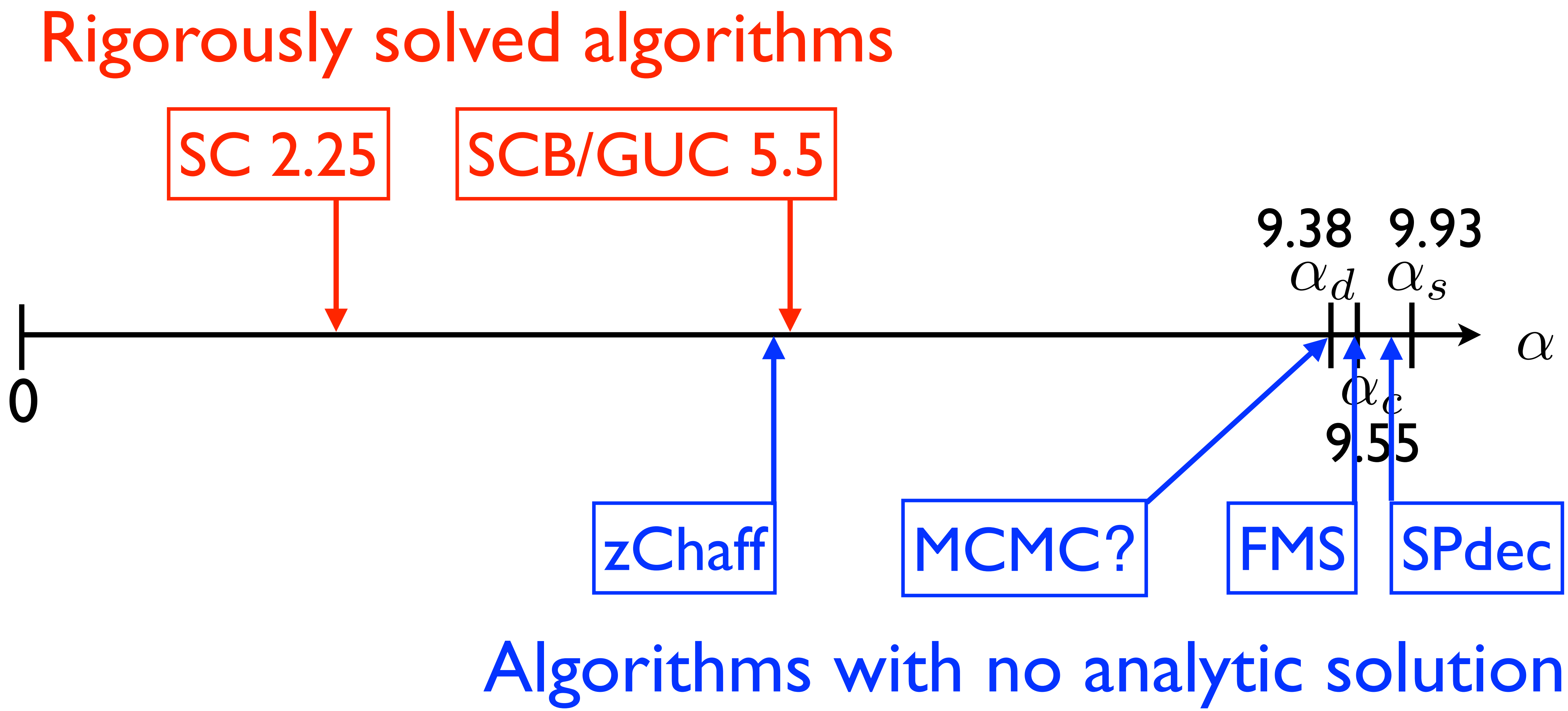


Performance of algorithms for random 4-SAT

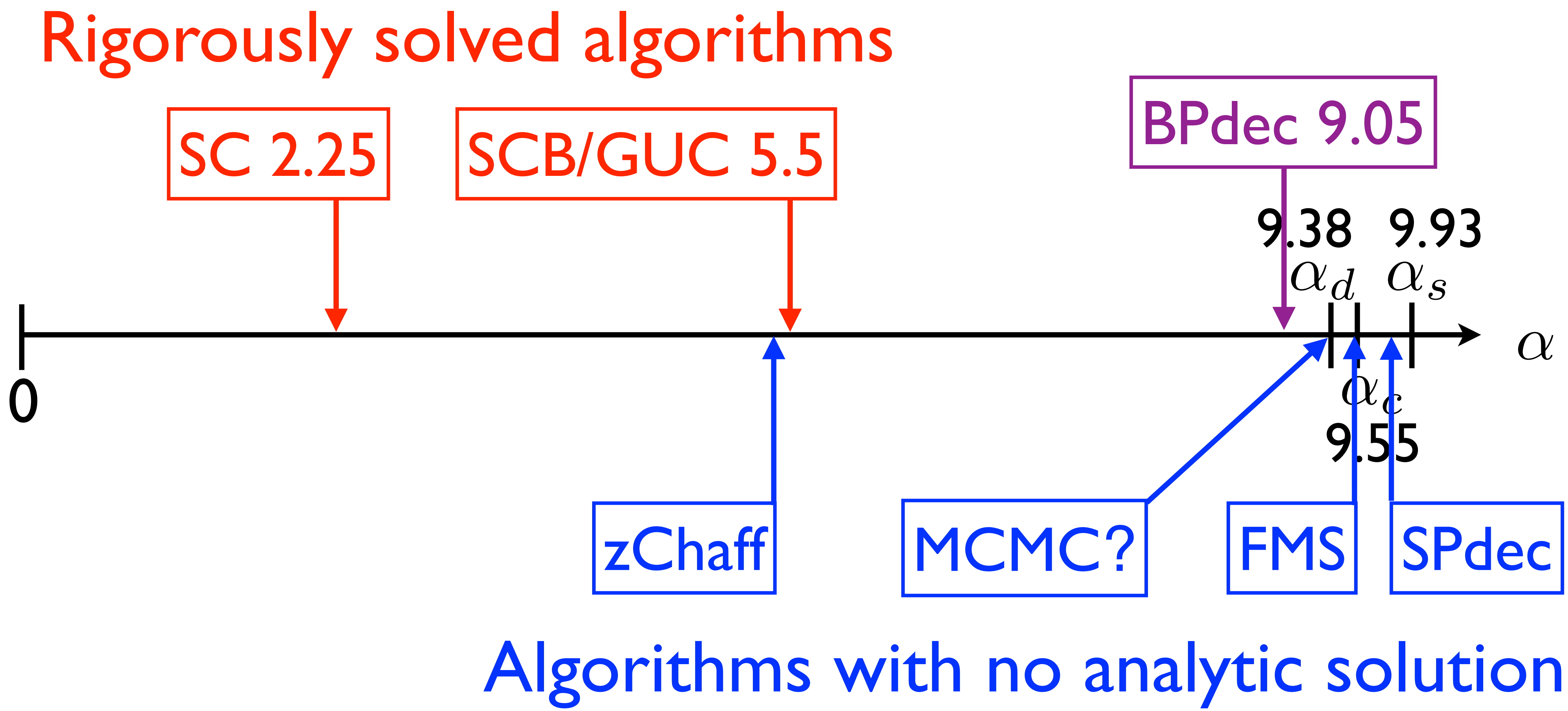
Rigorously solved algorithms



Performance of algorithms for random 4-SAT



Performance of algorithms for random 4-SAT



Two broad classes of solving algorithms

- **Local search**
  (biased) random walks in
  the space of configurations
  E.g. Monte Carlo, WalkSAT, FMS, ChainSAT, ...

- **Sequential construction**
  at each step a variable is assigned
  E.g. UCP, GUCP, BP/SP guided decimation

  - the order of assignment of variables

  - the information used to assign variables

The **oracle** guided algorithm
(a thought experiment)

- Start with all variables unassigned
- while (there are unassigned variables)
  - choose (randomly) an unassigned variable σ_i
  - ask the **oracle** the marginal of this variable $\mu_i(\,\cdot\,|\underline{\sigma}(t))$
  - assign σ_i according to its marginal

Samples solutions uniformly :-)

Oracle job is #P-complete in general :-(

Ensemble of θ-decimated CSP

1. Draw a CSP formula with parameter α
2. Draw a uniform solution $\underline{\tau}$ of this CSP
3. Choose a set D_θ by retaining each variable independently with probability θ
4. Consider the residual formula on the variables outside D_θ obtained by imposing the allowed configurations to coincide with $\underline{\tau}$ on D_θ

Not an ensemble of randomly uniform formulae conditioned on their degree distributions (step 2 depends on step 1)

Ensemble of θ-decimated CSP

- Residual entropy:

$$\omega(\theta) = \lim_{N\to\infty} \frac{1}{N} \mathbb{E}_F \mathbb{E}_{\underline{\tau}} \mathbb{E}_D [\ln Z(\underline{\tau}_D)]$$

$Z(\underline{\tau}_D)$ = number of solutions compatible with the solution "exposed" on D_θ

- Fraction of frozen variables:

$$\phi(\theta) = \frac{1}{N} \mathbb{E}_F \, \mathbb{E}_{\underline{\tau}} \, \mathbb{E}_{D_\theta} |W_\theta|$$

$$W_\theta = D_\theta \cup \{\text{variables implied by } D_\theta\}$$

Ensemble of θ-decimated CSP

- Compute $Z(\underline{\tau}_D)$ by the Bethe-Peierls approx.

$$\ln Z(\underline{\tau}_D) = - \sum_{i \notin D, a \in \partial i} \ln \left(\sum_{\sigma_i} \nu^{\underline{\tau}_D}_{a \to i}(\sigma_i) \eta^{\underline{\tau}_D}_{i \to a}(\sigma_i) \right) + \sum_a \ln \left(\sum_{\underline{\sigma}_{\partial a}} \psi_a(\underline{\sigma}_{\partial a}) \prod_{i \in \partial a} \eta^{\underline{\tau}_D}_{i \to a}(\sigma_i) \right)$$
$$+ \sum_{i \notin D} \ln \left(\sum_{\sigma_i} \prod_{a \in \partial i} \nu^{\underline{\tau}_D}_{a \to i}(\sigma_i) \right),$$

where messages satisfy standard BP equations
with the boundary condition

$$\eta^{\underline{\tau}_D}_{i \to a}(\sigma_i) = \delta_{\sigma_i, \tau_i} \text{ when } i \in D$$

Practical approximate implementation of the thought experiment (BP guided decimation algorithm)

a. Choose a random order of the variables $i(1),\ldots,i(N)$

b. for $t=1,\ldots,N$

   1. find a fixed point of BP eqns. with boundary condition $\eta^{\underline{\tau}_D}_{i\to a}(\sigma_i) = \delta_{\sigma_i,\tau_i}$

   2. draw $\sigma_{i(t)}$ according to the BP estimation of $\mu(\sigma_i|\underline{\tau}_{D_{t-1}})$

   3. set $\tau_{i(t)} = \sigma_{i(t)}$

When BP guided decimation is expected to work

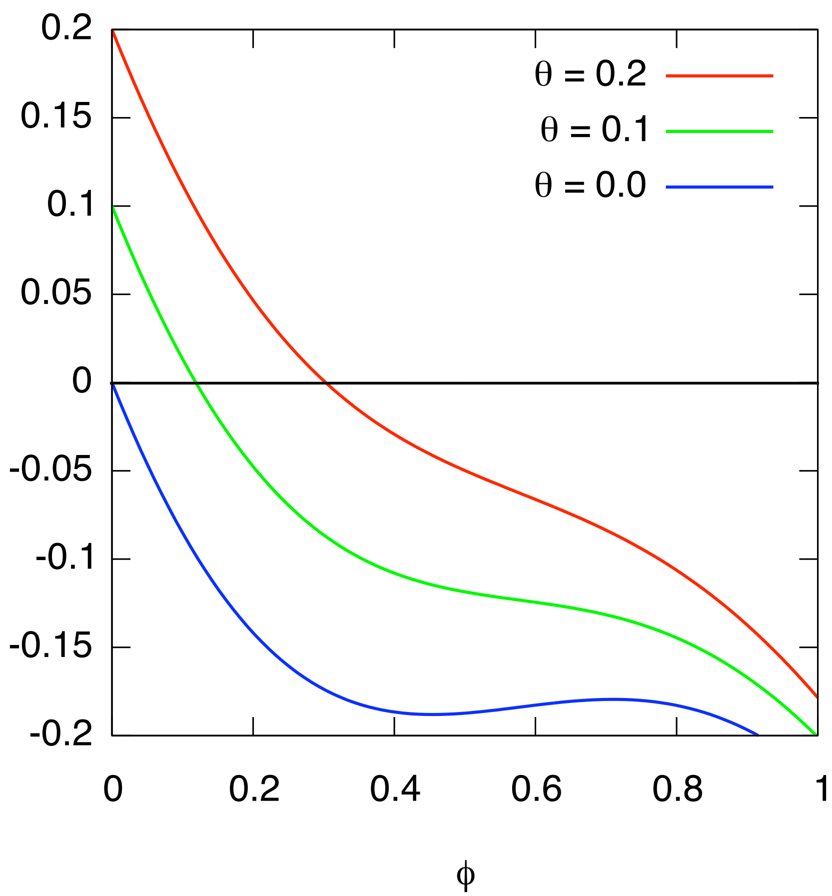
- At least 1 solution must exists ($\alpha < \alpha_s$)
- No contradictions should be generated
- Check for contradictions at each time
  - add step 0. where UCP/WP is run
- Can not go beyond condensation transition as BP marginals are no longer correct ($\alpha < \alpha_c$)

Results for random 3-XORSAT

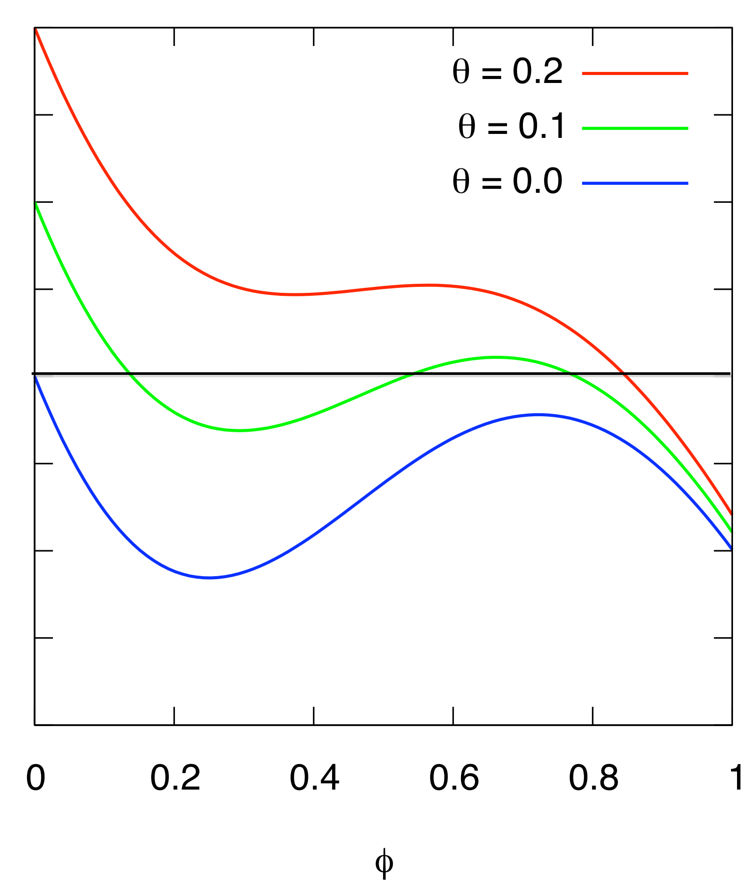
- Full analytic solution (by differential equations)

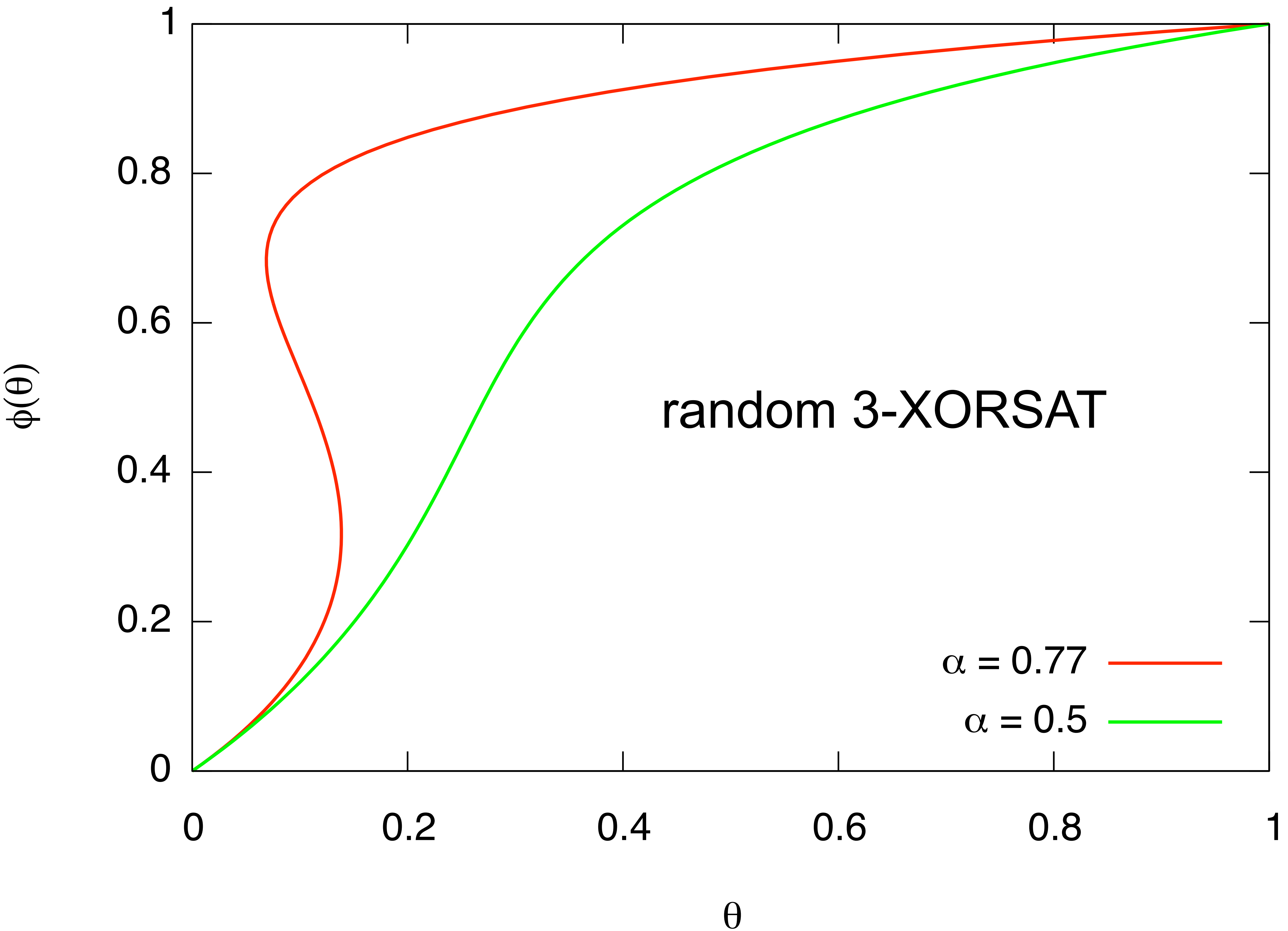
$$\phi = \theta + (1-\theta)\left(1 - e^{-\alpha k \phi^{k-1}}\right)$$

$\alpha < \alpha_a$

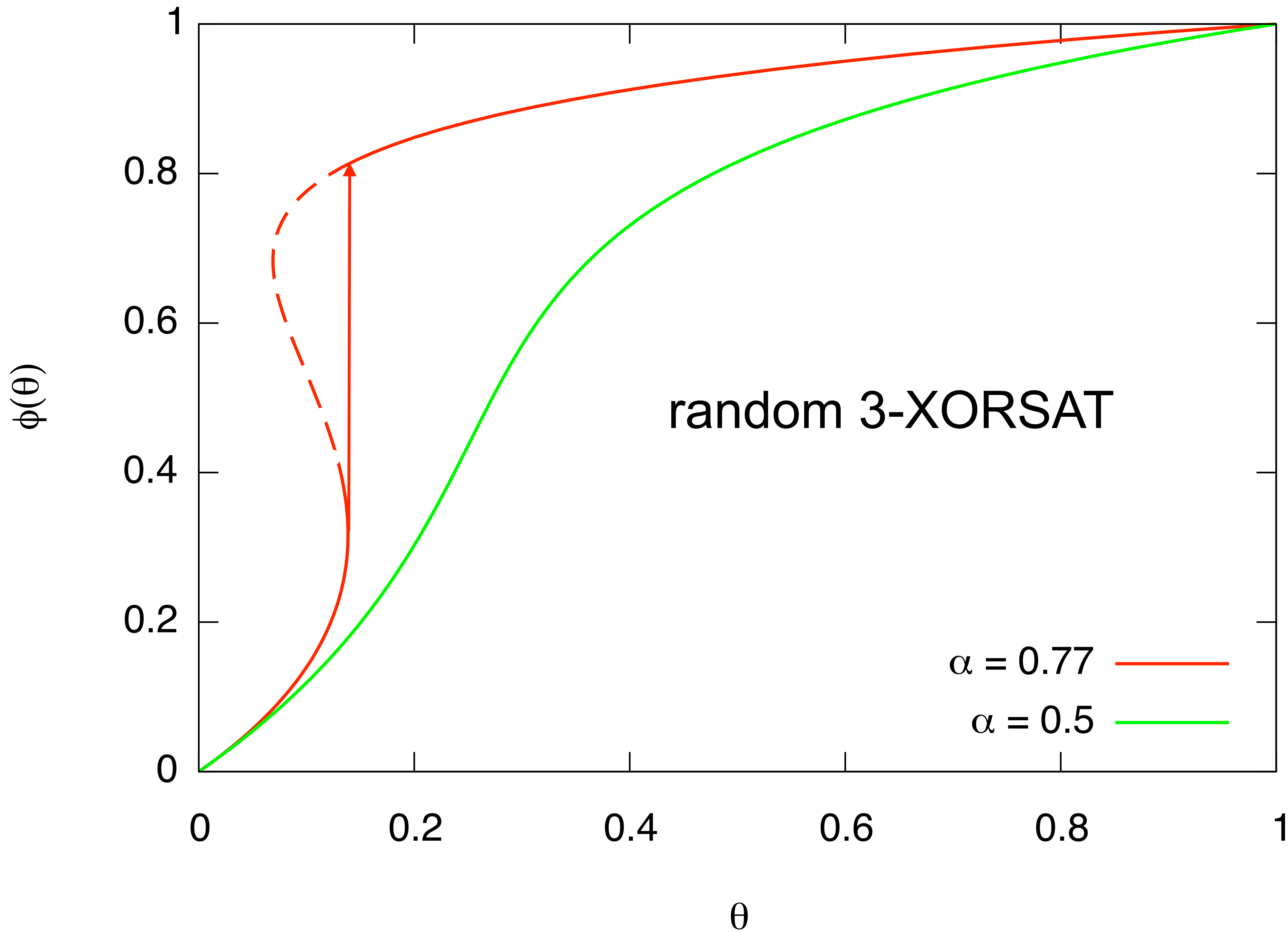


$\alpha > \alpha_a$





1
0.8
0.6
0.4
0.2
0
φ(θ)
random 3-XORSAT
α = 0.77
α = 0.5
0
0.2
0.4
0.6
0.8
1
θ



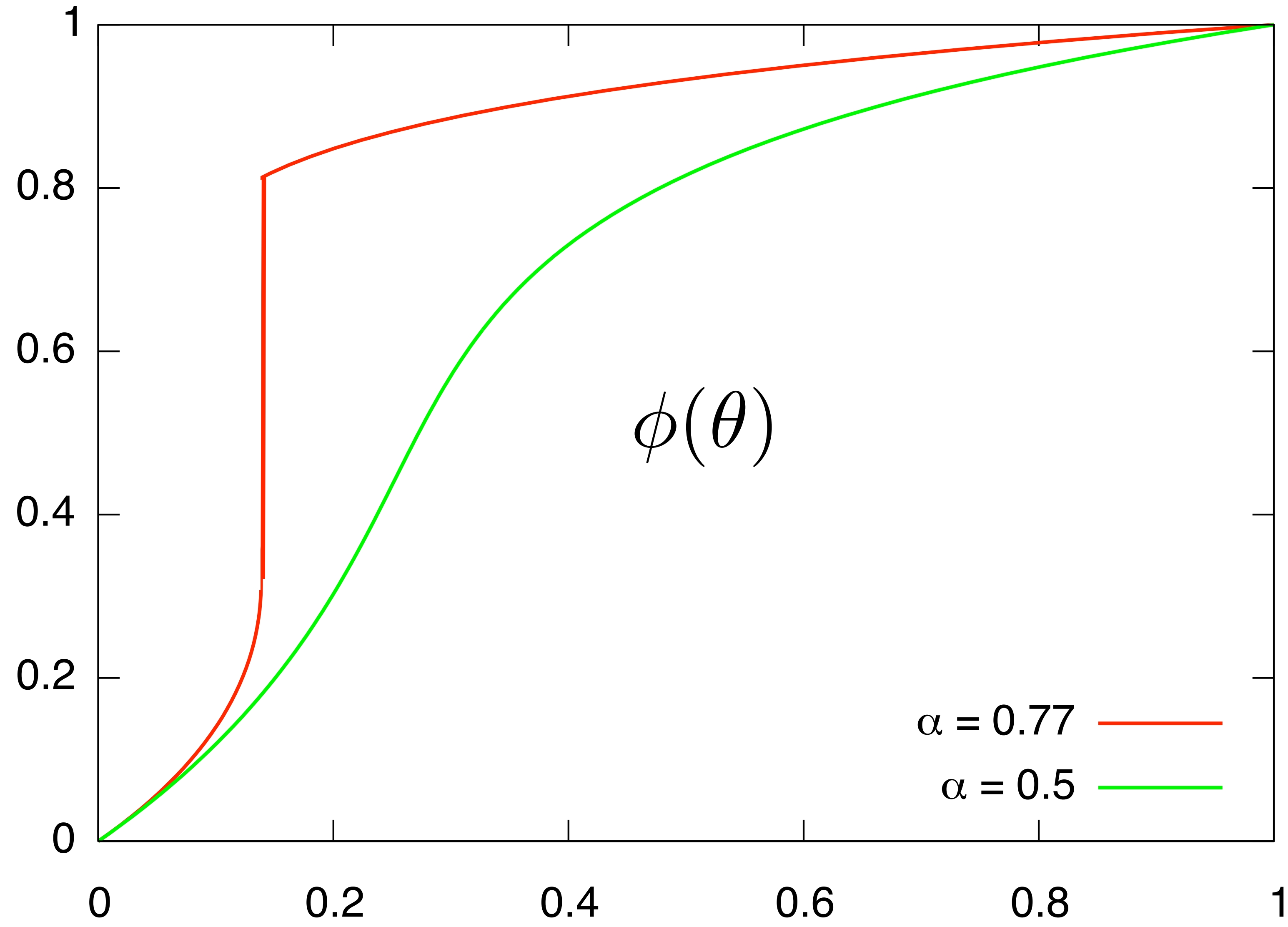
random 3-XORSAT
α = 0.77
α = 0.5
φ(θ)
θ
0
0.2
0.4
0.6
0.8
1

Results for random 3-XORSAT

Phase transition for $\alpha > \alpha_a = \frac{1}{k}\left(\frac{k-1}{k-2}\right)^{k-2}$ like UCP

Jump in $\phi(\theta)$
and
cusp in $\omega(\theta)$

$$\alpha_a(k=3) = \frac{2}{3}$$

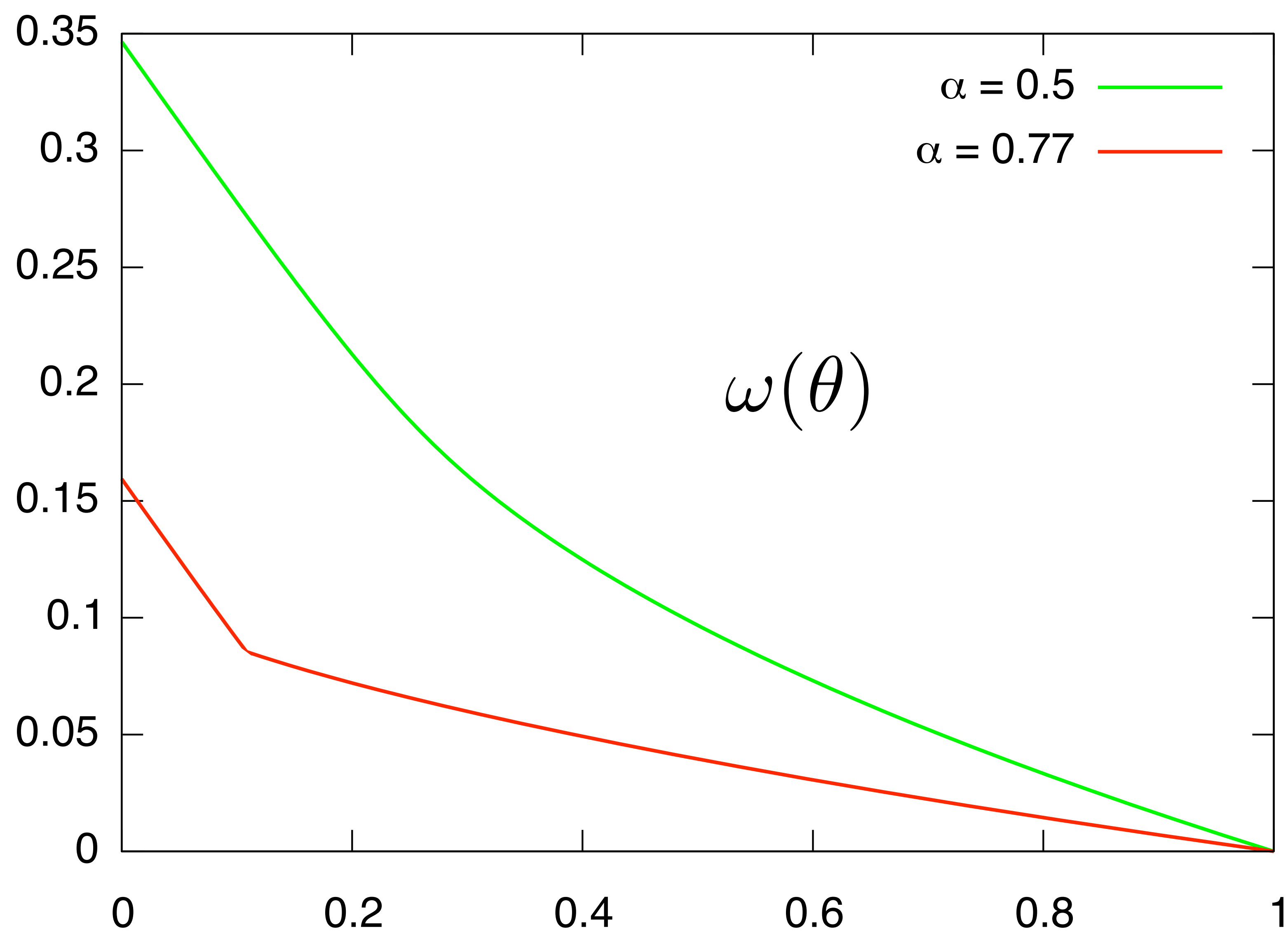


Results for random 3-XORSAT

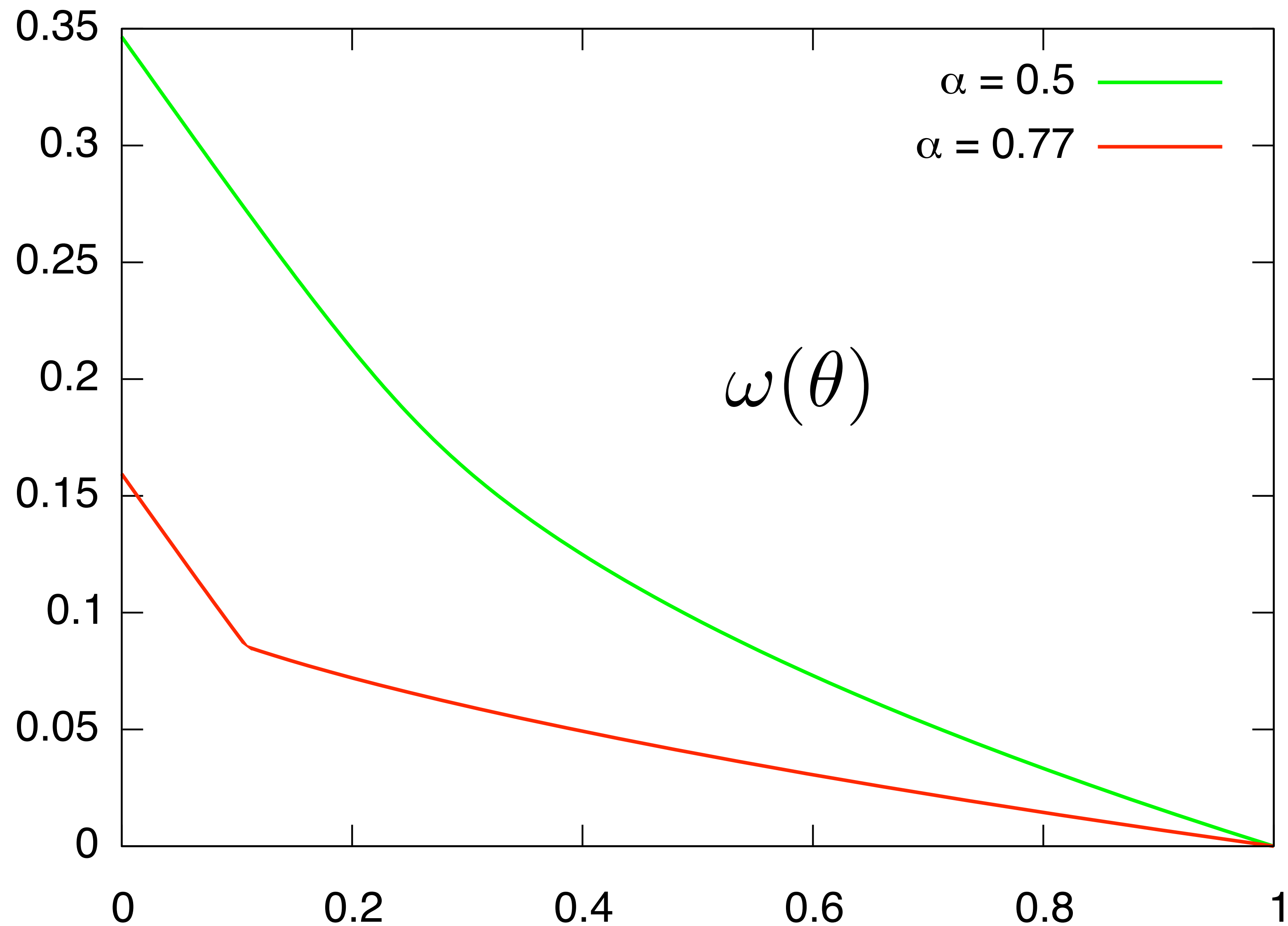
Phase transition for $\alpha > \alpha_a = \frac{1}{k}\left(\frac{k-1}{k-2}\right)^{k-2}$ like UCP

Jump in $\phi(\theta)$
and
cusp in $\omega(\theta)$

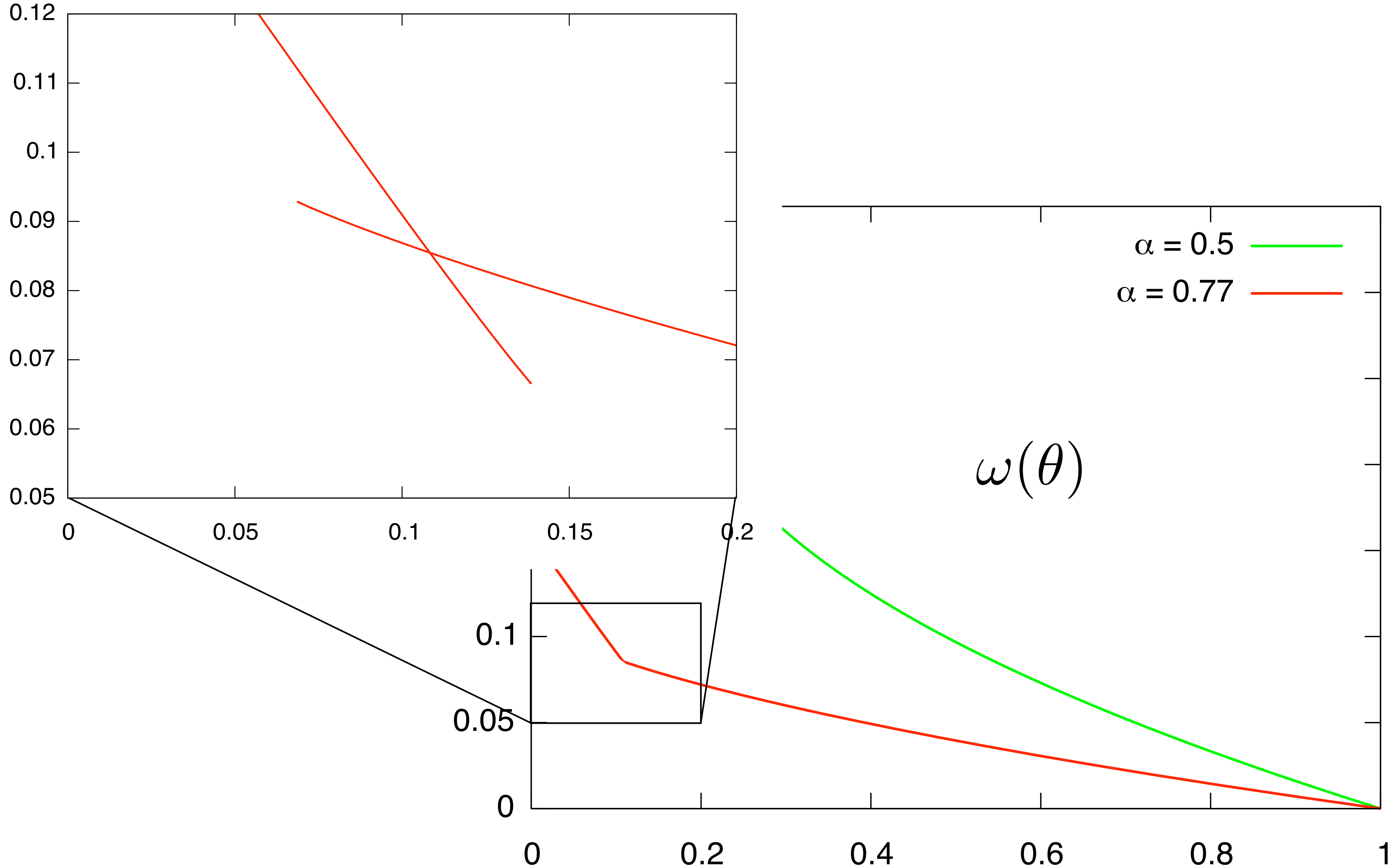
$$\alpha_a(k=3) = \frac{2}{3}$$



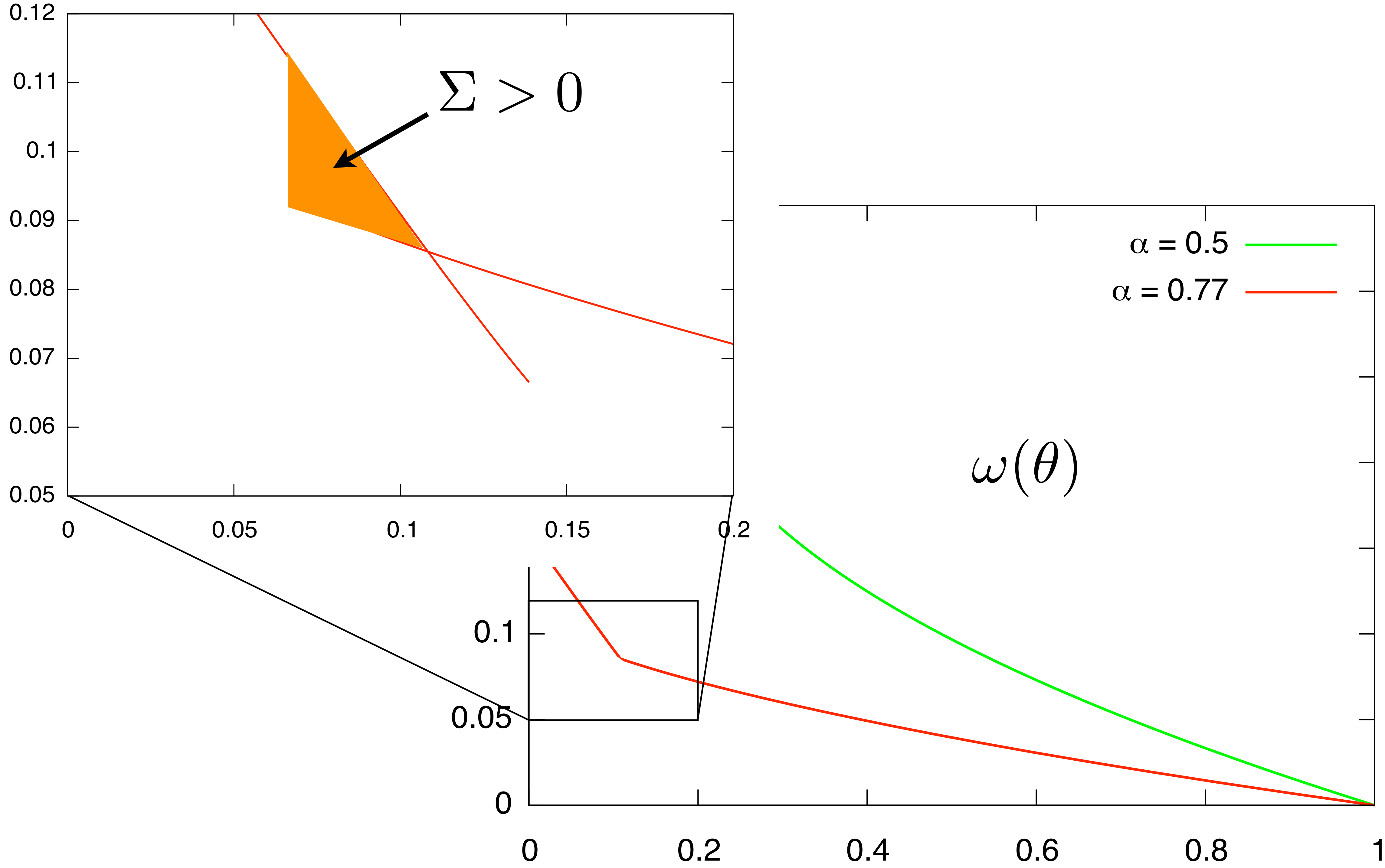
Results for random 3-XORSAT



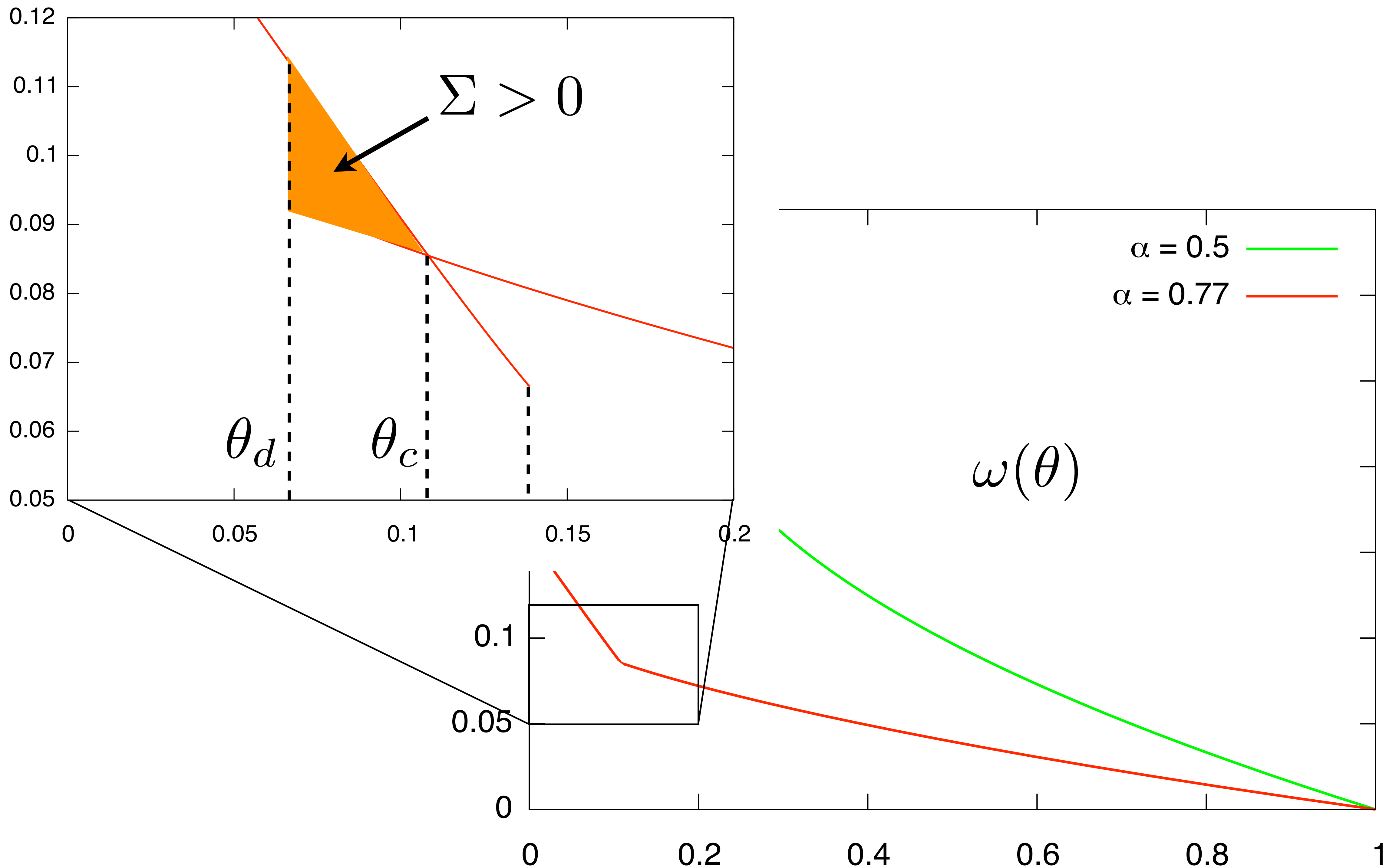
Results for random 3-XORSAT



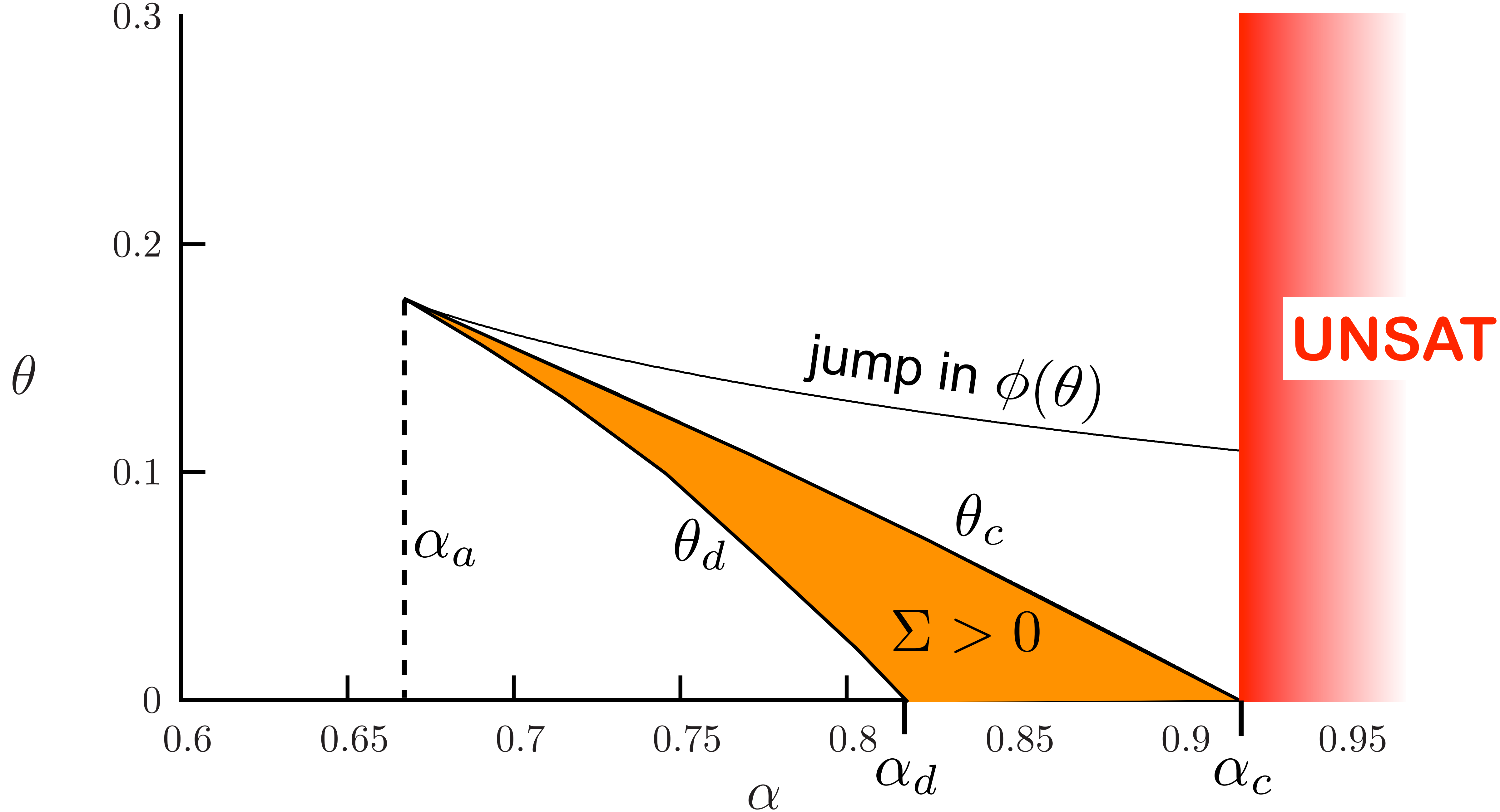
Results for random 3-XORSAT



Results for random 3-XORSAT



Phase diagram for random 3-XORSAT

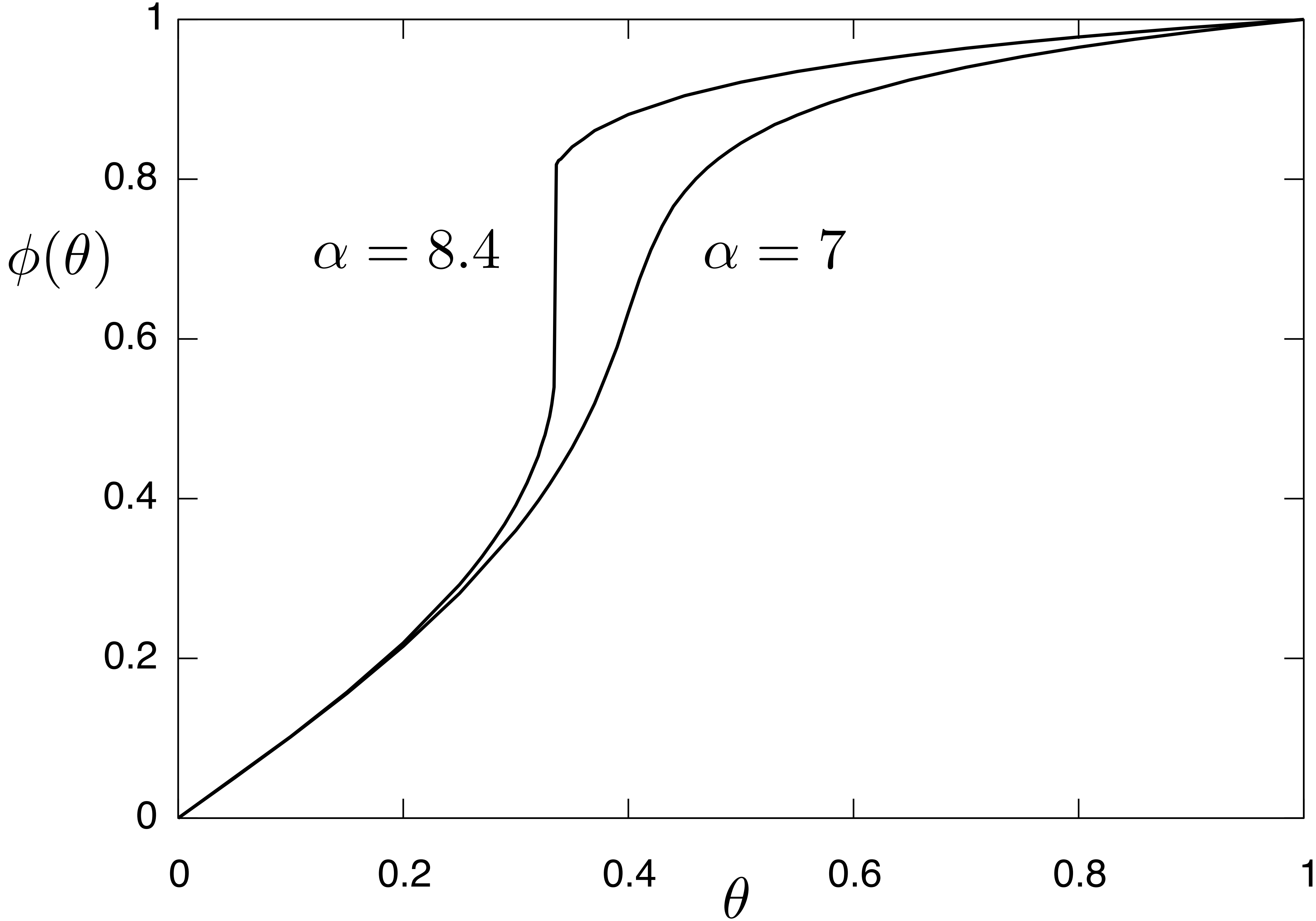


Numerics for random k-SAT

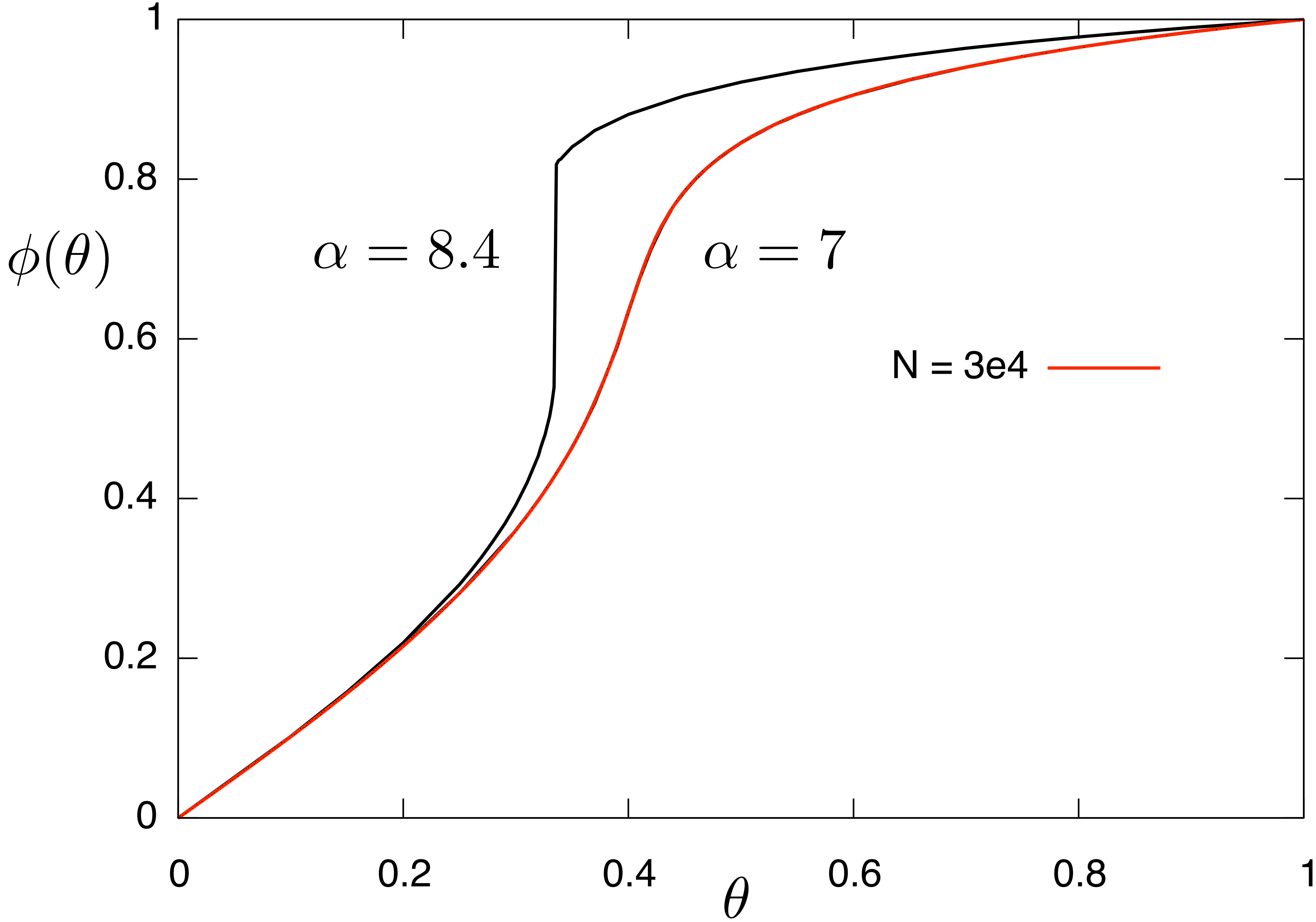
$$\alpha_d = 9.38$$
$$\alpha_c = 9.55$$
$$\alpha_s = 9.93$$

- k = 4,   N = 1e3, 3e3, 1e4, 3e4
- Run WP
  - integer variables, no approximation
- Run BP
  - much care for dealing with quasi-frozen variables
  - slow convergence (damping and restarting trick)
  - maximum number of iterations (1000)   Much larger than the diameter (~2)

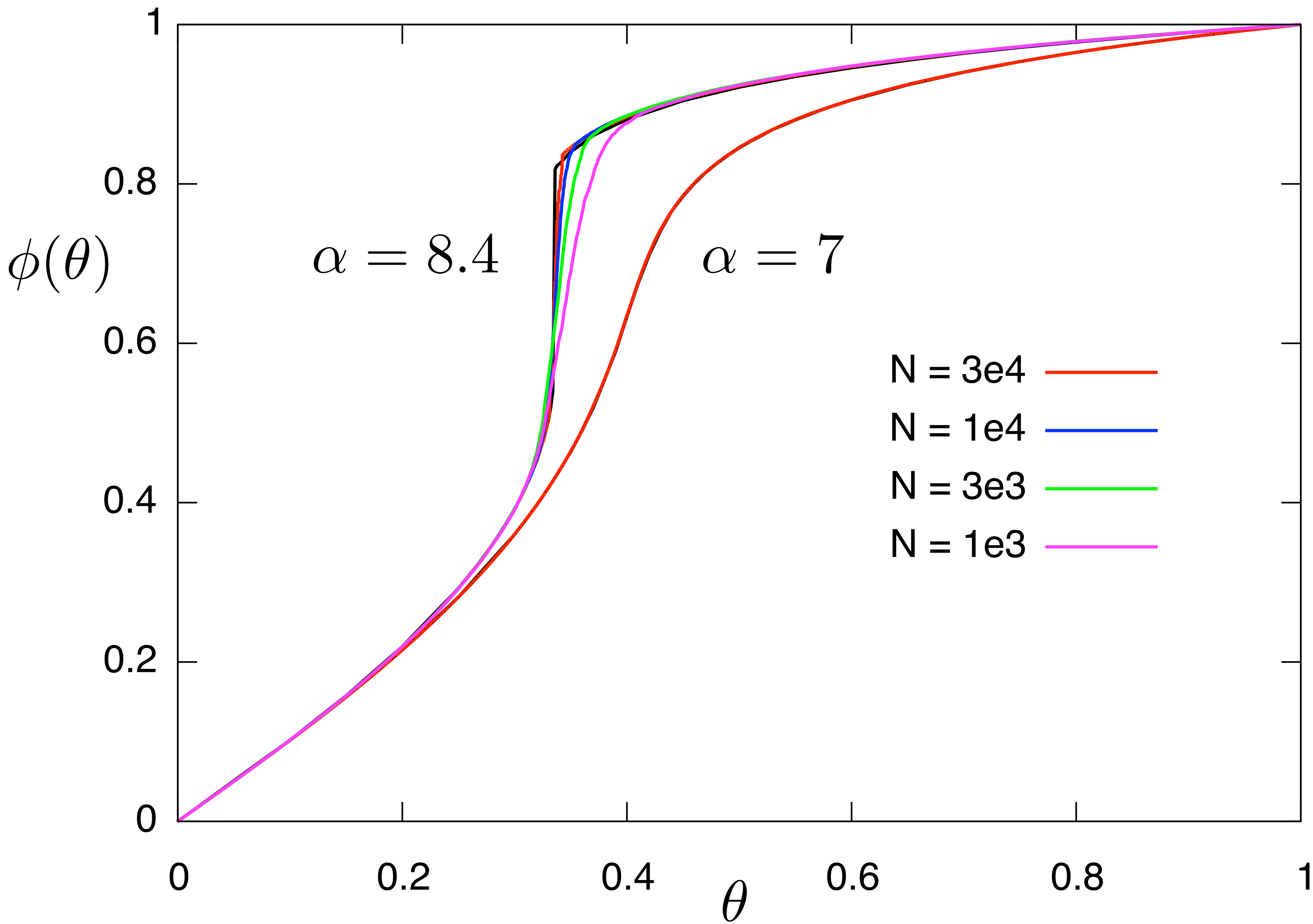
Results for random 4-SAT



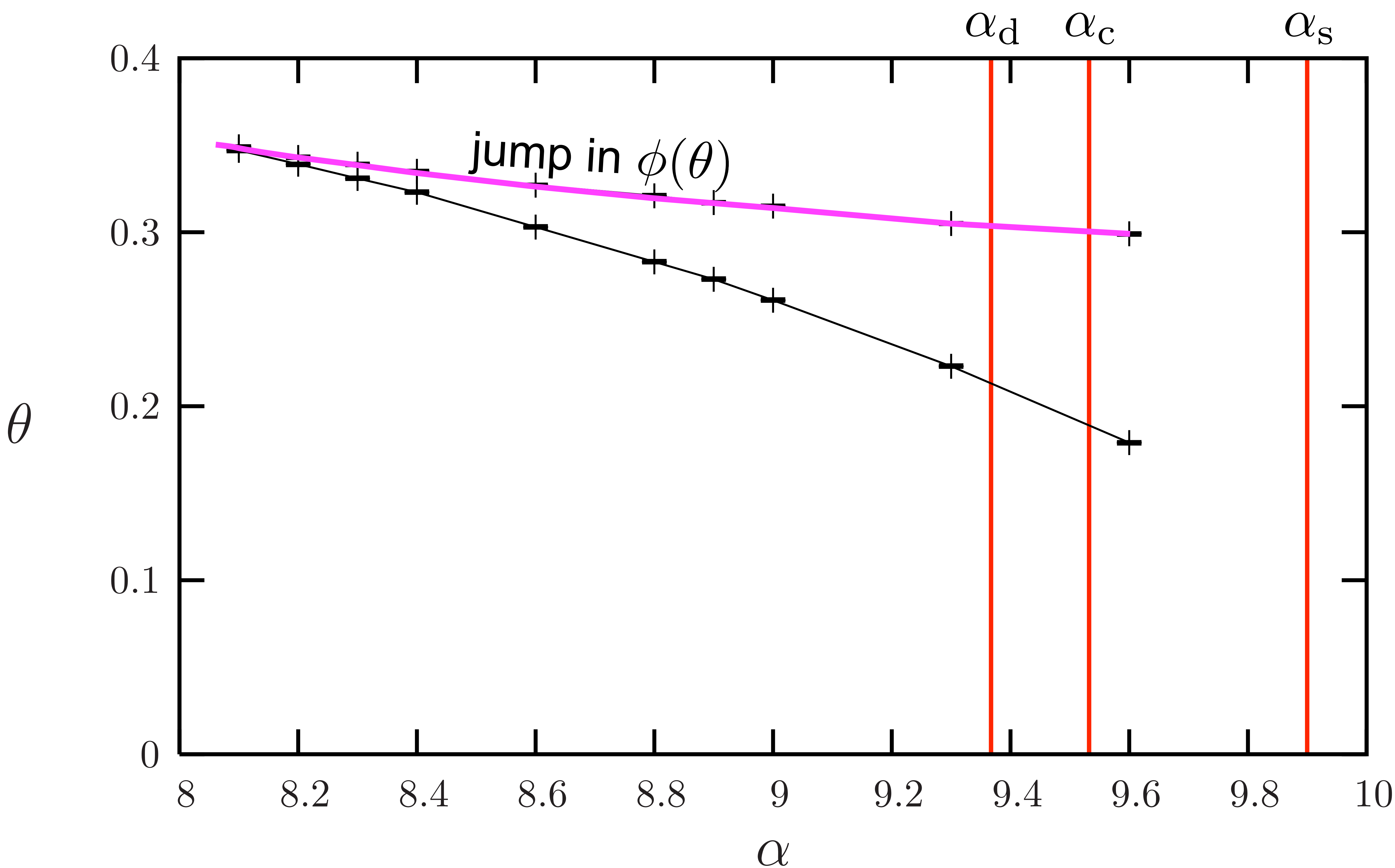
Results for random 4-SAT



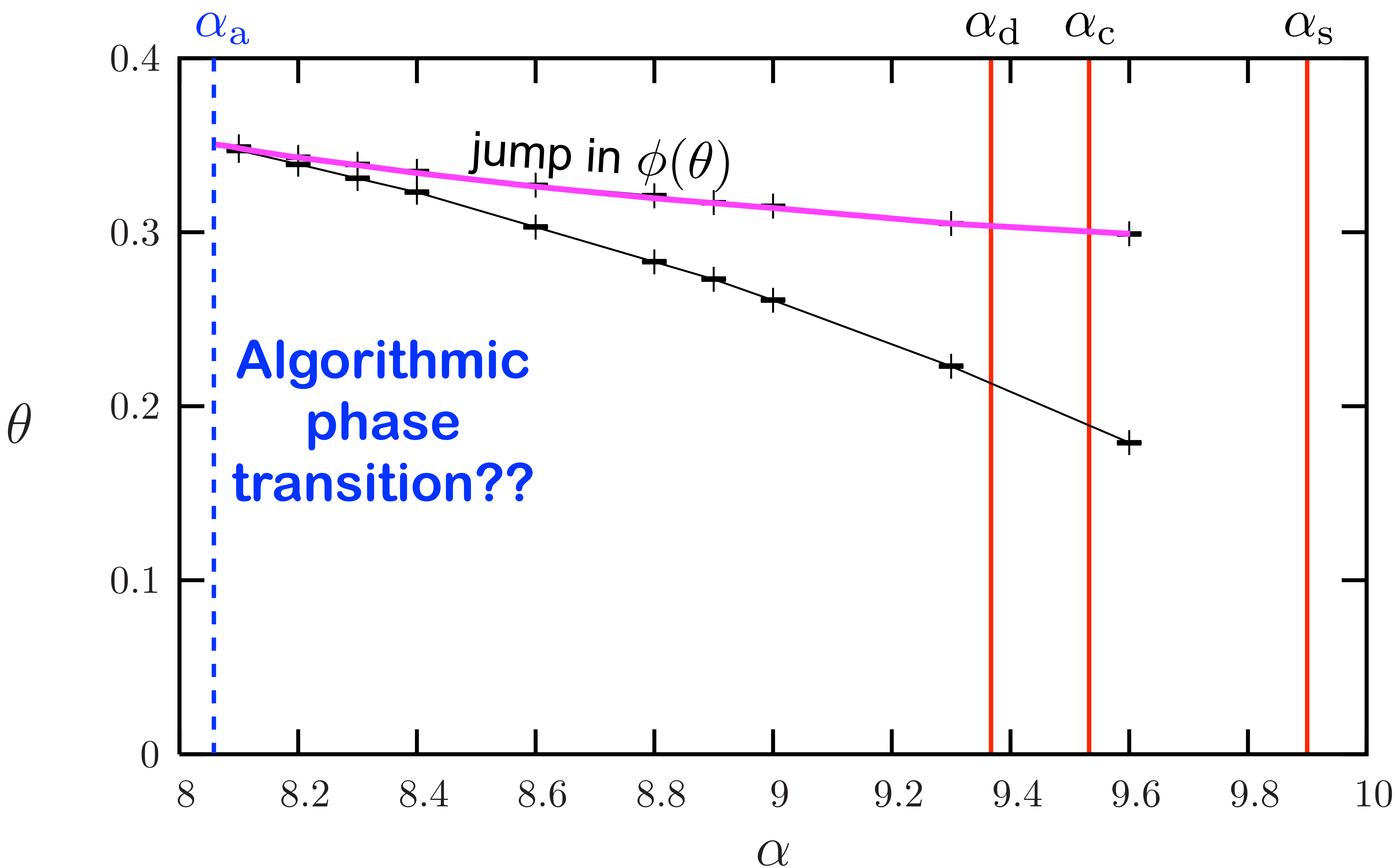
Results for random 4-SAT



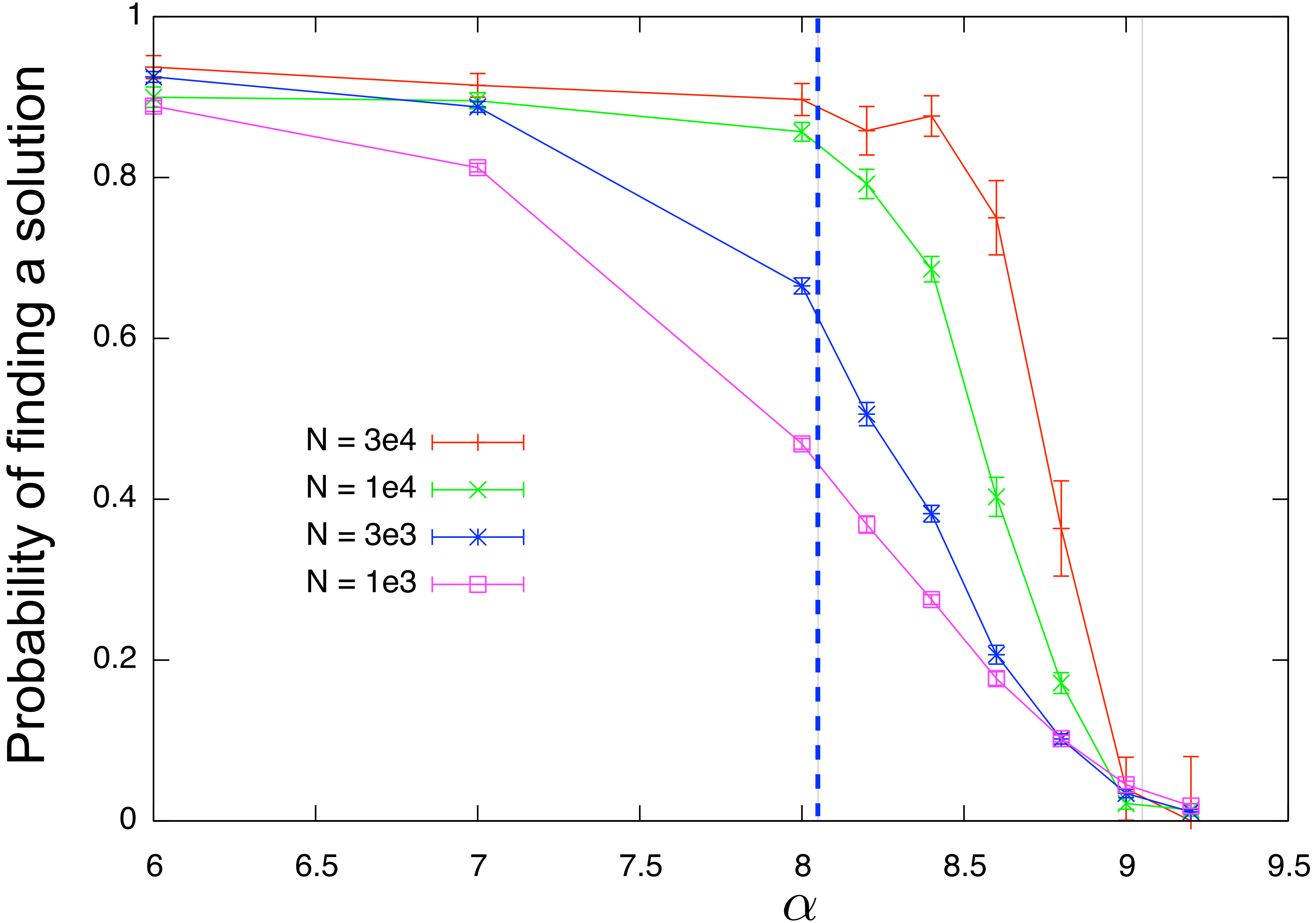
Results for random 4-SAT



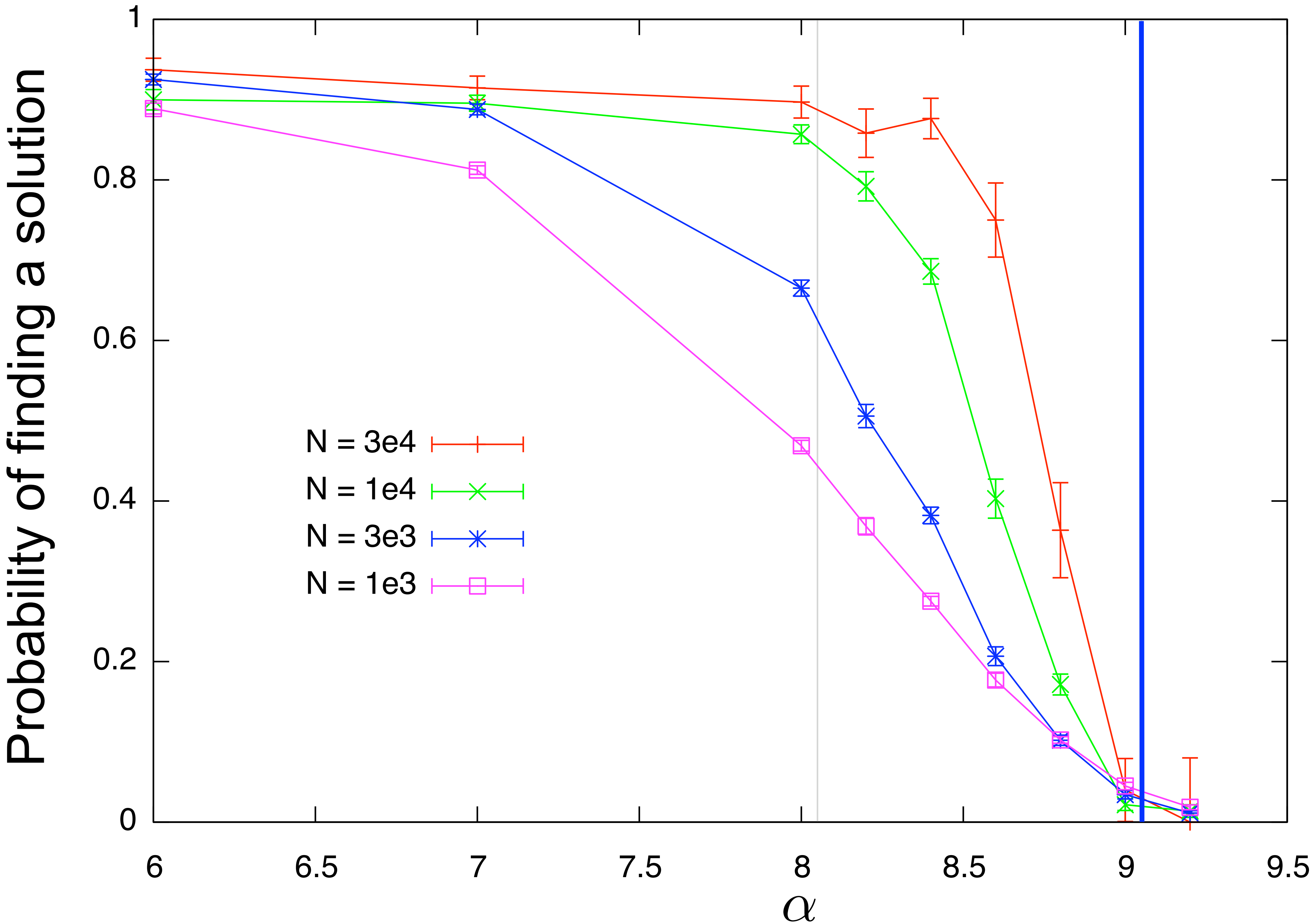
Results for random 4-SAT



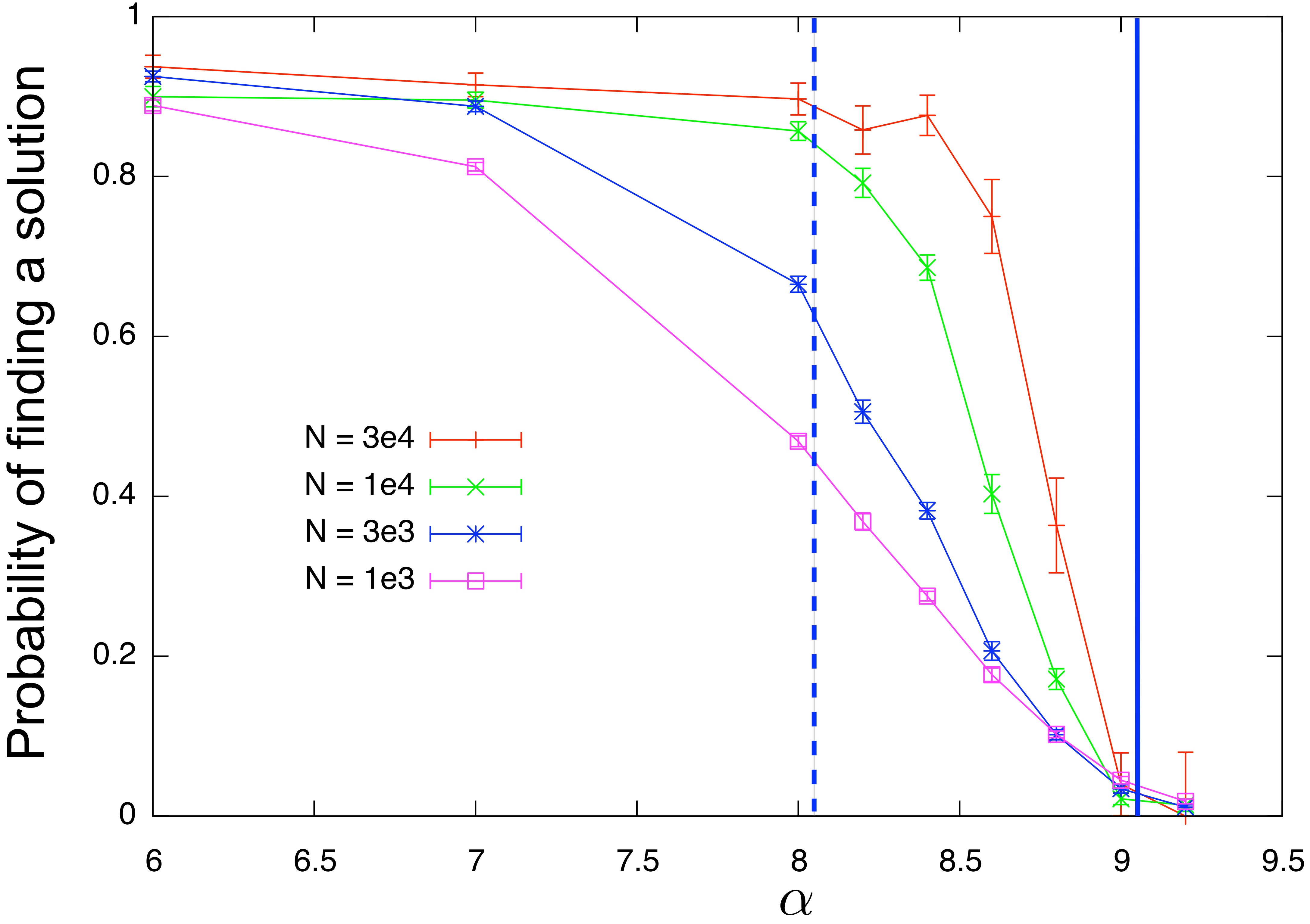
Results for random 4-SAT



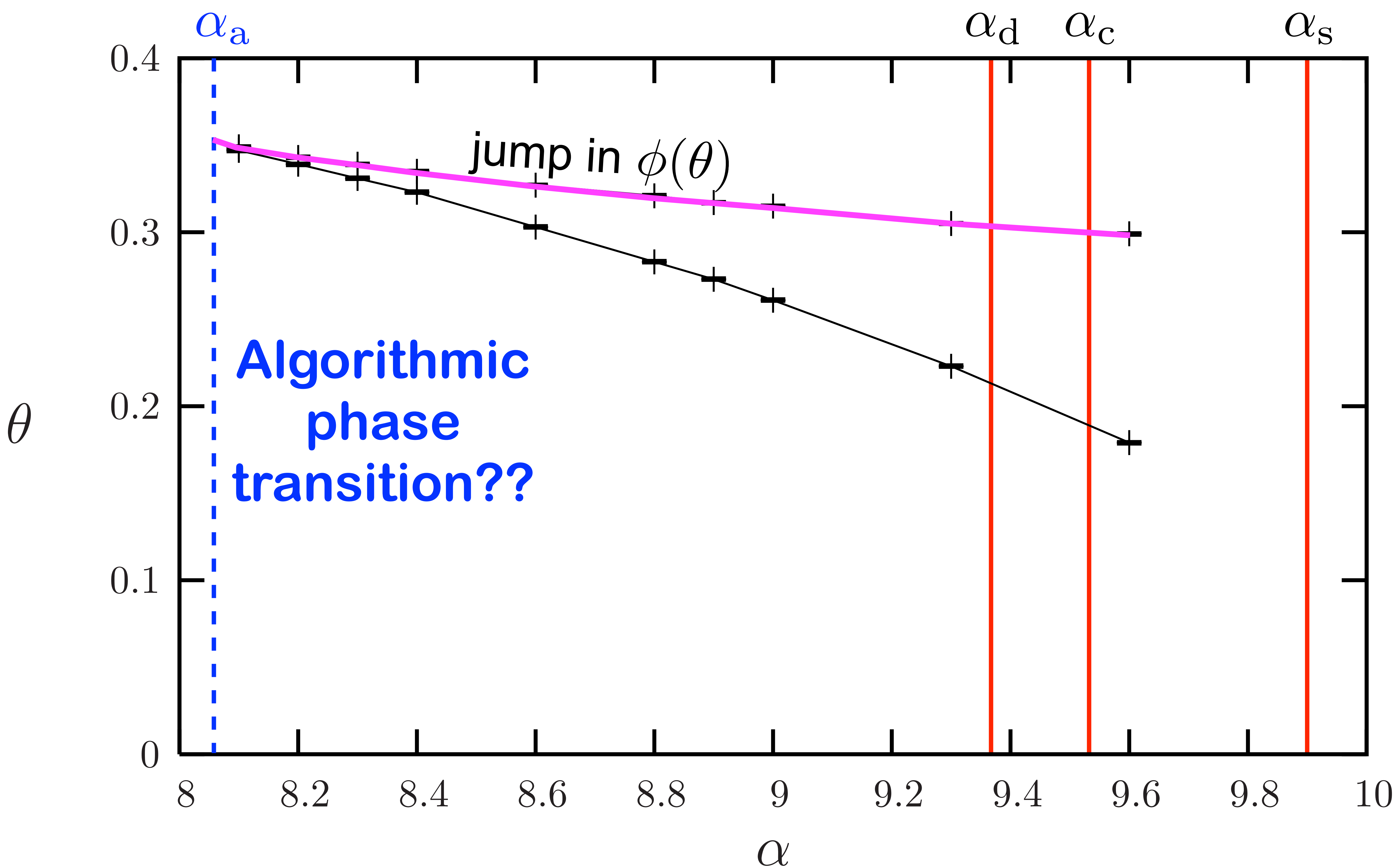
Results for random 4-SAT



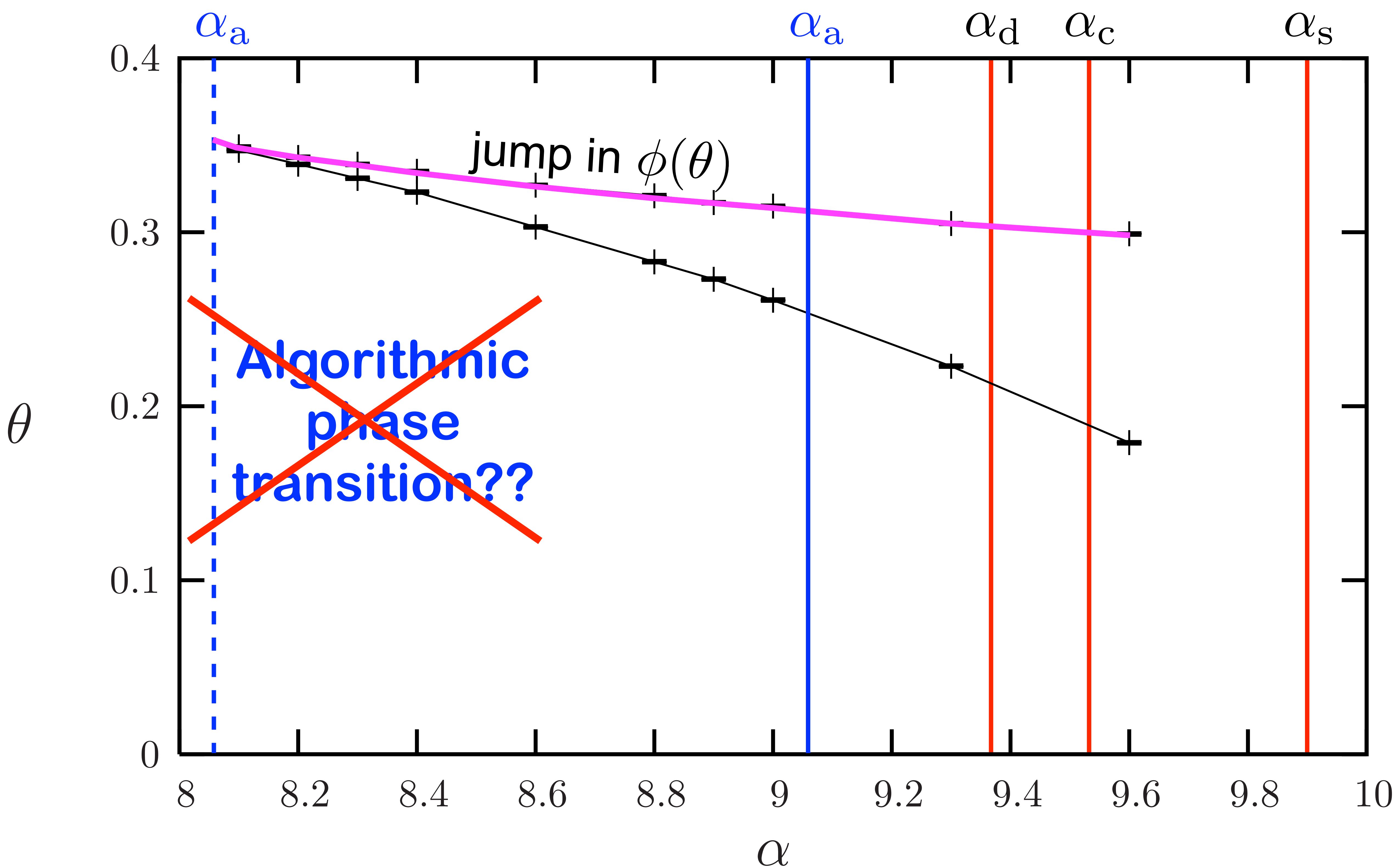
Results for random 4-SAT



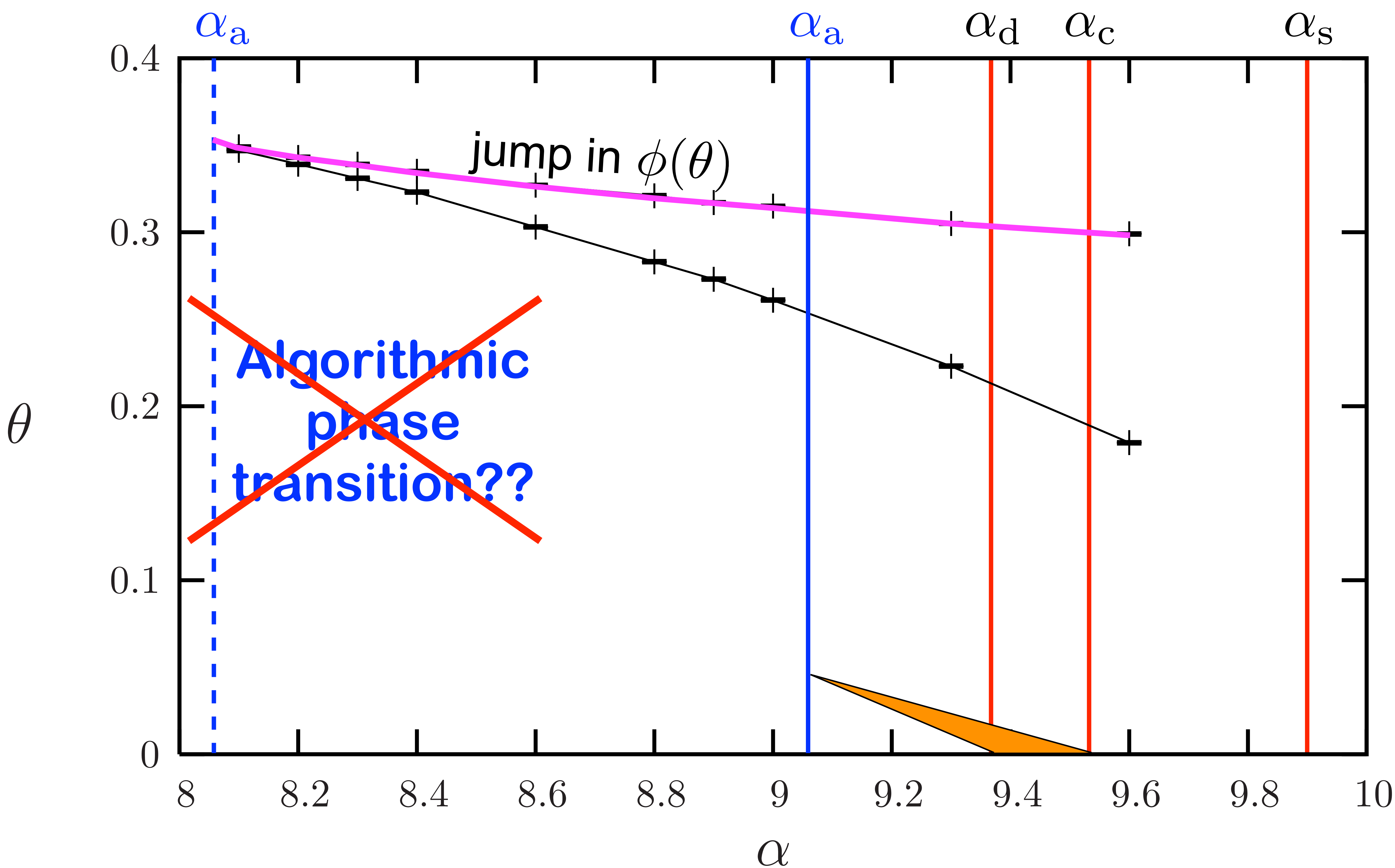
Results for random 4-SAT



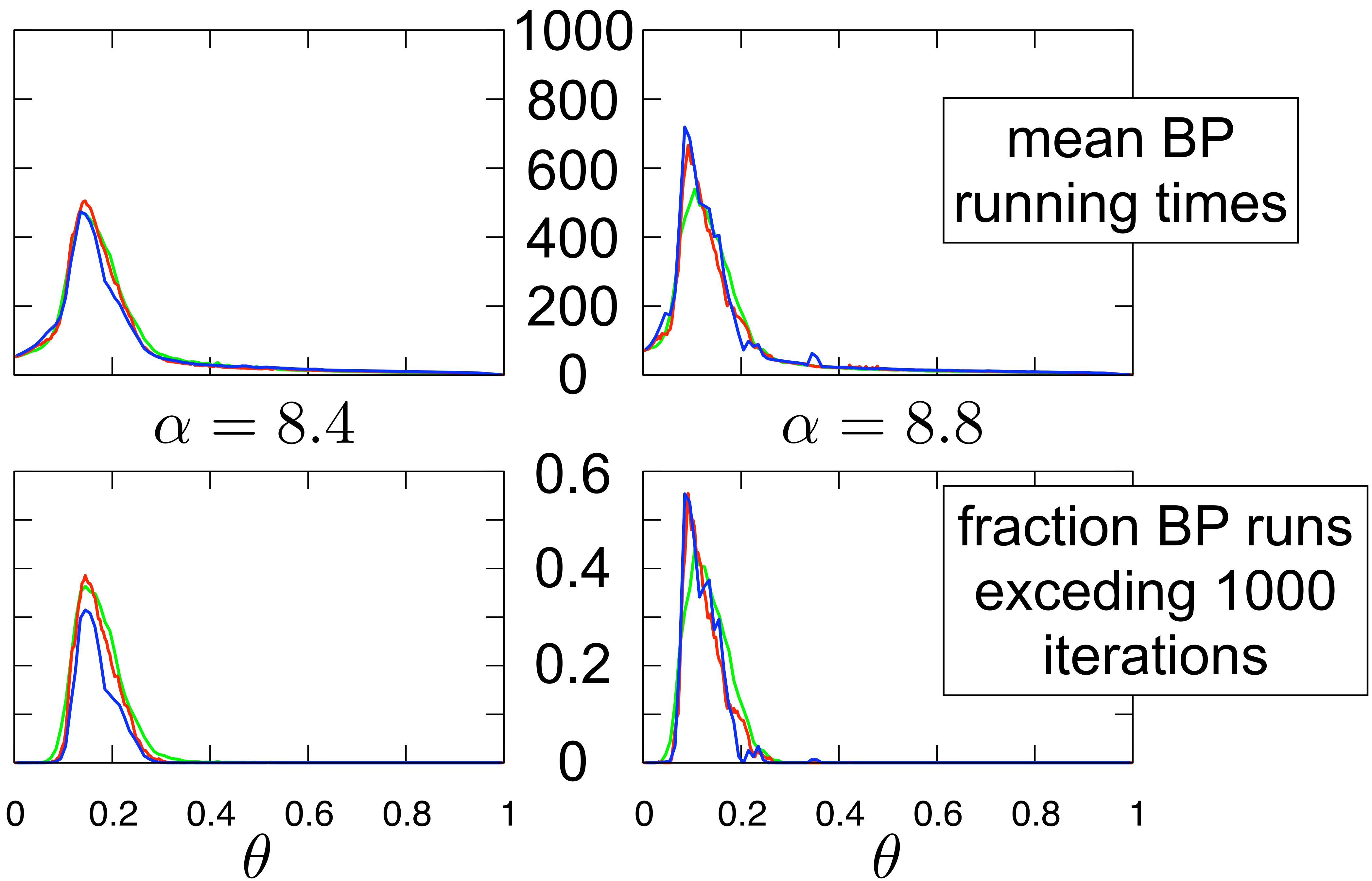
Results for random 4-SAT



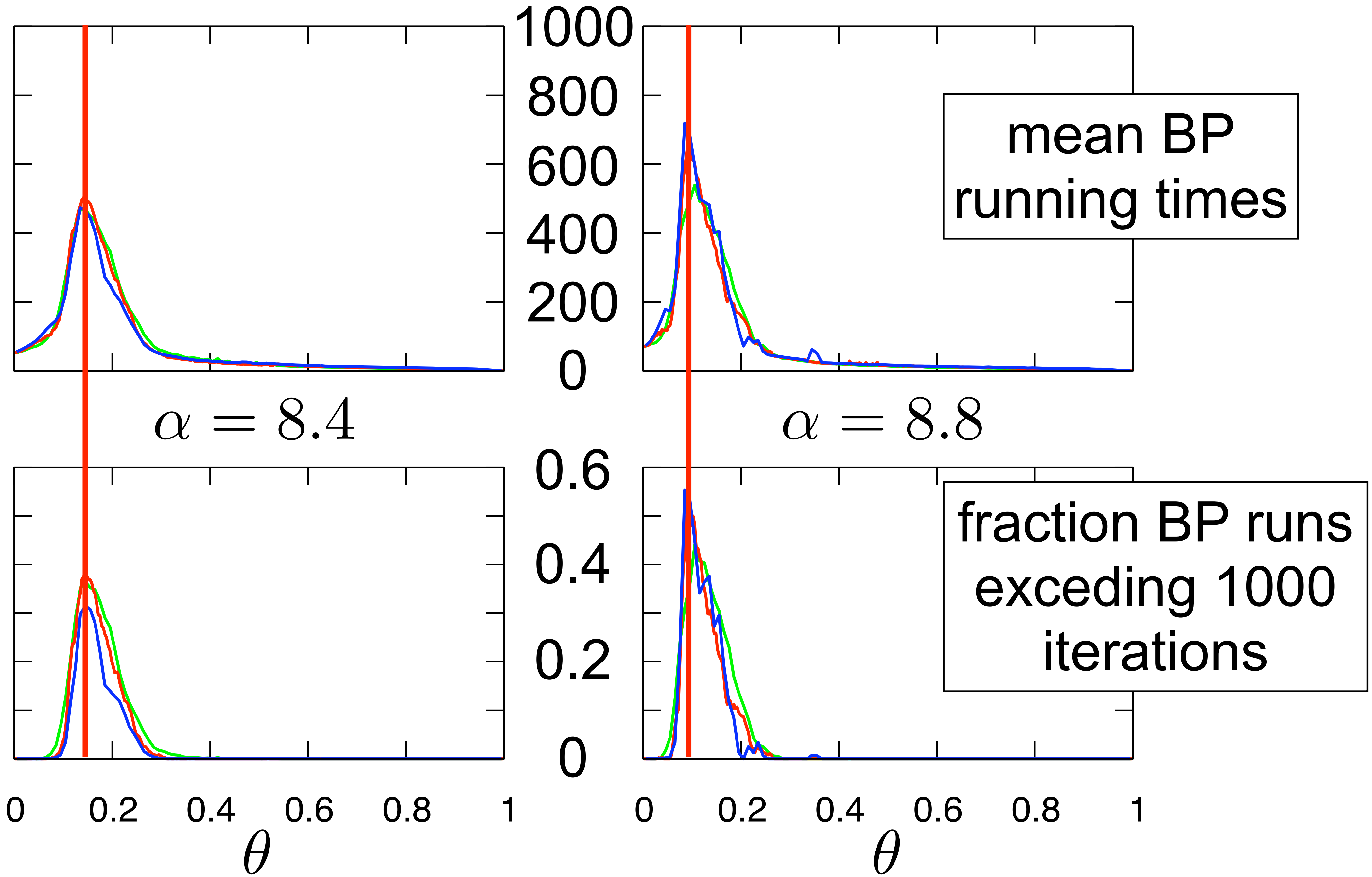
Results for random 4-SAT



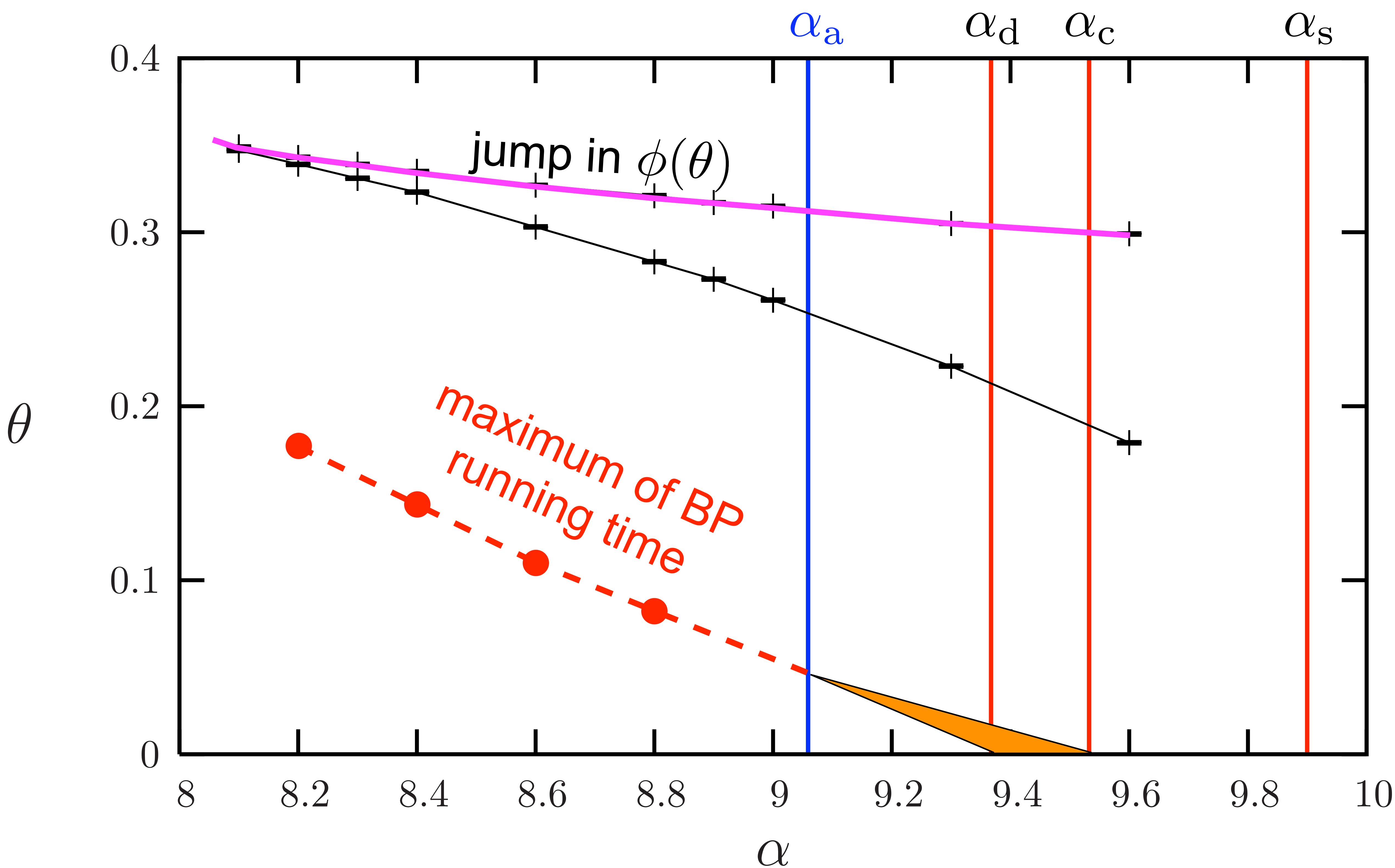
Results for random 4-SAT



Results for random 4-SAT



Results for random 4-SAT



Large k limit

$$\alpha_d \simeq \frac{\ln k}{k} 2^k \qquad \alpha_c \simeq \alpha_s \simeq 2^k$$

- Previous solvable algorithms

Pure Literal ("PL")	$o(1)$ as $k \to \infty$
Walksat, rigorous Walksat, non-rigorous	$\frac{1}{6} \cdot 2^k/k^2$ $2^k/k$
Unit Clause ("UC")	$\frac{1}{2}\left(\frac{k-1}{k-2}\right)^{k-2} \cdot \frac{2^k}{k}$
Shortest Clause ("SC")	$\frac{1}{8}\left(\frac{k-1}{k-3}\right)^{k-3} \frac{k-1}{k-2} \cdot \frac{2^k}{k}$
SC+backtracking ("SCB")	$\sim 1.817 \cdot \frac{2^k}{k}$

- Our prediction for BP guided decimation $\alpha_a \simeq \frac{e}{k} 2^k$

Large k limit

$$\alpha_d \simeq \frac{\ln k}{k} 2^k \qquad \alpha_c \simeq \alpha_s \simeq 2^k$$

- Previous solvable algorithms

Pure Literal ("PL")	$o(1)$ as $k \to \infty$
Walksat, rigorous Walksat, non-rigorous	$\frac{1}{6} \cdot 2^k/k^2$ $2^k/k$
Unit Clause ("UC")	$\frac{1}{2}\left(\frac{k-1}{k-2}\right)^{k-2} \cdot \frac{2^k}{k}$
Shortest Clause ("SC")	$\frac{1}{8}\left(\frac{k-1}{k-3}\right)^{k-3} \frac{k-1}{k-2} \cdot \frac{2^k}{k}$
SC+backtracking ("SCB")	$\sim 1.817 \cdot \frac{2^k}{k}$

- Our prediction for BP guided decimation $\alpha_a \simeq \frac{e}{k} 2^k$

- Algorithm Fix by A. Coja-Oghlan works up to $\frac{\ln k}{k} 2^k$

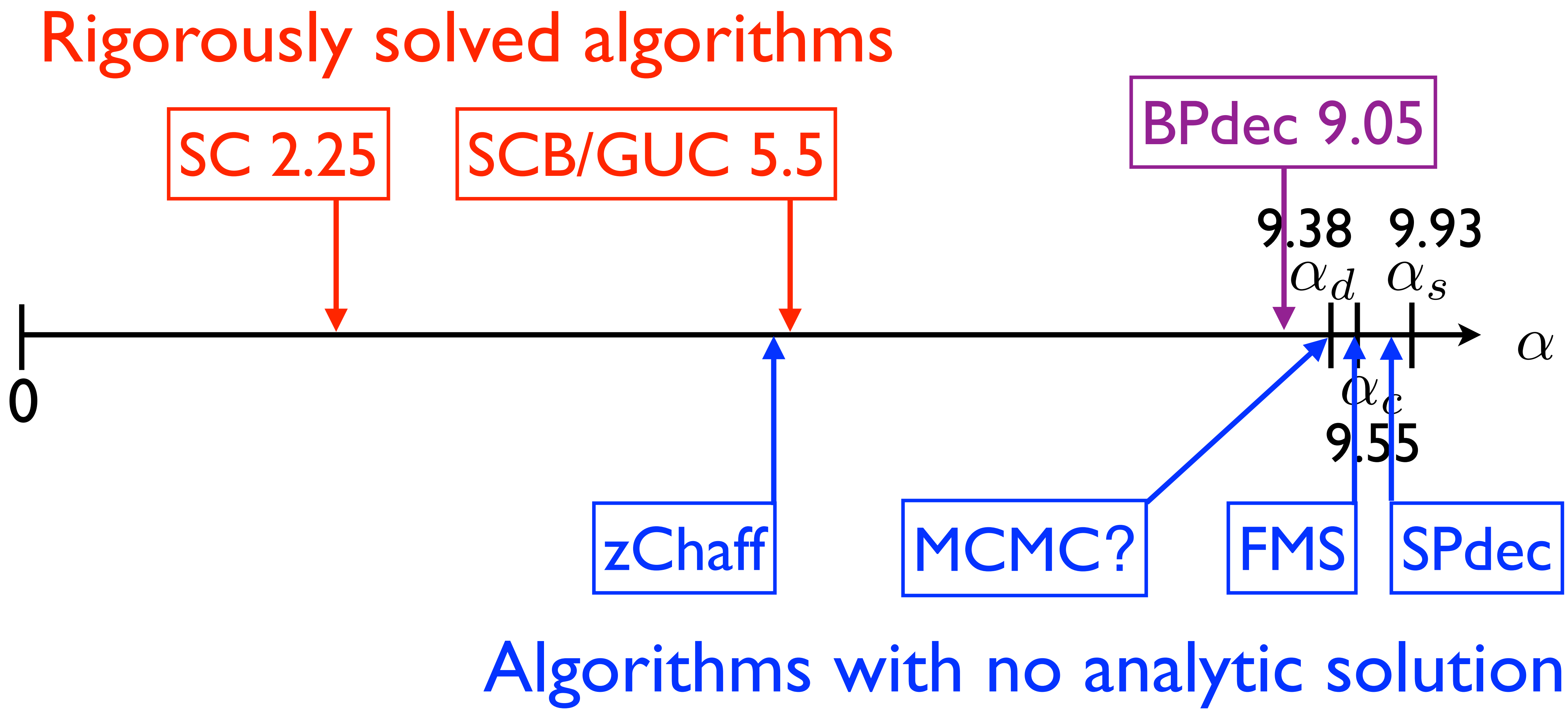
Large k limit

(pros and cons)

- Allows for rigorous proofs :-)
  - Phase transition in the decimation process proved rigorously by A. Coja-Oghlan and A. Pachon-Pinzon

- May lead to assertions that are not always true :-((especially for small k values)
  - Clustering threshold = rigidity threshold

Performance of algorithms for random 4-SAT



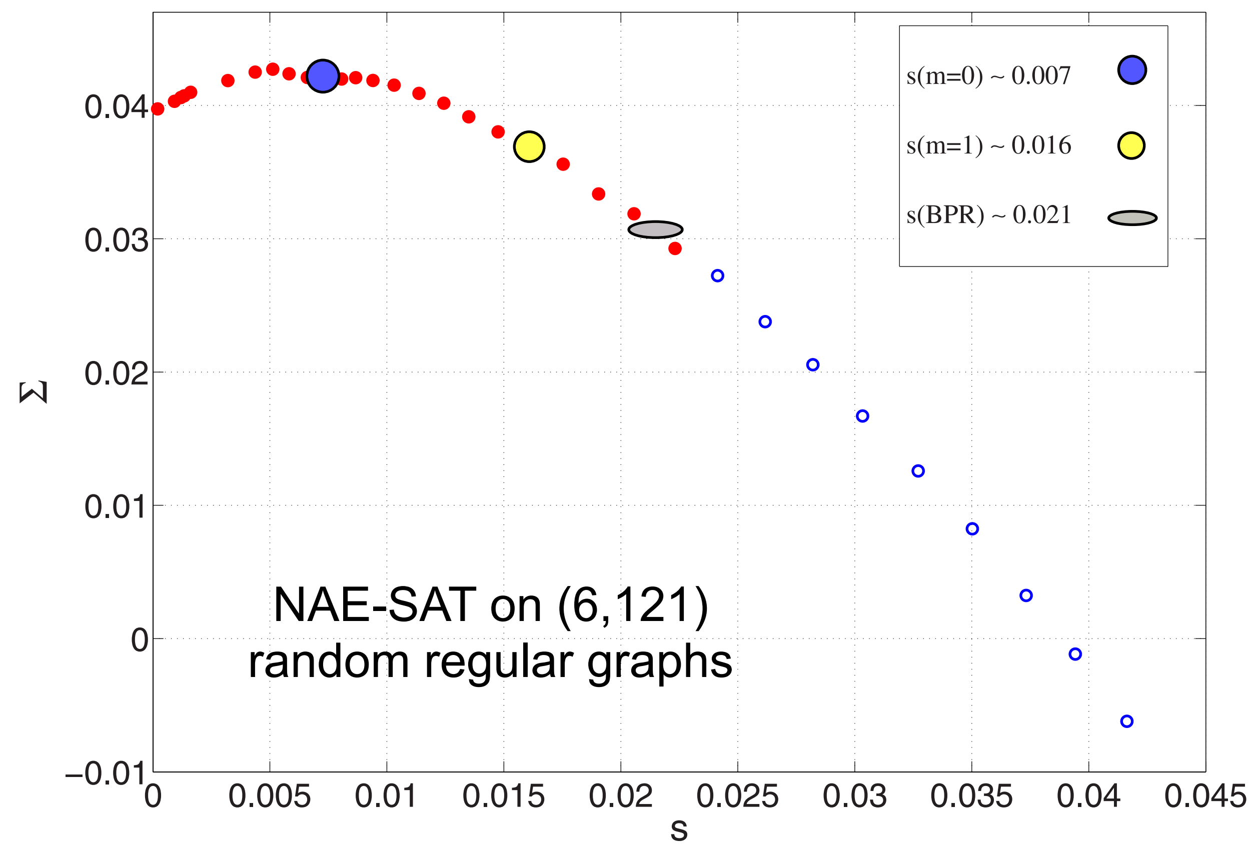
In summary...

- We have solved the oracle guided decimation algorithm
  -> ensemble of decimated CSP
- BP guided decimation follows closely this solution
- We improve previous algorithmic thresholds α_a
  from 5.56 (GUC) to 9.05 for k=4
  from 9.77 (GUC) to 16.8 for k=5
- **Conjecture:** in the large N limit for $\alpha < \alpha_a$
  BP guided decimation = oracle guided decimation
- **Todo:** bound the error on BP marginals

A conjecture for the ultimate algorithmic threshold

- Hypothesis 1: no polynomial time algorithm can find solutions in a cluster having a finite fraction of frozen variables (frozen cluster)

- Hypothesis 2:
  smart polynomial time algorithms can find solutions in unfrozen clusters even when these clusters are not the majority



Dall'Asta, Ramezanpour, Zecchina, PRE 77 (2008) 031118

A conjecture for the ultimate algorithmic threshold

- The smartest polynomial time algorithm can work as long as there exists at least one unfrozen cluster

- **Conjecture:**
  No polynomial time algorithm can find solutions when all clusters are frozen

- Stronger condition than the rigidity transition