Phase transitions in combinatorial optimization problems
Course at Helsinki Technical University, Finland, autumn 2007
by Alexander K. Hartmann (University of Oldenburg)
Lecture 4, 27. September 2007

Remarks: Outline of Talk shoould be slide-wise, i.e. few keywords per slide.

(show sample talk of myself)

Missing appointments

3.3 Numerical Results

Ensemble $\mathcal{G}(N, c/N)$ of random graphs:
N vertices, each poss. $N(N-1)/2$ edge is present with prob. c/N.
$\rightarrow c =$ average degree
Here: $c = 2.0$.

Phase Transition

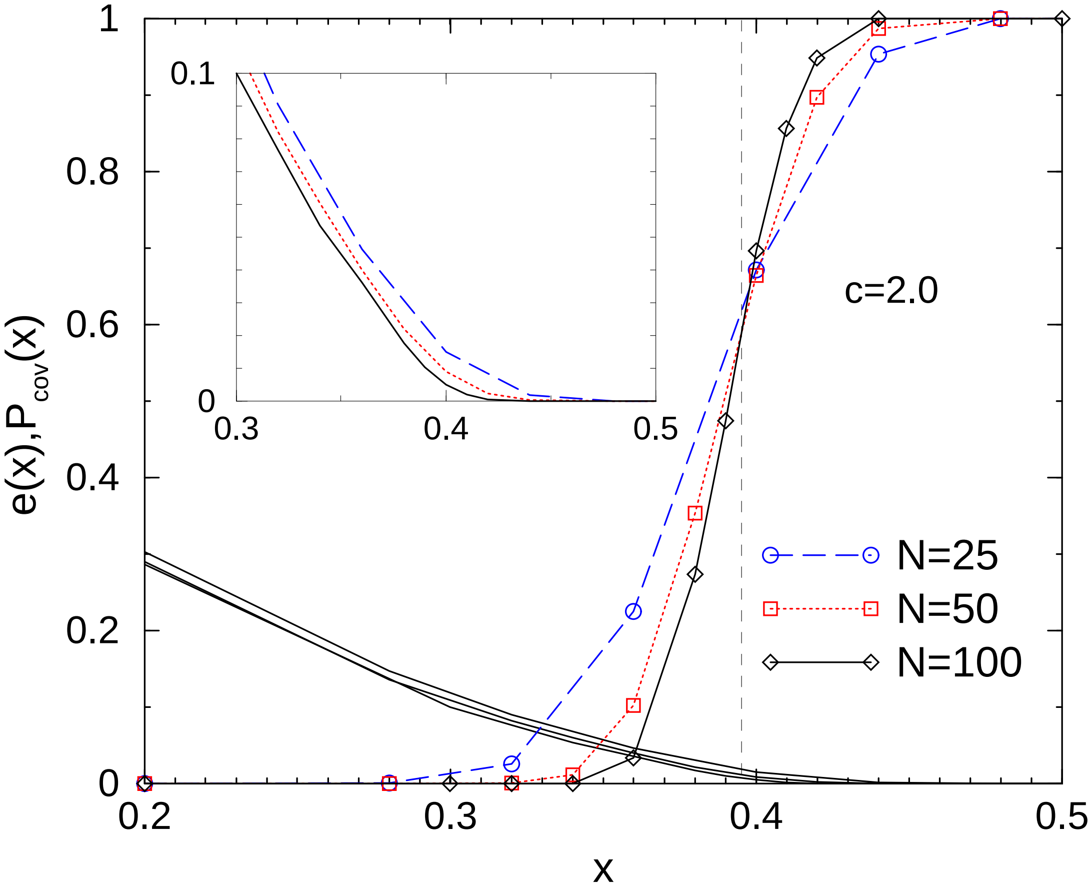


Figure 1: Probability $P_{\mathrm{cov}}(x)$ that a VC exists for a random graph ($c = 2$) as a function of the fraction x of covered vertices.
Three different system sizes $N = 25, 50, 100$ (averaged over $10^3 - 10^4$ random graphs).
Left: average energy density $e(x)$.
Inset: result for the energy in the region $0.3 \leq x \leq 0.5$.

Running time

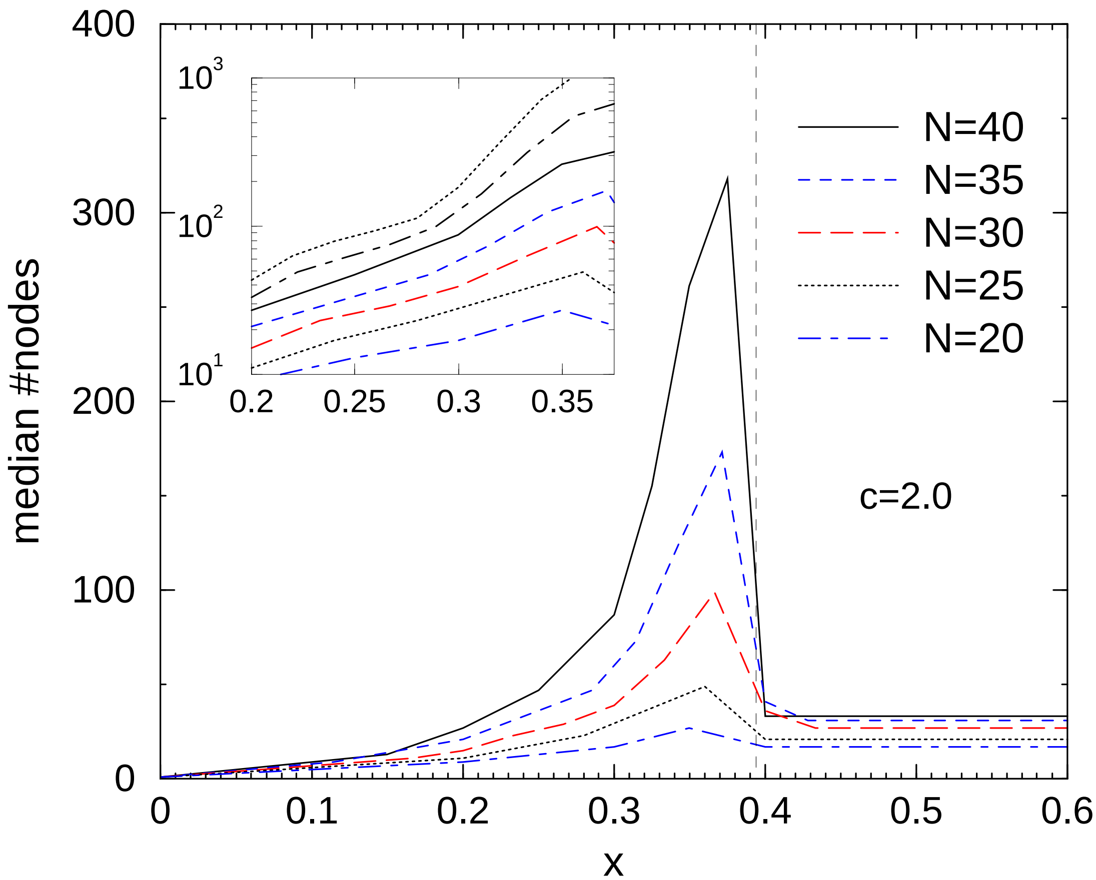


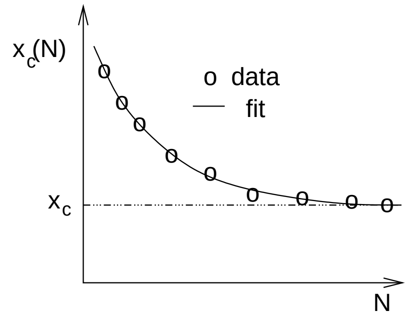
Figure 2: Time complexity of the vertex-cover algorithm = median number of nodes visited in the configuration. $N = 20, 25, 30, 35, 40$, $c = 2.0$. Right part ($x > 0.4$): Running times grows linearly. Inset: logarithmic scale (also $N = 45, 50$) $\rightarrow$ time complexity grows exponentially with N.

Finit-Size Scaling

Determine $x_{\mathrm{c}}(N)$ for different graph sizes N
fit to the data a function

$$x_{\mathrm{c}}(N) = x_{\mathrm{c}} + aN^{-b} \qquad (1)$$

(frequently found behavior in physical systems)



Matches well:

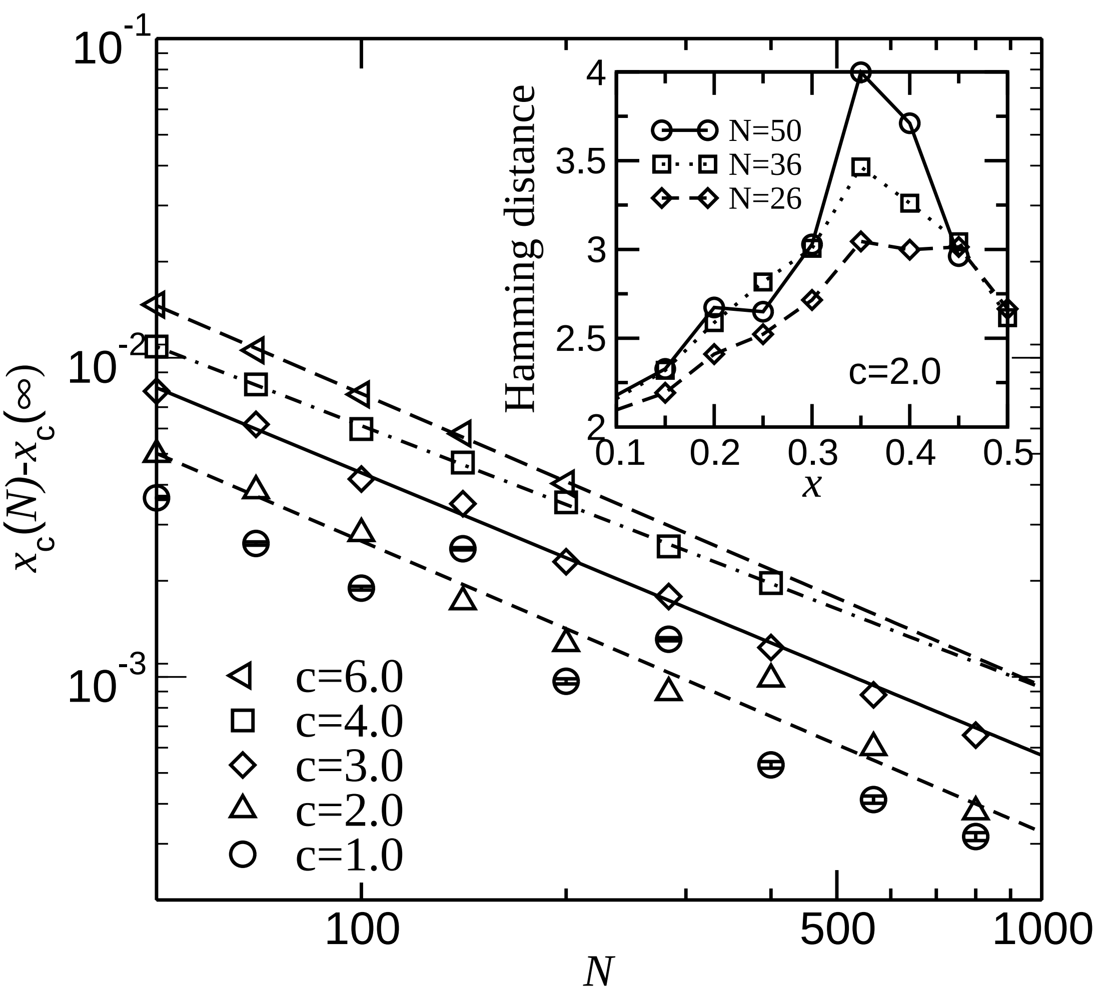


Figure 3: Finite-size scaling behavior of the critical cover size. The location of the transition point $x_{\mathrm{c}}(N)$ as a function of graph size N for different average degree c. Inset: scaling of the correlation volume as a function of x for different sizes. Error bars are, at most, of the order of the symbol size.

b does not depend much on the connectivity c:
$b(c=2) = 0.91(9)$, $b(3) = 0.88(4)$, $b(4) = 0.82(4)$, $b(6) = 0.92(11)$

Phase diagram

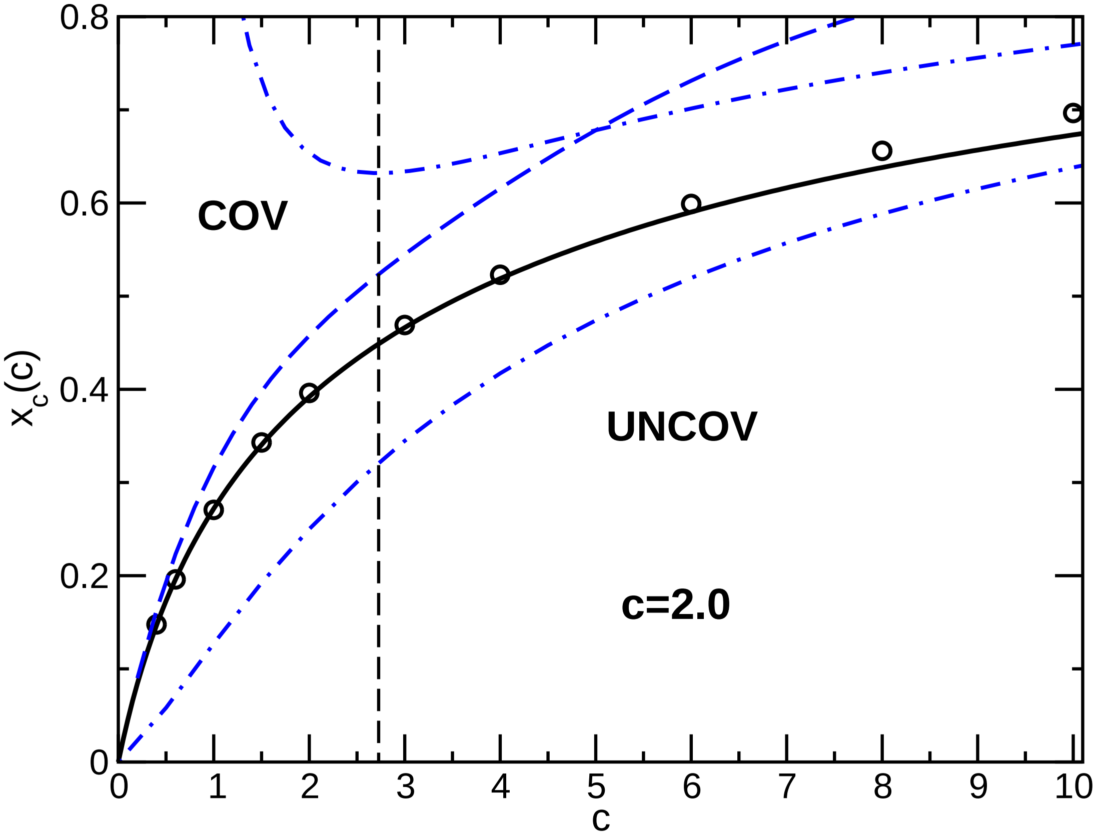
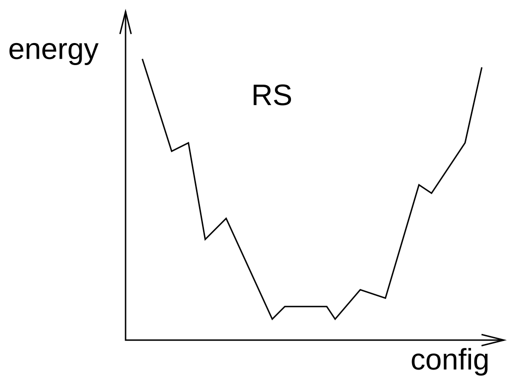


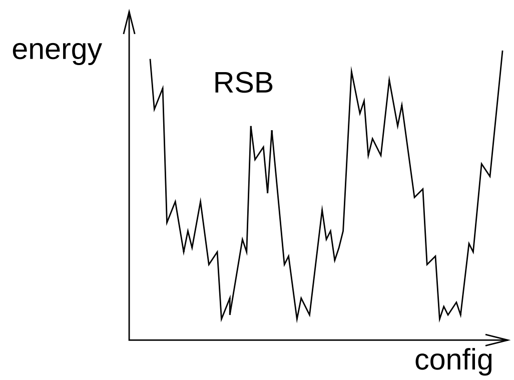
Figure 4: Phase diagram. Circles: numerical simulations. Line: analytical result. Bounds: dashed/dashed-dotted lines. Vertical line at $c = e \approx 2.718$.

Analytical Result:

$$x_c(c) = 1 - \frac{2W(c) + W(c)^2}{2c}\,, \tag{2}$$

$W(c)$: Lambert-W function: $W(c)\exp(W(c)) = c$.
Result exact until $c = e \approx 2.718$: Assumption of Replica symmetry (RS) ($\leftrightarrow$ simple organisation of phase space) is true.
$c > e$: Replica symmetry breaking (RSB) ($\leftrightarrow$ complex phase space) $\rightarrow$ cannot be calculated exactly here.





Note: percolation at $c_{\text{crit}} = 1.0 < e$!

3.4 Leaf-Removal algorithm

Speed up for finding minimum-size VCs (optimization problem 2)

Basic idea: only full VCs wanted
$\rightarrow$ all edges must be covered
$\rightarrow$ all edges $\{i, l\}$ to leaves l (degree 1) must be covered
$\rightarrow$ either i or l must be covered
$\rightarrow$ no harm in covering i, i.e. neigbours of leaves.
$\rightarrow$ all edges incident to i are covered
$\rightarrow$ maybe more leaves generated

algorithm leaf-removal ($G = (V, E)$)
begin
  Initialize $V' = \emptyset$
  while there are leaves i (i. e.vertices with degree $d_i = 1$) **do**
  begin
    Let j be the neighbor of a leaf i
    cover j, i. e., $V' = V' \cup \{j\}$
    Remove all edges adjacent to j from E
    Remove i and j from V
  end
  return (V')
end

Running time: $\mathcal{O}(M)$ ($= O(N)$ for random graphs with fixed c)

Remaining graph: called core

Each component of core: treated with brand-and-bound algorithm.

Example: Leaf removal

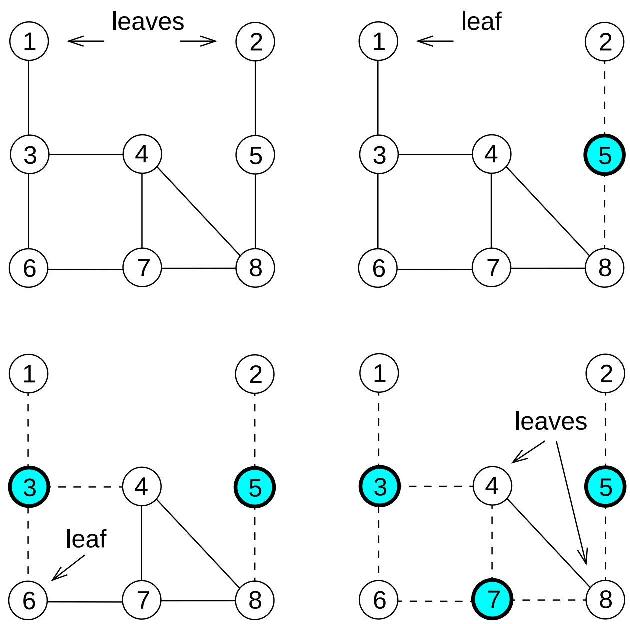


Figure 5: Example of the leaf-removal algorithm. Upper left: initial graph, vertices 1 and 2 are leaves. Upper right: graph after the first iteration, vertex 5 has been covered (shown in bold) and the incident edges removed (shown with dashed line style). Bottom: graph after second and third iteration.

Previous sample graph
Two leaves, vertices 1 and 2

Iteration 1: say vertex 5 (neighbor of 2) is covered. (edges $\{2,5\}$ and $\{5,8\}$ are covered and removed)

It. 2: v. 3 covered (neighbor of 1) ($\rightarrow$ edges $\{1,3\}$ and $\{3,6\}$)
$\rightarrow$ new leaf (vertex 6)

It. 3: v. 7 covered (neighbor of 6)
$\rightarrow$ just one egde left (i.e. two leaves 4,8)

It. 4: v. 8 covered

→ min. VC found! □

Note: for random graphs, connectivities $c < e$: core is not extensive
→ core = collections of components of $\mathcal{O}(\log N)$ (Bauer and Golinelli, Europ. Phys. J. B **24**, 339 (2001))
→ per component: running time of brand-and-bound algorithm exponential in $\log N$, i.e. polyonmial in N
→ min VC can be found typically in $\mathcal{O}(N^k)$ for $c < e$.

3.5 Monte Carlo (MC) simulations

General simulation approach used in (statistical) physics.
See books:

- M. E. J. Newman und G. T. Barkema, Monte Carlo Methods in Statistical Physics (Clarendon Press, Oxford, 1999).
- D. P. Landau and K. Binder, A Guide to Monte Carlo Simulations in Statistical Physics, (Cambridge University Press, Cambridge 2000).

Works very well for VC on random graphs, even for large c.

Basic idea: interpret VCs as configuration of physical system, a hard-core lattice gas, MC introduces a dynamics into the system. Idea: dynamic is guided to lead into minimum VCs.

3.5.1 The hard-core lattice gas

Arbitrary covers V_{vc} on graph $G = (V, E)$ including those larger than the minimum VC:
→ at least at one end-point of any edge there is a covering mark

Define uncovered vertices as occupied by particles.
→ not allowed: particles at both endpoints of an edge

particles have chemical radius of one = a hard-core repulsion