Phase Transitions in Problem Difficulty

Carlos Cotta

Departamento de Lenguajes y Ciencias de la Computación
Universidad de Málaga

http://www.lcc.uma.es/∼ccottap

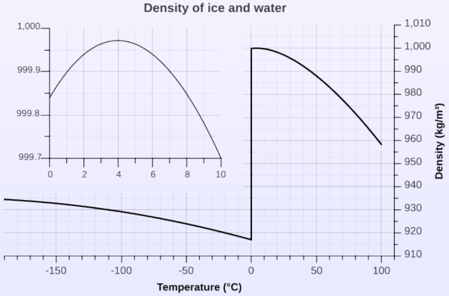
Comput Eng, Softw Eng, Comput Sci & Math – 2021-2022

UNIVERSIDAD
DE MÁLAGA

Index

What Are Phase Transitions?

Phase transition is a term borrowed from Physics to denote the passing of a system from one state to another, which entitles a change (often abrupt and discontinuous) in its properties.



For example, water abruptly increases volume by $\sim 9\%$ when it changes from liquid to solid.

Phase Transitions in Problem Solving

The same idea can apply to the resolution of computational problems.

Let γ be a control parameter describing some structural property of instances of the problem $\mathcal{P}$ under consideration. The solvability of the instance and even the behavior of a search algorithm S applied to an instance $I(\gamma) \in \mathcal{P}$ may depend on this control parameter.

For instance, for some values of the control parameter the problem instance may be easy to solve, but for other values it may be hard to solve.

Phase Transitions and Constrained Satisfaction Problems

Consider the case of constrained satisfaction problems: find values to variables $X = \{x_1 \ldots, x_n\}$ subject to constraints $C_1, \ldots, C_m$, where each constraint C_i is a logical statement about the values of the variables in X.

A control parameter describing the number and/or structure of constraints may define two homogeneous subsets of instances:

1. under-constrained instances: the constraints are not very limiting and hence these instances have many solutions.
2. over-constrained instances: the constraints are very limiting and hence these instances do not have any solution.

Criticality in Constrained Satisfaction Problems

Both subsets may in some cases be easy to solve (i.e., a solution can be easily found in under-constrained instances, or it can be quickly determined that it does not exist in over-constrained instances).

In the transition between these two regions we have critically-constrained instances:

- Any such instance may or may not have a solution.
- If it does, it may have very few solutions so finding any of these can be time-consuming.
- If it does not, exhausting the search space and establishing this fact can be hard as well.

Example: 3-SAT

The 3-Satisfiability Problem (3-SAT)

Let $X = \{x_1 \ldots, x_n\}$ be a collection of Boolean variables. An instance of 3-SAT is defined as a logical formula

$$\bigwedge_{i=1}^{m} \underbrace{\bigvee_{j=1}^{3} v_{ij}}_{C_i}$$

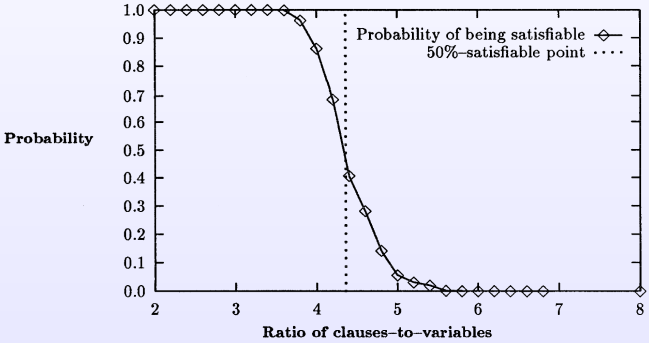
where each v_{ij} is either some $x_k \in X$ or its negation $\neg x_k$.

Let the control parameter be $\gamma = m/n$, i.e., the ratio between the number of clauses and the number of variables.

Example: 3-SAT

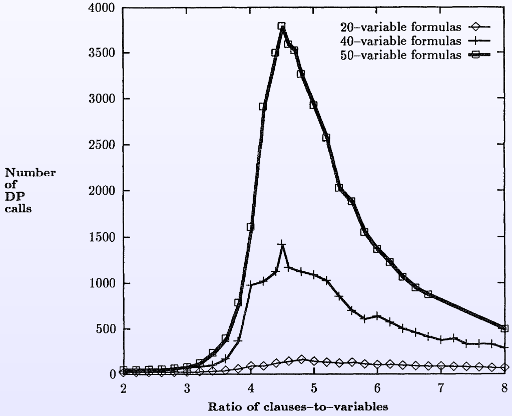
3-SAT solvability

Instances with $\gamma < 4$ are under-constrained. Instances with $\gamma > 4$ are over-constrained.



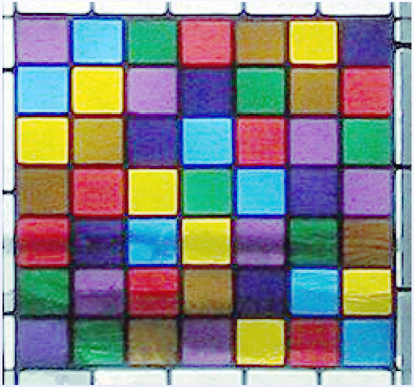
Example: 3-SAT

3-SAT hardness



The peak of difficulty coincides with the transition between these two regions. The maximum is around $\gamma = 4.3$.

Latin Squares



Latin squares are combinatorial puzzles popularized by mathematician Leonhard Euler in the mid 1700s, but whose origins can be traced back to at least the early 1700s in Korea.

They have applications in design of experiments, and in error correcting codes.

Latin Squares

Latin Squares

An order-n Latin square is a numerical puzzle defined over a board:

1. Its size is $n \times n$ squares.
2. Each square can contain a number between 1 and n.
3. No number is repeated in any column or row.

Notice that Latin squares can be regarded as a simplified version of Sudoku.

Constructing a Solution

			4					
	1		6		9		5	8
9				3				
3								
				5	8	7		
2							1	3
6	2				1	4		
	4						8	
					2	1	7	

Feasible solutions are constructed by filling a square at a time. Each square can contain a number in $[1, n]$.

Some squares are initially fixed and cannot be modified throughout the search process.

We will use as a control parameter the proportion of clues (i.e., the number of squares initially fixed).

$$\gamma = \frac{\#clues}{n^2}$$

Goals

1. Build a backtracking algorithm to solve an arbitrary instance of the problem.
2. Use this algorithm to find the critical complexity point. To this end:
    1. The backtracking algorithm must count the number of recursive steps performed.
    2. Test cases with different values of the control parameter will be used to empirically assess the complexity of the algorithm as a function of this parameter.

Specific Bibliography

T. Hogg, B.A. Huberman, C.P. Williams
Phase transitions and the search problem
Artificial Intelligence 81(1–2):1-15, 1996

D. Mitchell, B. Selman, H. Levesque
Hard and Easy Distributions of SAT Problems.
Proceedings of the Tenth National Conference on Artificial Intelligence, AAAI, 1992

Image Credits

- Water density: By Klaus-Dieter Keller, created with QtiPlot, Font: Liberation Sans - Own work, CC BY-SA 3.0, `https://commons.wikimedia.org/w/index.php?curid=19093965`
- 3-SAT figures: D. Mitchell, B. Selman, H. Levesque, ©AAAI, 1992
- Stained glass at Gonville and Caius College, in Cambridge (UK): Photographed by Schutz. Clipped and color-adjusted by ccottap.