Computational Complexity and Predictability of Chaotic Dynamical Systems

or, relating computational and physical complexity

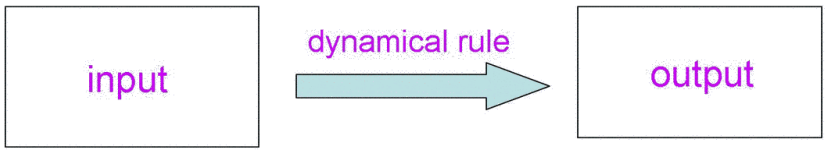
S. N. Coppersmith
Department of Physics
University of Wisconsin

Funding: NSF-DMR, NSF-EMT (with E. Bach, R. Joynt, D. von Melkebeek)

Computational complexity: study of how computational resources needed to solve a problem grow with size of problem specification

Relation to study of complex systems:

- Gives insight into resources needed to compute dynamical evolution of a given model
- A computer is a dynamical system



Results from computational complexity yield insights into complex systems, and vice versa

CS $\rightarrow$ Physics: Interactive proofs and the predictability of dynamical systems at exponentially long times

Physics $\rightarrow$ CS: New insight into the P versus NP question

Computational complexity, and dynamical systems at exponentially long times

In a chaotic dynamical system, nearby orbits diverge exponentially in time — number of bits needed to characterize orbits grows linearly with time.

This statement is true for both continuous (in space) and discrete dynamical systems.

Is there a fundamental distinction between continuous and discrete dynamical systems?

Computational complexity theory: there are fundamental differences between continuous and discrete systems at exponentially long times

We don't usually ask about exponentially long times because exponentials blow up very fast

$((2^N)(10^{-12}\ s) >$ age of the universe when N=150)

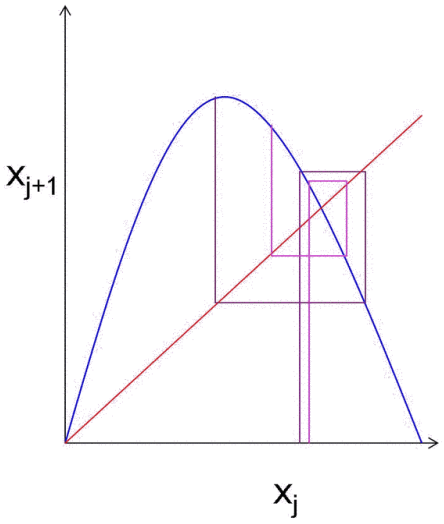
But recent results in computational complexity theory imply that such calculations can be contemplated for discrete dynamical systems.

Spatially continuous dynamical system example: the logistic map

$x_{j+1} = \lambda x_j(1-x_j) \quad [x \in (0,1)]$

When $\lambda > \lambda_c$, orbits obtained starting from nearby points diverge exponentially in time. So number of bits needed to distinguish orbits grows linearly with time. (Can show analytically when $\lambda=4$.)

Exponentially long time $\rightarrow$ exponentially many bits needed to characterize orbits.

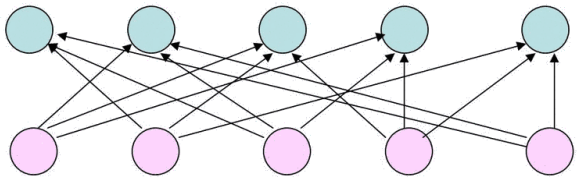


Discrete dynamical system example: Kauffman net (N-K model)

N Boolean variables ($\sigma_i(t) = 0$ or 1), each with K randomly chosen inputs

$$\sigma_i(t+1) = f_i(\sigma_{j_1(i)}(t), \sigma_{j_2(i)}(t), \ldots, \sigma_{j_K(i)}(t))$$

f_i = randomly chosen Boolean functions



Can compute exponentially many configurations in polynomial space (PSPACE)

Any question about the dynamics of a Boolean net, even at exponentially long time, can be verified by a polynomially bounded computational agent.

Shamir, 1992: PSPACE=IP

IP=class of problems that can be verified via an Interactive Proof

Interactive proofs (IP):

Babai; Goldwasser, Micali, Rackoff

Proof protocol with:

1. polynomially bounded Verifier with a supply of random coins
2. computationally unbounded but not-necessarily-honest Prover

Prover tries to convince Verifier that a given statement is true. Problem is in IP if there is a protocol for which Verifier is convinced with certainty if statement is true, and convinced with low probability if statement is false.

IP is a generalization of NP

NP: Problems for which a solution can be verified in a number of steps that grows no faster than polynomially with the size of the problem specification.

Prover provides answer, and Verifier checks it

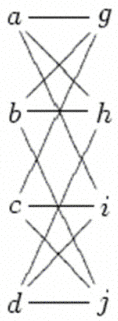
IP: Verifier can flip coins and can ask Prover (a polynomially bounded number of) questions.

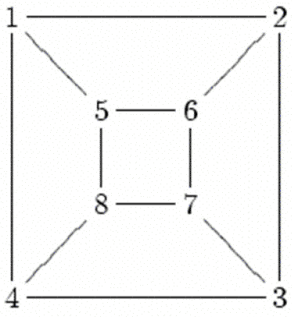
How big is the class IP?

At first, it was felt that IP would only be a little bigger than NP (after all, verifier is polynomially bounded).

But IP is much bigger than NP!

Graph isomorphism (GI): are two graphs related by a relabelling of vertices?





isomorphism:

f(a)=1
f(b)=6
f(c)=8
f(d)=3
f(g)=5
f(h)=2
f(i)=4
f(j)=7

An isomorphism can be checked in polynomial time $\Rightarrow$ GI is in NP

No polynomial time algorithm for GI is known

Graph nonisomorphism — demonstrate that two graphs are not isomorphic

Graph nonisomorphism is in co-NP (complement of NP); need to demonstrate that no relabelling exists
Not known whether problem is in NP.

First sign of power of Interactive Proofs: Interactive proof for graph nonisomorphism (Goldreich, Micali, Wigderson, 1991)

Interactive proof for graph nonisomorphism

(Goldreich, Micali, Wigderson, 1991)

Prover claims two graphs A and B are not isomorphic. (Both verifier and prover know what A and B are.)

1. Verifier flips coin and picks a graph and a permutation of vertices, and sends permuted graph to Prover
2. Prover reports whether permuted graph is isomorphic to graph A or to graph B

Prover can always be correct only if A and B are not isomorphic

IP = PSPACE

Any problem in PSPACE can be written as a polynomial of bounded degree in each variable and with polynomially many variables (the rub is that it is not known how to calculate the coefficients in less than exponential time)

Lund, Fortnow, Karloff, Nisan (1992)

But a polynomially bounded Verifier can tell if a prover were to provide the coefficients of this polynomial.

An aside: quantum dynamics

Calculating quantum dynamical system is in PSPACE (Bernstein & Vazirani), so quantum evolution is amenable to Interactive Proof.

Summary

- Computational complexity theory provides new insight into complex dynamical systems
- Discrete dynamical systems with N degrees of freedom are in IP
- Does this result have implications for the predictability of such systems?