

# 1. Natural Computing

Marten van Dijk, vandijk@engr.uconn.edu

- \* What is computation? (How do we, humans, think?)  
 ↓  
 Philosophical question  
 ↓  
 Needs abstract thinking as opposed to researching and implementing an optimal design for some HW/SW system.  
 ↓  
 Need a rather mathematical language.

- \* Natural computing is the field of research that investigates
  1. Human-designed computing (models) inspired by nature.
  2. Computing (Information processing) taking place in nature.

Nature:	Science:	
Functioning of the brain	→ neural computation	} Alg.
Darwinian evolution of species	→ evolutionary "	
Intercellular communication	→ cellular automata	
Behavior of groups of organisms	→ swarm intelligence	
Natural immune systems	→ artificial immune systems	
<del>Membrane</del> compartmentalized ways in which cells process information	→ membrane computing	

Quantum Computing } Models → Replace HW  
Molecular " }  
Allowing massive parallelism

12

- 2: Computational nature of
- self-assembly (nanosciences)
  - biochemical reactions
  - bacterial communication
  - systems biology

Information processing has become a central notion and thinking habit in many disciplines

\* Go over Syllabus.  $\Delta$  Expectations.

\* Cellular Automata

- Discrete models of physical systems (Physics)
- Models of massively parallel computation (Comp. Sc.)
- Discrete-Time dynamical systems (Mathematics)

Model of real-world phenomena:

- massively parallel
  - homogeneous
  - all interactions are local
  - add reversibility
  - " conservation laws
- } local update rule



Simple automata are capable of performing arbitrary computation tasks (computational universality)

13

lectures:

1. Basic concepts

2. Universalities

- Turing (ability to compute any recursive function)
- Intrinsic (ability to simulate any other cell. automaton)
- Simulation of boolean circuits

3. Reversibility: every configuration has only one previous configuration

4. Lattice Boltzmann methods as a natural modeling framework.

## \* Neural Computation

Neural networks that aim to

- gain understanding of biological neural systems (synapses, neurons, interaction)

- solve problems in artificial intelligence (without resembling a real biological system)

↓  
classification

pattern recognition

} Support Vector Machines (dist. function)

} Bayesian techniques (prior distribution)

Large models can be learned from huge databases in reasonable amount of time.

## Lectures:

24

1. Spiking Neural Networks (take into account the firing rate + moment of spike firing)
2. Neurofuzzy Networks: effective in modeling and control of non-linear processes
3. SVM: learns non-linear functions.
4. Mathematical approaches to model biological neural networks

## \* Evolutionary Computation

Algorithms gleaned from models of organic evolution

Principles:

- natural selection
  - genotypic variation
- } → search & optimization tasks

Modern evol. alg. → ability to adapt & self-adapt their strategy

## Lectures:

1. Basic concepts evol. alg.
2. " " genetic "
3. Optimizing multiple conflicting object functions.
4. Memetic algorithms combine stochastic global search techniques with evolutionary alg.



## \* Molecular Computation

15

Program molecules so that they perform

- a desired computation (solving a hard problem)
- fabricate a desired object
  - DNA-based logic circuits
  - DNA "walkers" that move along a track
  - molecular motors
- ~~a specific~~ control the functioning of a specific molecular system

Data encoded as (bio)molecules, e.g. DNA strands.

Lectures:

1. Foundations
2. Molecular computing machines
3. Biomolecular devices: autonomous & programmable
4. Membrane computing: Based on cells defined by membranes that selectively communicate with each other.

## \* Quantum Computing

Understand the fundamental limits of information processing by nature itself

Shor's factorization / Grover's search alg.

Applied in crypto (which is about transfer of information)

# lectures

16

1. Quantum inf. processing  $\rightarrow$  very mathematical.
2. Bell's inequalities  $\rightarrow$  philosophy of physics
3. Algorithms
4. Physical Implementation

Next week CMO1 on Cellular Automata.

L7

- Read 1 - 2.4 in more detail
- Get rough idea of 2.5
- Try to understand some of 3.1. Do you get Ex. 2?  
and Ex 3
- In 3.2, do you get Ex 4, Ex 5
- Glance over 4-6:

What are tilings?

Why are the alg. questions of interest?

How do CAs relate to dynamical systems?

⇒ See directory CellularAutomata-Kari for more details if you are interested. (It has definitions of Turing machines, topology, undecidability etc.)