Bio-inspired Algorithms

CS454 AI-Based Software Engineering

Shin Yoo

Biomimicry

Imitation of the models, systems, and elements of nature

for the purpose of solving complex human problems.

`https://en.wikipedia.org/wiki/Biomimetics`

Morpho Butterfly

- The blue colour is not pigmentation; it is **structural colouration**.
- Microstructures (ridges, cross-ribs, etc) in the scale of the butterfly interferes with specific wavelengths of the light, resulting in the blue colour.
- Mirasol display technology from Qualcomm is based on the reflective properties of the butterfly.

https://commons.wikimedia.org/wiki/File:Morpho_didius_Male_Dos_MHNT.jpg

Burrs

- Swiss electrical engineer, George de Mestral, had to remove burdock burrs (seeds) from his cloths and his dog’s furs whenever he returned from walks in Alps.
- Eventually he invented Velcro hooks in 1951.

https://en.wikipedia.org/wiki/Biomimetics#/media/File:Bur_Macro_BlackBg.jpg

What about algorithms?

Supervised

Unsupervised

Machine Learning

RL

Learning

Planning

NLP

Perception

Logical Reasoning

Intelligence in General

Motion

AI: All Types of Machine Intelligence

Problems to Solve

Approaches

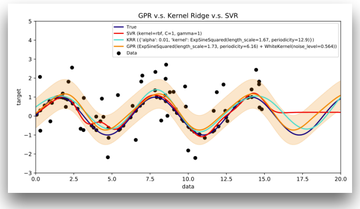
Symbolic

$\vdash \forall x \neg A(x) \rightarrow \neg A(a)$

$\vdash A(a) \rightarrow \neg \forall x \neg A(x)$

$\vdash A(a) \rightarrow \exists x A(x)$

Statistical



Soft computing

ANN

GA

Meta-heuristic

Deep Neural Networks

Biomimicry in Algorithms

- In some sense, almost all algorithms we learn in this course are results of biomimicry: annealing, evolution.
- Artificial Neural Network and the connectionism is an explicit act of biomimicry.

Particle Swarm Optimisation

- A machine learning technique loosely based on a flock of birds searching for food. Imagine a flock, in which:
  - the intensity of the cry of a bird is proportional to how much food (insects) it can find at its current location
  - birds know who are nearby
  - birds know who is crying out the loudest

Particle Swarm Optimisation

- The flock has a good chance of converging to the location with the most food, if each bird follows the direction which is the combination of the following three:
  - keep the current direction
  - return to the location where it found the most food
  - move towards the neighbouring bird whose cry is the loudest
- GA is competitive; PSO is cooperative.

Particle Swarm Optimisation

$$x_i^{t+1} = x_i^t + v_i^{t+1}$$

$$v_i^{t+1} = av_i^t + b(x_i^p - x_i^t) + c(x_j^t - x_i^t)$$

$$v_i^{t+1} = av_i^t + br_s(x_i^p - x_i^t) + cr_t(x_j^t - x_i^t), r_s \in [0, s], r_t \in [0, t]$$

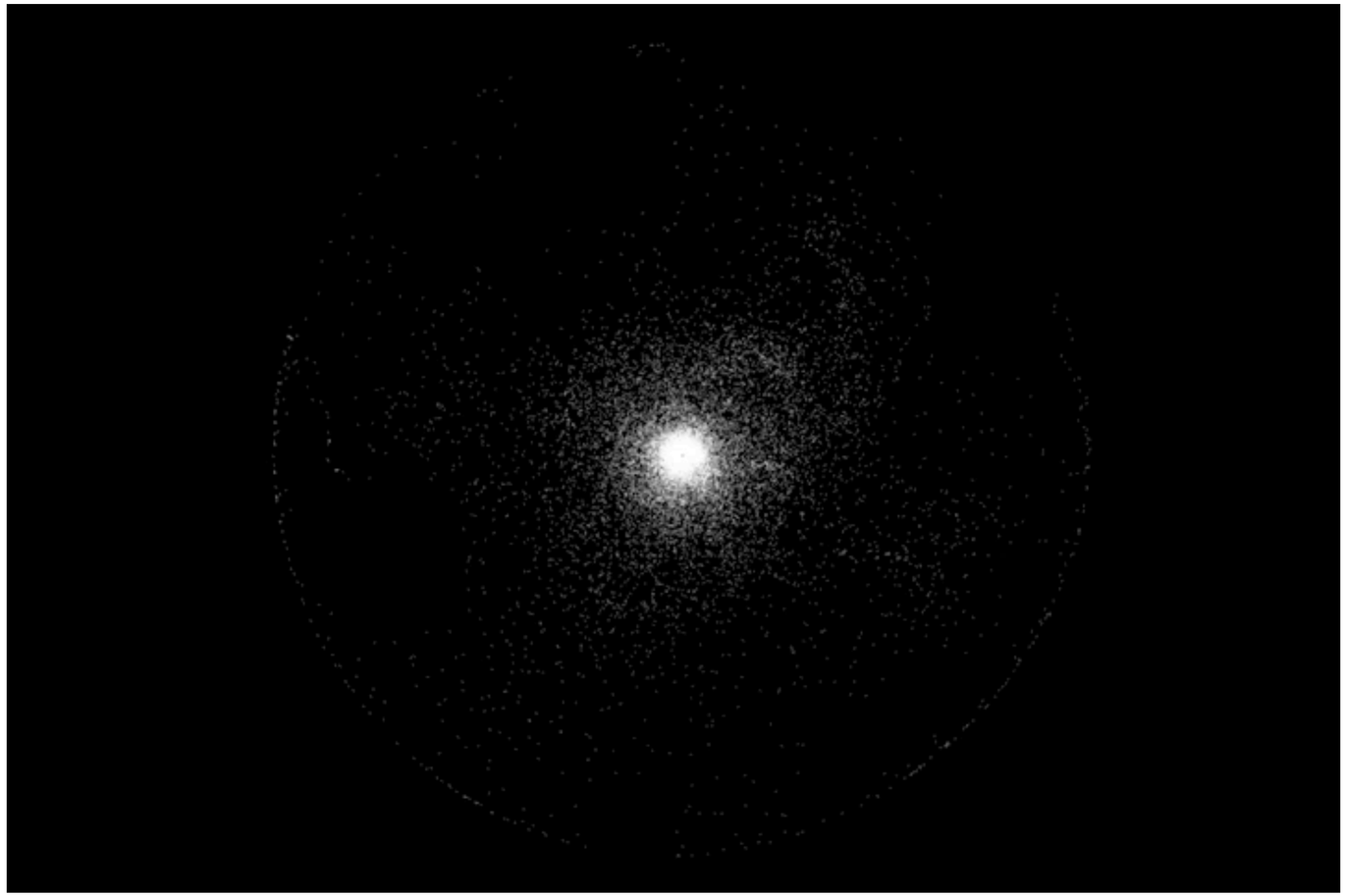
Particle Swarm Optimisation

- Inherently designed for continuous space/real values.
- Can be mapped/demapped to integer domains.
- Trickier to adapt to combinatorial/discrete spaces.
  - Operations in PSO: distance between two solutions (-), multiplication of a velocity (*), adding two velocities (+), and applying the velocity to positions (+)
  - These three operators need to be re-defined over the discrete sets.

https://www.youtube.com/watch?v=bbbUwyMr1W8

Ant Colony Optimisation

- Ants in a colony needs to balance exploitation and exploration in terms of foraging for food.
- Scouts leave pheromone trail when they explore, find food, and return; subsequent workers follow the deposit of pheromones, and leave their own, strengthening the trail.



Emergence of exploration network in a colony of Argentine ants *Linepithema humile* (video from icouzin Lab in Princeton)

http://icouzin.princeton.edu/pheromone-trail-networks-in-ants/

Ant Colony Optimisation

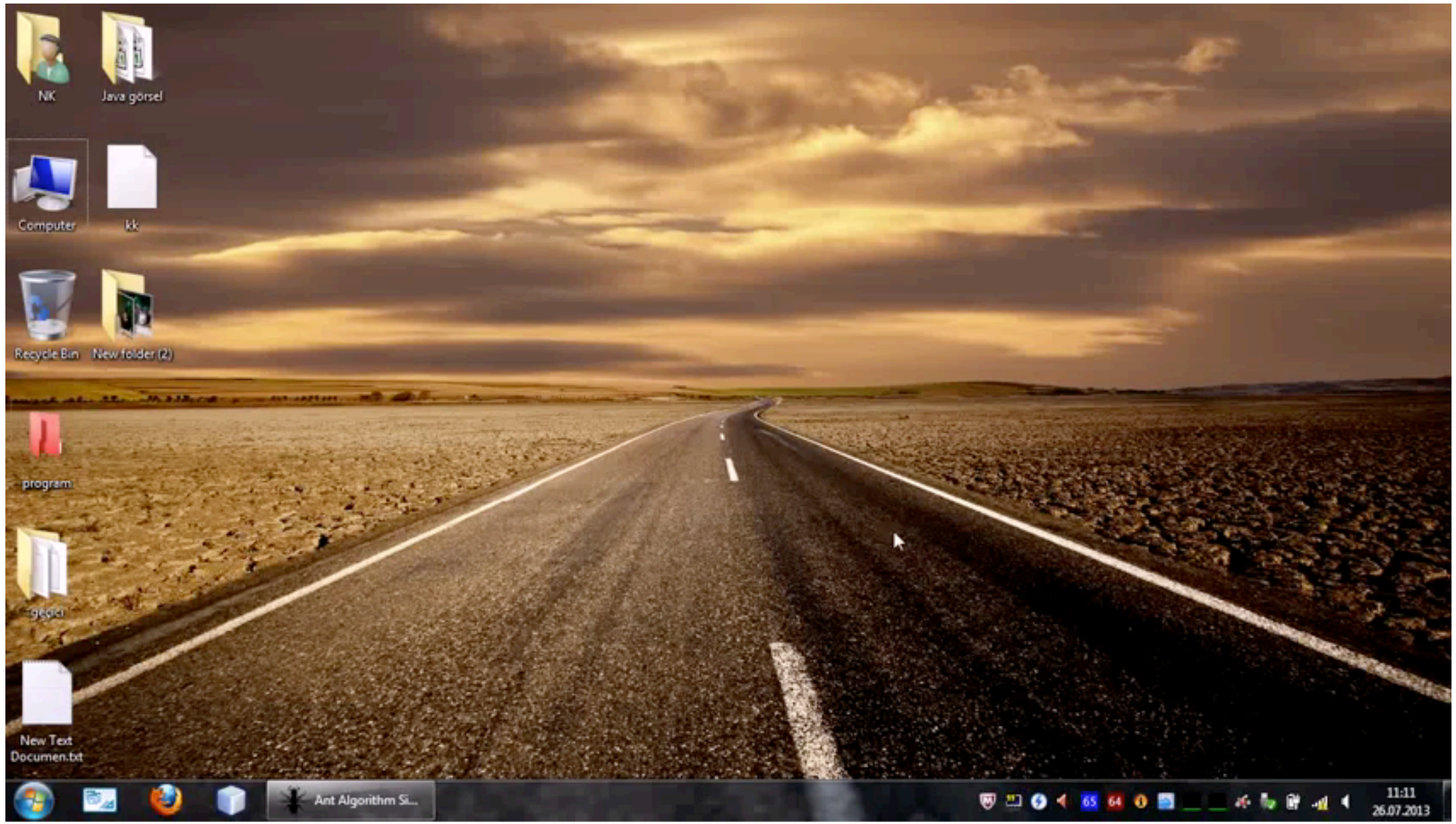
- We want to mimic ants to explore graphs (for example, to solve TSP).
- However, there is no guarantee that the scout ant always finds the best path.
- Solution: our pheromone trail is artificial, so make it evaporates (?)

ACO for TSP

- Initialisation: drop ants on random nodes on the graph. Also, deposit small amount of pheromone on all edges uniformly.
- Ants choose which edge to cross probabilistically, considering the length of the edge and the amount of pheromone on the edge.
- When ants finish a tour, they retrace their tour, depositing pheromones in amounts inversely proportional to the length of the route.
- Before starting another round, pheromones evaporates a little.
- Eventually, ants converge on the shortest path.

ACO for TSP

- Probability of ant k choosing the edge from node i to j: $p_{ij}^k = \frac{\tau_{ij}^a \eta_{ij}^b}{\sum_{h \in J^k}^H \tau_{ih}^a \eta_{ih}^b}$, where η_{ij} is the visibility of the node j from i, calculated as $\frac{1}{l_{ij}}$ (l_{ij} is the length of the edge). Weights a and b balances whether ants chooses only based on the shortest edge ($a = 0$), or only based on the pheromone deposit ($b = 0$). H is the available edges; J^k is the set of nodes not visited by ant k yet.
- Pheromone deposit: $\Delta\tau_{ij}^k = \frac{Q}{L^k}$, where Q is the estimated shortest tour and L^k is the length of the tour by ant k.
- Amount of pheromone on each edge: $\Delta T_{ij} = \sum_k^M \Delta\tau_{ij}^k$
- Evaporation: $\tau_{ij}^{t+1} = (1 - \rho)\tau_{ij}^t + \Delta T_{ij}$, where ρ is the evaporation rate constant.



https://www.youtube.com/watch?v=eVKAIufSrHs

Strength of ACO

- Edge selection is probabilistic: a small number of ants will traverse paths that are not shortest.
- When the graph changes, ACO can adapts with second-best partial solutions.

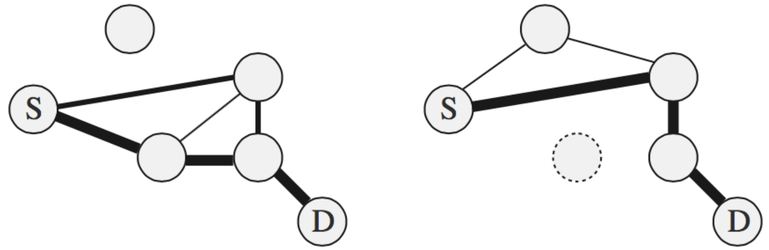


Figure 7.8 *Left*: Virtual ants maintain multiple paths between source and destination nodes. Shorter paths are traversed by more ants (thicker line). *Right*: If a node (or edge) fails, ants immediately use and reinforce the second shortest path available.

Dario Floreano and Claudio Mattiussi, *Bio-inspired Artificial Intelligence*, MIT Press

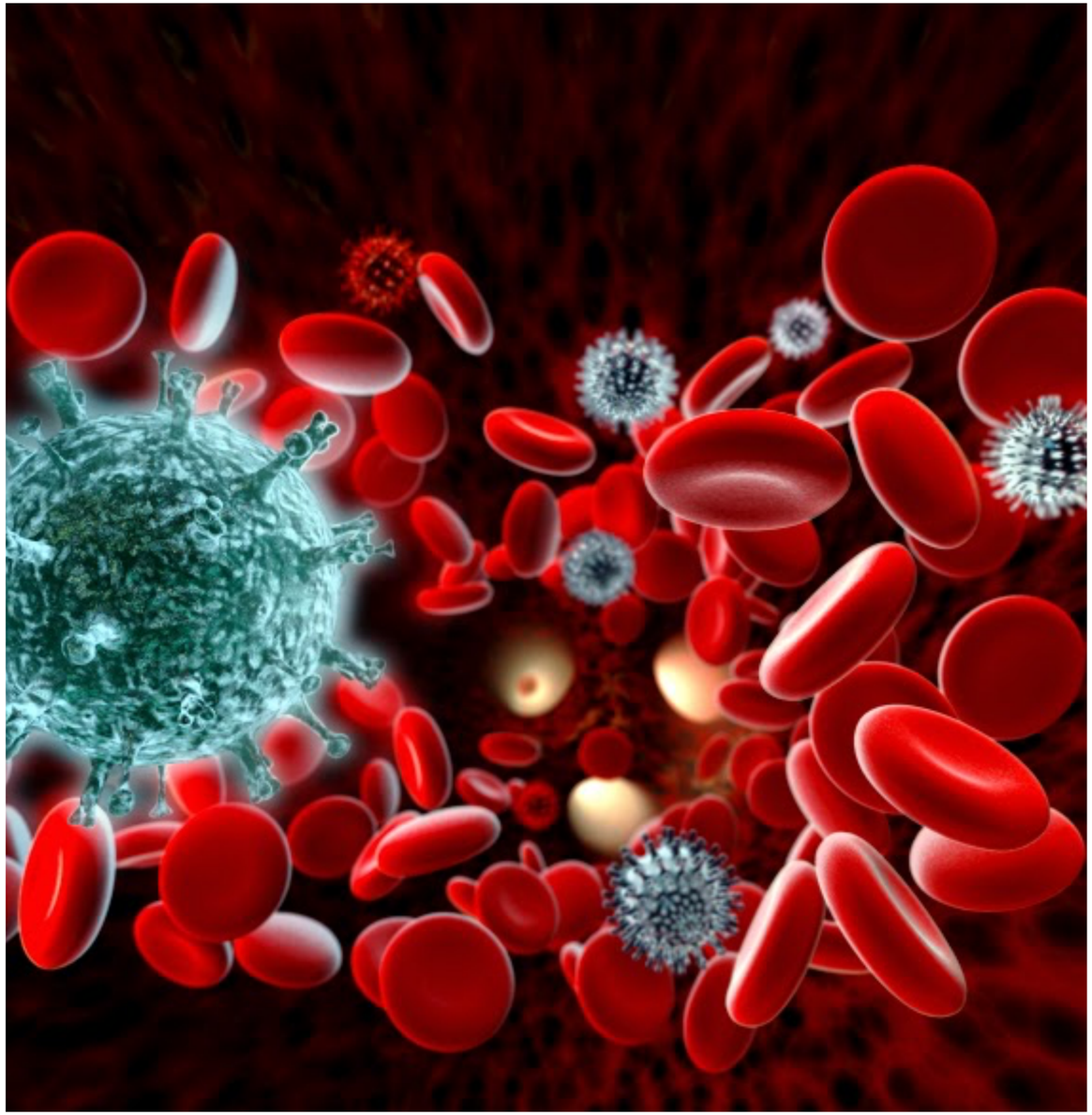
Applications

- Evaluations on simulated US T1 Internet backbone and NTT telephone networks shows equal performance to industry standard routing algorithms in terms of throughput, but much better in terms of packet delay.
- Problems should be able to be represented as path-finding; there are attempts to solve combinatorial optimisation, for example, but not so successful compared to routing.

G. Di Caro and M. Dorigo, *AntNet: Distributed Stigmergetic Control for Communications Networks*, Journal of Artificial Intelligence Research, Volume 9, pages 317-365, 1998

Artificial Immune System

- Biological immune systems are vast complex systems; simulating the entire mechanism would be pointless.
- Certain stages can be emulated for pattern recognition, etc.



“The problem of protecting computer systems can be viewed generally as the problem of learning to distinguish *self* from *other*.”

Self-nonself discrimination in a computer, Forrest et.al, 1994

Non-self Detection

- Many applications in security: how to detect what is not normal?
  - Cannot anticipate all attacks.
  - Cannot anticipate all normal usages.
  - Having a fixed set of rules/detectors does not work.

Negative Selection Algorithm

- Maintain a pool of multiple detection algorithms that are unique. Each may protect different sites/parts/etc.
- Detection is probabilistic: we risk allowing intrusion at one site, but this is mitigated by having multiple detectors.
- A robust system should detect (probabilistically) all foreign activities, and not just known intrusion patterns.

Negative Selection Algorithm

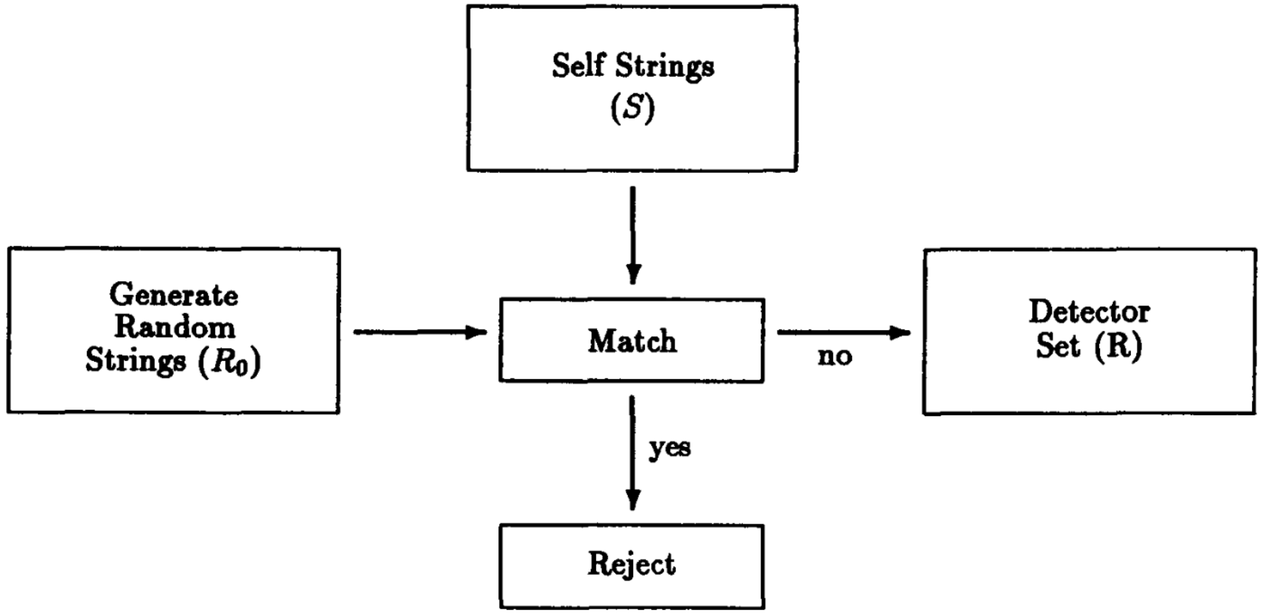


Figure 1: Generation of Valid Detector Set (Censoring) .

S. Forrest, A. Perelson, L. Allen, and R. Cherukuri, *Self-nonself discrimination in a computer,* Research in Security and Privacy, 1994.

Negative Selection Algorithm

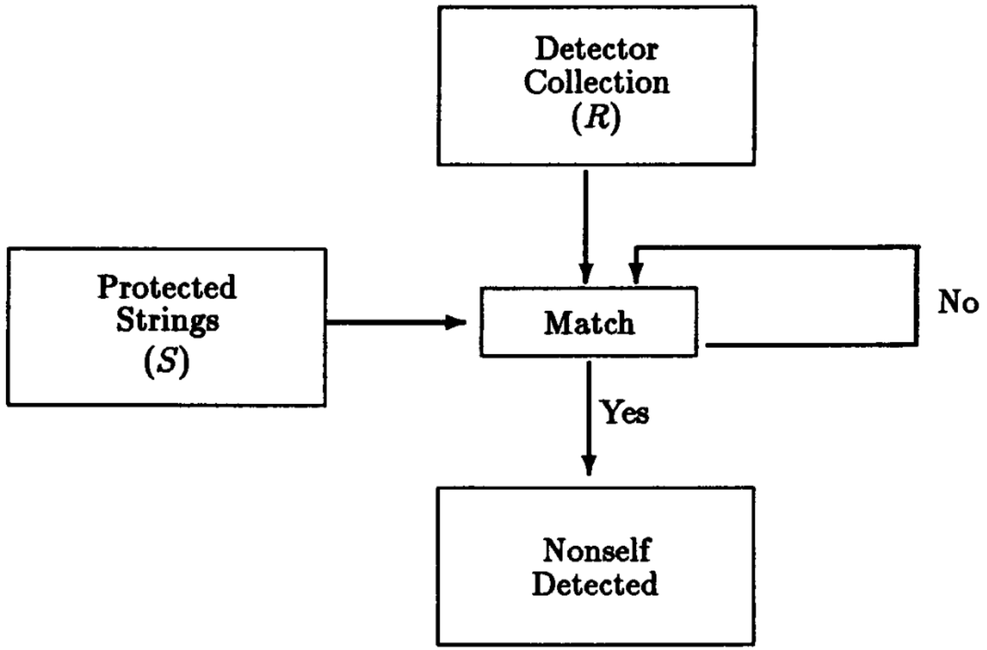


Figure 2: Monitor Protected Strings for Changes.

S. Forrest, A. Perelson, L. Allen, and R. Cherukuri, *Self-nonself discrimination in a computer,* Research in Security and Privacy, 1994.

Probabilistic Detection

- `r`-contiguous match: two strings match if `r` characters from the same location matches.
- For strings of length `l`, composed of `m` alphabets, the probability of `r` contiguous letters matching is sufficiently low for 1:1 matching.
- With multiple detectors, we can reasonably learn to distinguish the self probabilistically.

m	r	l	P_M
2	8	32	0.0502023
2	8	64	0.108697
2	8	128	0.2151
2	8	256	0.391316
2	16	32	0.000137329
2	16	64	0.000381437
2	16	128	0.000869474
2	16	256	0.00184483
128	8	32	$3.33067 * 10^{-16}$
128	8	64	$7.77156 * 10^{-16}$
128	8	128	$1.66533 * 10^{-15}$
128	8	256	$3.44169 * 10^{-15}$
128	16	32	~ 0.0
128	16	64	~ 0.0
128	16	128	~ 0.0
128	16	256	~ 0.0

Table 1: Example values of P_M for varying values of m (alphabet size), r (number of contiguous matches required for a *match*), and l (string length).

AIS: Applications

- Anomaly detection: represent what you want to distinguish as strings:
  - Network packets: matching is made between TCP fields.
  - System processes: strings are sequences of system calls.

Coevolution

- Two population evolving at the same time as parts of a larger dynamic system: can be either cooperative or competitive.
- Very interesting concept, but it seems like we understand very little.

Sorting Network

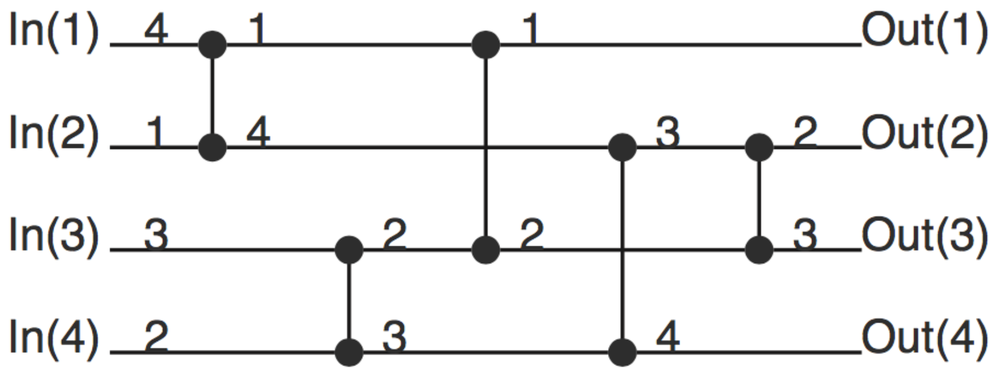
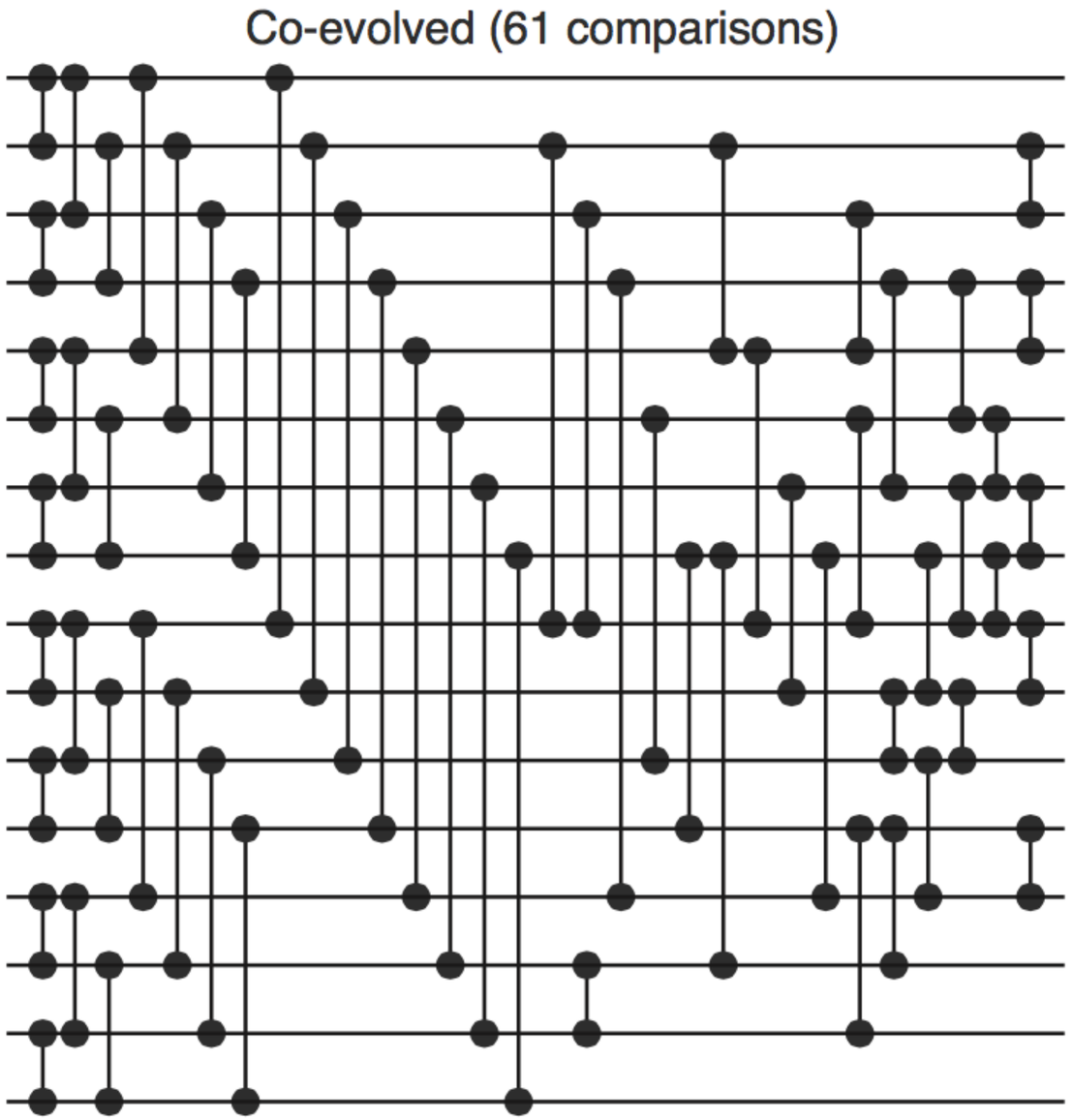


Figure 7.21 A sorting network for arranging four arbitrary numbers in nondecreasing order. Vertical connectors represent comparators that exchange the position of two incoming elements if the lower is smaller than the higher. Redrawn from (Knuth 1998, p. 221).

Coevolving Sorting Networks

- Hillis, 1992
  - Shortest sorting network for 16 inputs, developed by human, uses 60 comparisons.
  - By co-evolving sorting networks and test cases (i.e. inputs), Hillis achieved 61 (65 without coevolution).



Coevolution of Programs and Tests

- Given pre- and post-condition for programs:
  - One population of programs: take pre-condition satisfying input, and tries to evolve programs that satisfy post-condition.
  - One population of test cases: against evolved programs, tries to satisfy precondition but break the post-condition.

Coevolution of Programs and Tests

- Target programs
  - MaxValue
  - AllEqual
  - TriangleClassification
  - Swap (two elements in an array)
  - Order (swap two elements in an array only if they are out of order)
  - Sort
  - Median

Coevolution of Programs and Tests

Table 2
Configurations in which extra functions were added to the base set of GP primitives. These functions are correct implementations of the specifications we address in our case study.

Configurations	Added primitives
Order_2	
Sorting_2	
Sorting_3	
Median_2	
Median_3	
Median_4	

Table 3
Number of correct evolved programs out of 100 independent runs for each program versio

Programs	RS	COE	SSP
MaxValue	8	12	15
AllEqual	0	3	3
TriangleClassification	0	0	13
Swap	0	12	18
Order_1	0	0	0
Order_2	0	18	91
Sorting_1	0	0	0
Sorting_2	2	0	0
Sorting_3	97	97	86
Median_1	0	0	0
Median_2	0	0	0
Median_3	17	8	16
Median_4	99	96	99

AlphaGo Zero

From Wikipedia, the free encyclopedia

AlphaGo Zero is a version of DeepMind's Go software AlphaGo. AlphaGo's team published an article in the journal *Nature* on 19 October 2017, introducing AlphaGo Zero, a version created without using data from human games, and stronger than any previous version.[1] By playing games against itself, AlphaGo Zero surpassed the strength of AlphaGo Lee in three days by winning 100 games to 0, reached the level of AlphaGo Master in 21 days, and exceeded all the old versions in 40 days.[2]

Training artificial intelligence (AI) without datasets derived from human experts has significant implications for the development of AI with superhuman skills because expert data is "often expensive, unreliable or simply unavailable."[3] Demis Hassabis, the co-founder and CEO of DeepMind, said that AlphaGo Zero was so powerful because it was "no longer constrained by the limits of human knowledge".[4] Furthermore, AlphaGo Zero performed better than standard reinforcement deep learning models (such as DQN implementations[5]) due to its integration of Monte Carlo tree search. David Silver, one of the first authors of DeepMind's papers published in *Nature* on AlphaGo, said that it is possible to have generalised AI algorithms by removing the need to learn from humans.[6]

Google later developed AlphaZero, a generalized version of AlphaGo Zero that could play chess and Shōgi in addition to Go. In December 2017, AlphaZero beat the 3-day version of AlphaGo Zero by winning 60 games to 40, and with 8 hours of training it outperformed AlphaGo Lee on an Elo scale. AlphaZero also defeated a top chess program (Stockfish) and a top Shōgi program (Elmo).[7][8]

Part of a series on
Artificial intelligence

Major goals [show]
Approaches [show]
Philosophy [show]
History [show]
Technology [show]
Glossary [show]

V • T • E

Complex Dynamics

- Red Queen effects: features evolved earlier are lost, because the landscape (i.e. the other population) changed.
- Hard to interpret fitness changes: did I get better, or did my opponent get worse?
- It has been often observed that two populations tend to get comfortable with each other at mediocre level, instead of continuously improving each other.

Bio-inspired algorithms

- It is NOT about mimicking micro-behaviours.
- There are many hypes.
- The key is to understand the original behaviour and mechanism sufficiently, before trying to mimic them.