How a Peak on a Completely-flatland-elsewhere can be Searched for?

A Fitness Landscape of Associative Memory by Spiking Neurons

Akira Imada

Brest State Technical University
Belarus

We came accross

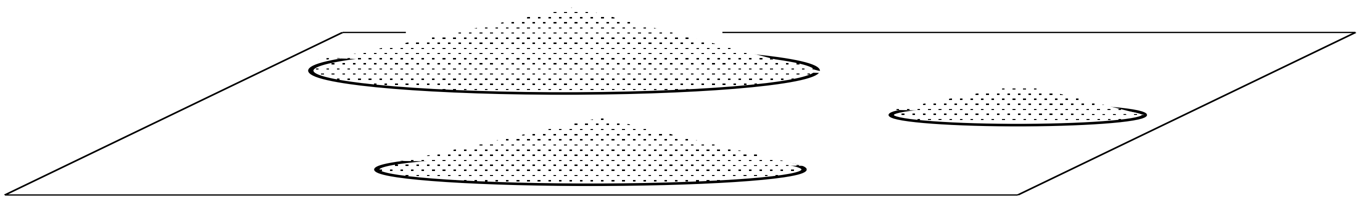
an extremely difficult
PEAK
to be searched for

in a
FITNESS LANDSCAPE

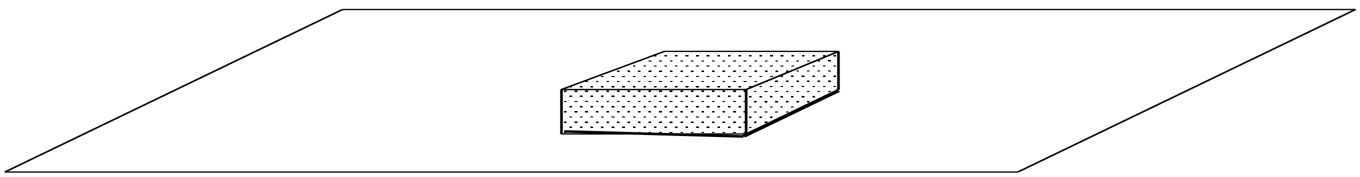
explored by an
EVOLUTIONARY COMPUTATION

□ A reason for the title — A tiny-flat-island in a huge-lake

(a)



(b)



To realize an

ASSOCIATIVE MEMORY

we use

SPIKING NEURONS

whose learning is by

EVOLUTIONARY COMPUTATIONS.

□ Contents

1. **What is Associative Memory?**

    · **Traditional Model by Hopfield.**

    vs.

    · **A Model using Spiking Neurons.**

2. **Can we evolve Spiking Neurons?**

    · **We study it by observing Fitness Landscape.**

(cont'd)

3. Results of our Experiments.

- **A downhill walk from Hebb's peak!**
- **How difficult to find other peaks!**

4. A proposition of Test-function.

5. Summary & Concluding remarks.

□**Associative Memory:**

- **stores patterns**
  - **in a distributed way (among neurons),**
- **recalls patterns**
  - **from noisy and/or partial input.**

$\Downarrow$ **i.e.**

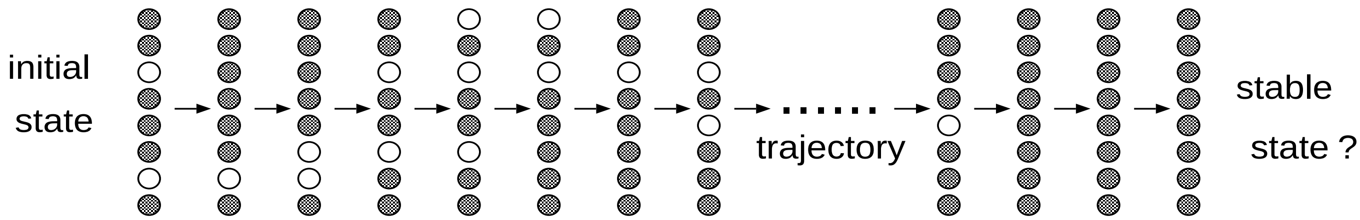
- **gives us**
  - **perfect recollection from imperfect information**

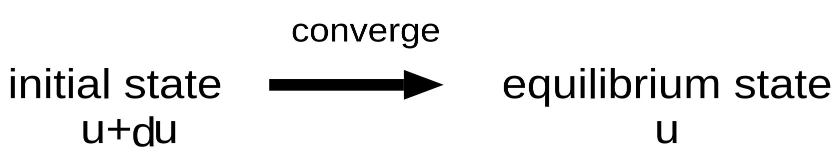
□**The Associative Memory can be realized with:**

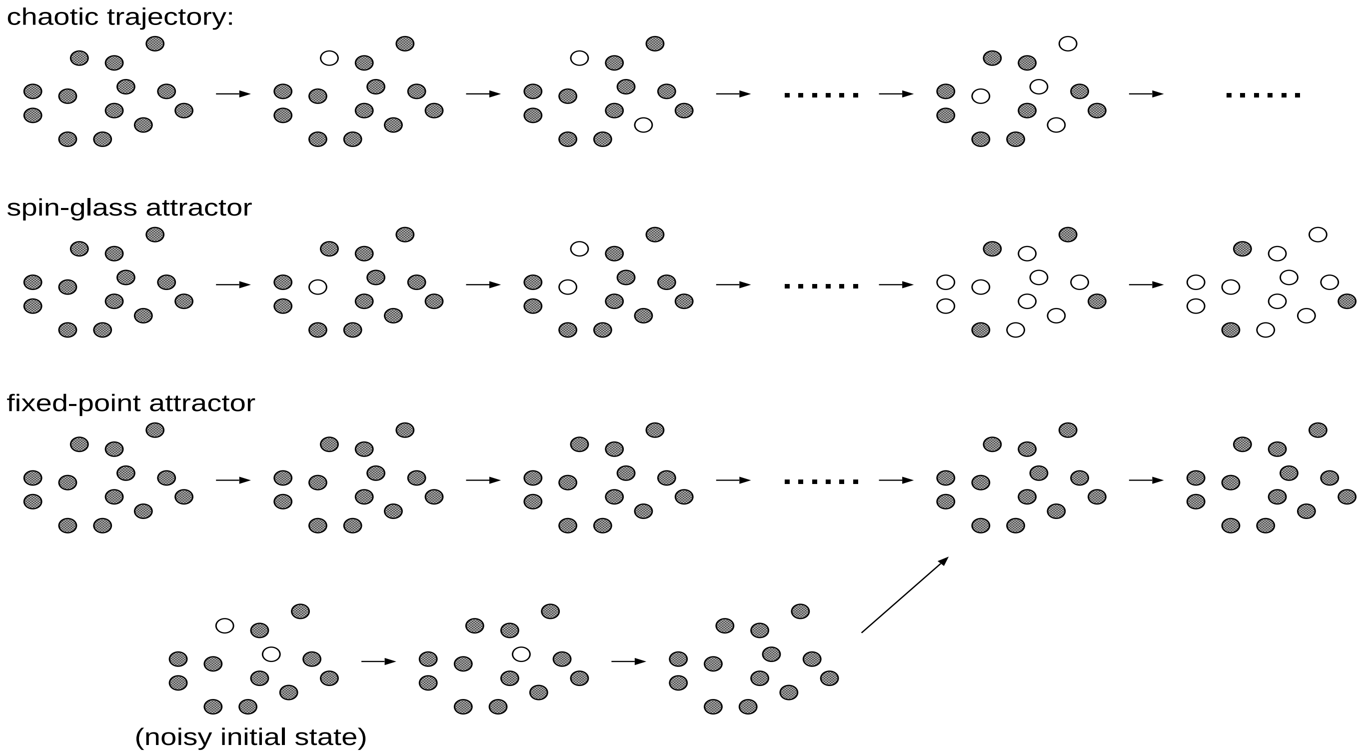
- **Fully-connected Neural Network Model (Hopfield type)**
- **Artificial Immune System Model**
- **Spiking Neurons**

  (To be more biologically plausible)
- **etc.**

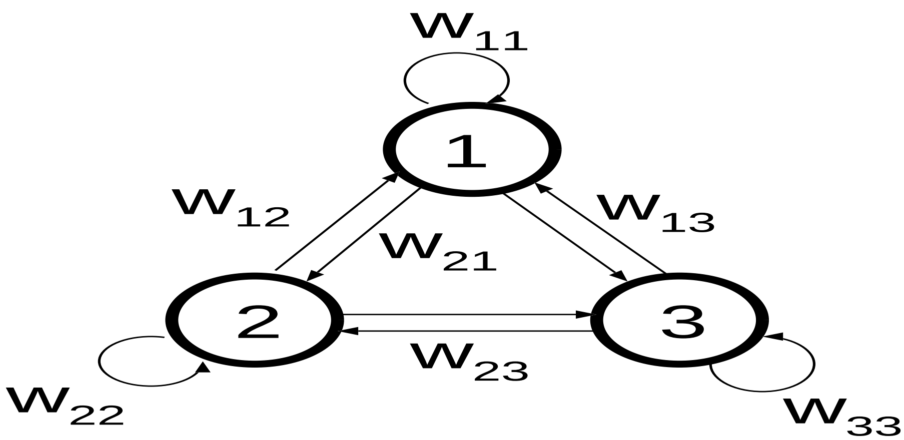
associative memory = dynamical system






chaotic trajectory:
spin-glass attractor
fixed-point attractor
(noisy initial state)

□ A Schematic Diagram of the Hopfield Model



□ State Transition of the Hopfield Model

$$s_i(t+1) = sgn(\sum_{j}^{N} w_{ij} s_j(t))$$

□ Hebbian Weights

$$w_{ij} = \frac{1}{N} \sum_{\mu=1}^{p} \xi_i^{\nu} \xi_j^{\nu}, \quad (i \neq j), \quad w_{ij} = 0.$$

(To store p patterns: $\mathbf{x}^{\mu} = (\xi_1^{\mu}, \xi_2^{\mu}, \cdots \xi_N^{\mu}) \quad \mu = 1, 2, \cdots, p$)

□ Hopfield Model has a Crucial Drawback.

- N neurons store only $2N$ patterns at most (Gardner).

$$p < 2N.$$

□ From the Hopfield Model to Spiking Neuron Model

Influence from other neurons
via
ELECTRIC CURRENTS

$\Downarrow$

changes
MEMBRANE VOLTAGE
of the neuron.

(cont'd)

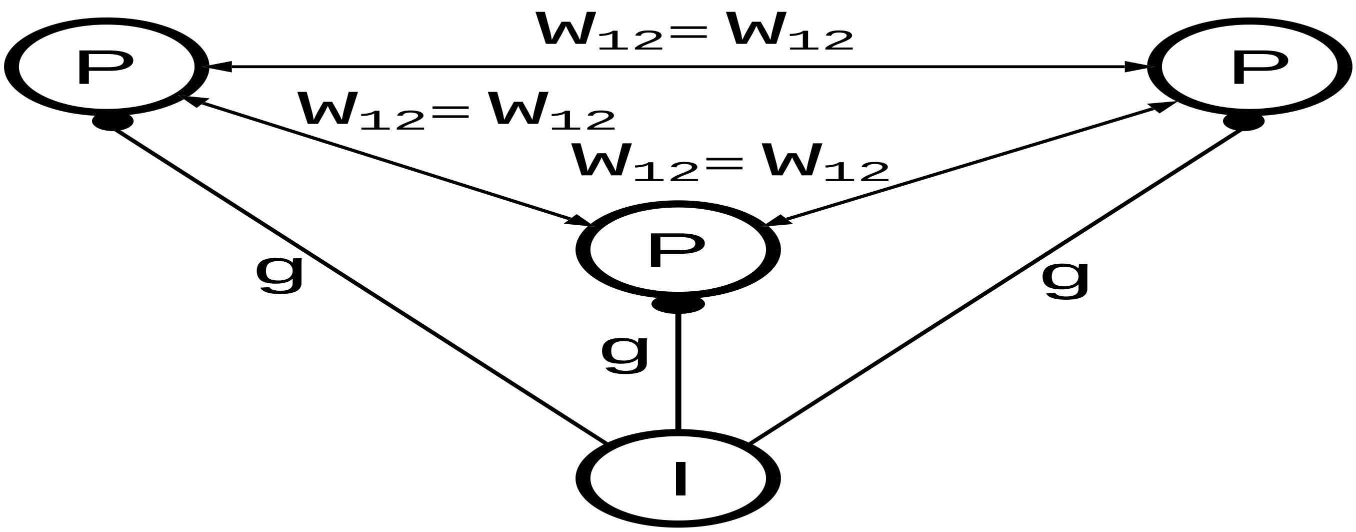
$\Downarrow$

If it exceeds a threshold

$\Downarrow$

the neuron emits
a SPIKE.

□ Pyramidal Cells & Interneurons



PYRAMIDAL CELL

$\Vert$
(positive current)
$\Downarrow$

PYRAMIDAL CELL

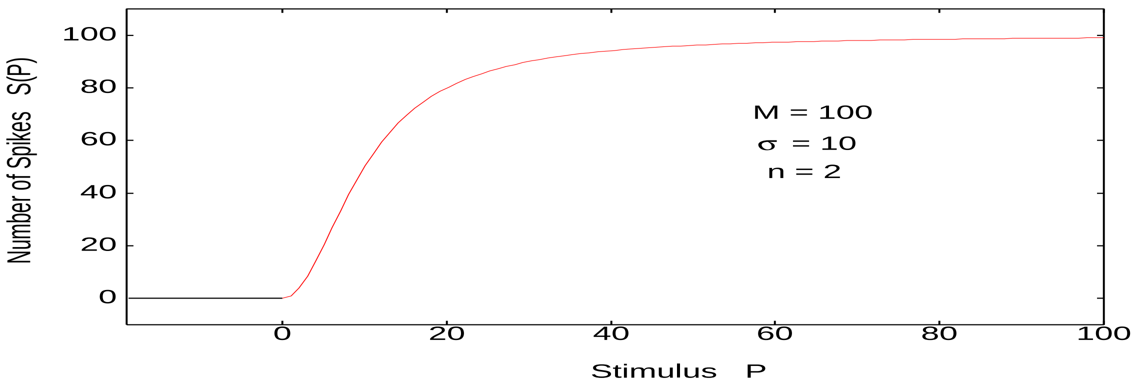
$\Uparrow$
(negative current)
$\Vert$

INTERNEURON

□ Response of a Single Neuron to an External Stimulus:

- **Spike rate vs stimuli (Naka-Rushton function)**

$$S(P) = \begin{cases} MP^n/(\sigma^n + P^n) & \text{if } P \geq 0 \\ 0 & \text{if } P < 0 \end{cases}$$



(cont'd)

- **Spike rate under time-variant stimulus:**

$$\frac{dr(t)}{dt} = \frac{1}{\tau}(-r(t) + S(P))$$

□ Response of Multiple Neurons

• Stimulus to a Pyramidal Cell

P_i in the previous Eq.

$$S(P_i) = MP_i^n/(\sigma^n + P_i^n)$$

is now expressed as

$$P_i = (\sum_{j=1}^{N} w_{ij} \cdot R_j - g \cdot G)_+^2$$

(cont'd)

- **Spiking Ratio of a Pyramidal Cell**

And

$$\frac{dr}{dt} = \frac{1}{\tau}(-r + MP^n/(\sigma^n + P^n))$$

becomes

$$\tau_R \frac{dR_i}{dt} = -R_i + \frac{100(\sum_{j=1}^{N} w_{ij}R_j - 0.1G)_+^2}{100 + (\sum_{j=1}^{N} w_{ij}R_j - 0.1G)_+^2}$$

(cont'd)

- **Spiking Ratio of Interneuron**

$$\tau_G \frac{dG}{dt} = -G - 0.07 \sum_{j=1}^{N} R_i$$

□ How to encode patterns?

For the pattern

$$(\xi_1, \xi_2, \cdots, \xi_N)$$

we encode as

$$\xi_i = \begin{cases} 1 & \text{if } R_i \geq M/2 \\ 0 & \text{if } R_i < M/2 \end{cases}$$

⇒Rate Coding

□ GA Implementation:

1. Represent a series of w_{ij} as a *population* of strings (*chromosome*).

W_{11}	W_{12}	W_{13}	···	W_{21}	·········	W_{N1}	···	W_{NN}

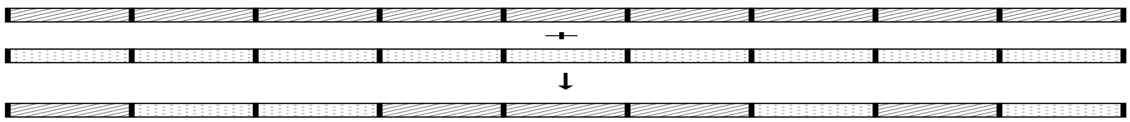
2. Evaluate *fitness* by "How good is each individual?"
3. Generate an initial *population* at random.

(cont'd)

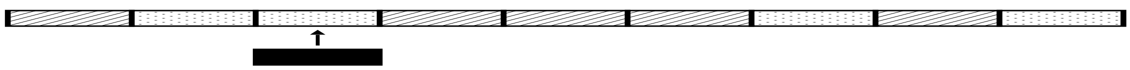
4. Evolve them with

   – *Selection*

   – *Crossover*

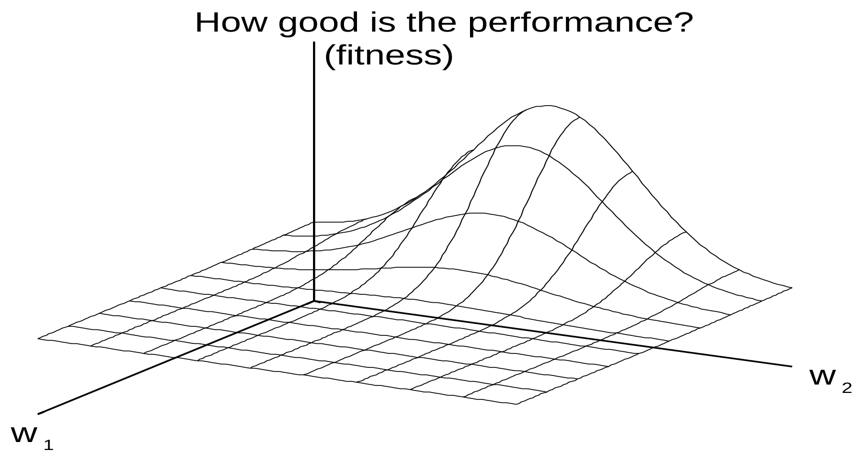


   – *Mutation*



5. Better Solutions from *generation* to *generation.*

□ A Conceptual Illustration of Fitness Landscape



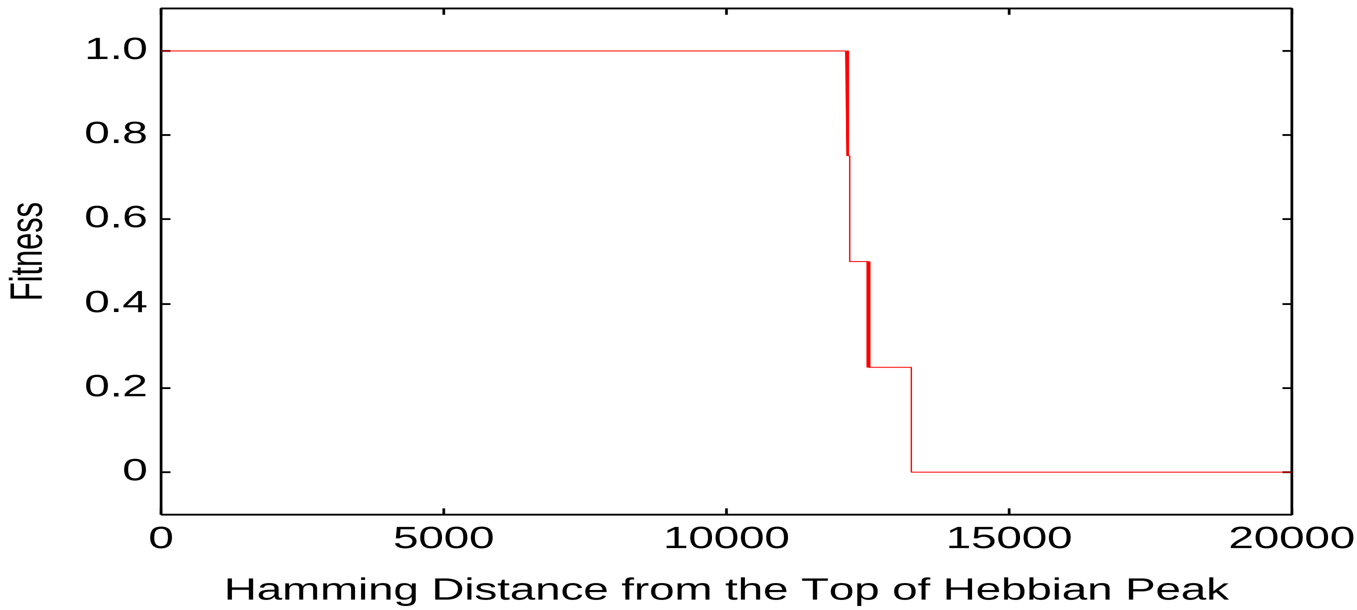
□ One specific solution were already known — Hebbian peak.

"When fire then wire" principle

$$\Downarrow$$

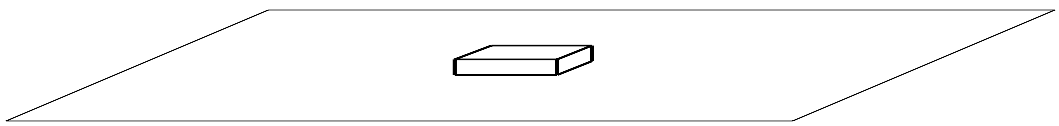
$$w_{ij} = sgn(R_i - M/2) \cdot sgn(R_j - M/2)$$

- **A Downhill Walk by Flipping zero to one:**

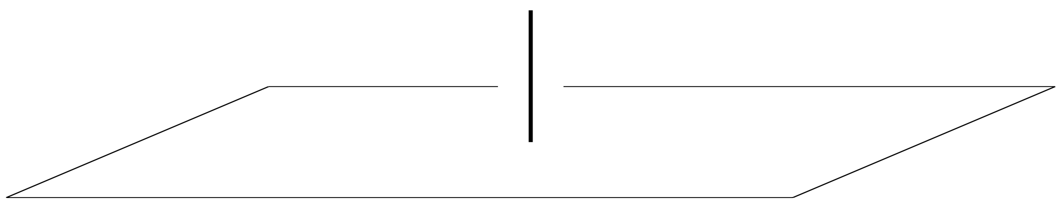


□ Search for an island

(a) A tiny island in a huge lake



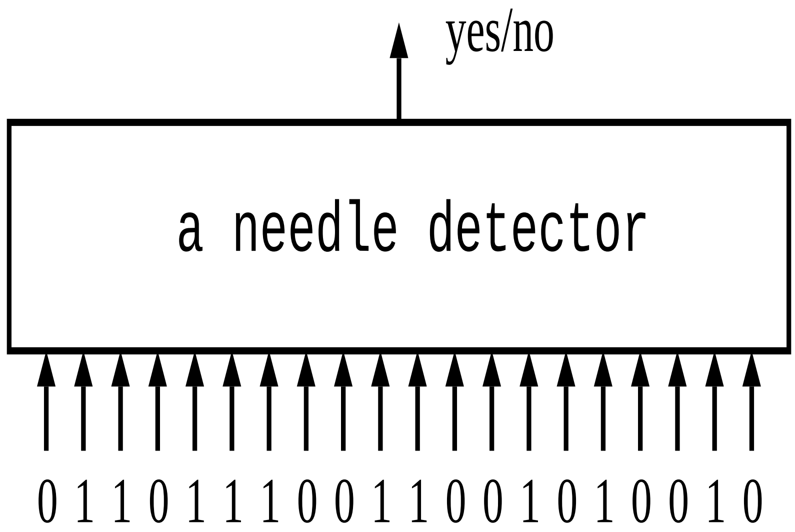
(b) A needle in a haystack



□ Hinton & Nowlan's Search for A-needle-in-haystack

- A-needle ⇒ Only one configuration of 20 bits of binary string.
- Haystack ⇒ $2^{20} - 1$ search points.
  - · say, (11111111110000000000) is assigned fitness one, while others are fitness zero.

□ Black box to detect a needle

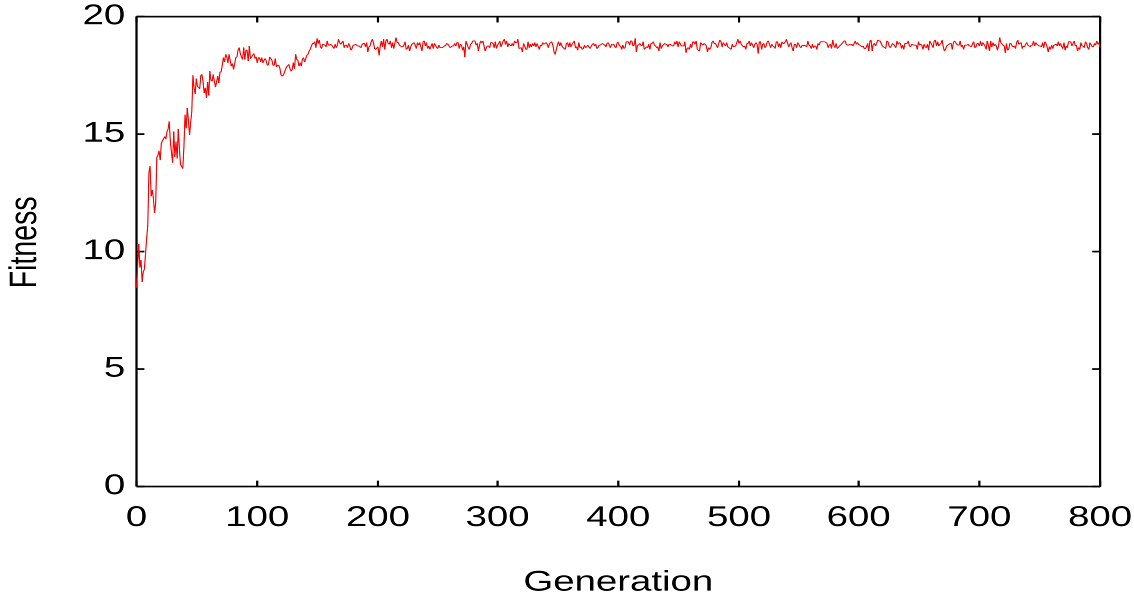


(cont'd)

- **lifetime learning of each individual (Baldwin Effect).**
  - about 25% are "1", 25% are "0", and the rest of the 50% are "?" .
  - They are evaluated with all the "?" position being assigned "1" or "0" at random ⇒ *learning*
  - Each individual repeats the learning up to 1000 times
  - If it reaches the point of fitness one at the n-th trial, then the *degree to which learning succeeded* is calculated as

$$1 + 19 \cdot (1000 - n)/1000.$$

□ Search for a needle



Test Function — A Tiny Flat Island in a Huge Lake:

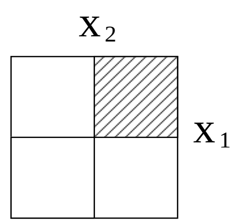
Assuming a n-dimensional hypercube all of whose coordinate x_i *are*

$$-1 \leq x_i \leq 1 \ \ (i = 1, \cdots, n),$$

find an algorithm to locate a point in the region A

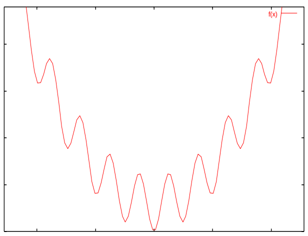
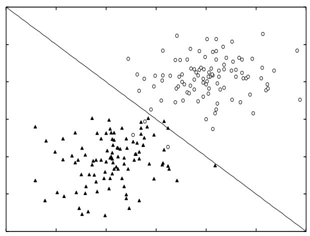
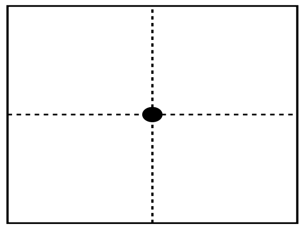
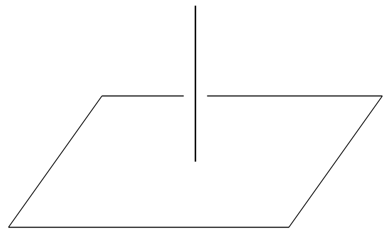
$$A : \{(x_1, x_2, \cdots, x_n) \ | \ 0 \leq x_i \leq a \ \ (i = 1, \cdots, n)\},$$

where all the points in A are assigned fitness = 1 and elsewhere 0.



□ How can AIS search for a needle, if any?

Function-optimization, pattern-classification, anomaly-detection, or ...?

□ Conclusion

- Can *Artificial Immune System* search for a needle?

- Otherwise, what method would be?

— Call for Challanges!