A Multimodal GA and a Visualization Technique Applied to the Hopfield Model of Associative Memory

Akira Imada
Anadolu University
Turkey

A visualization technique

using

Principal Component Analysis (PCA)

observes

a convergence behaviour

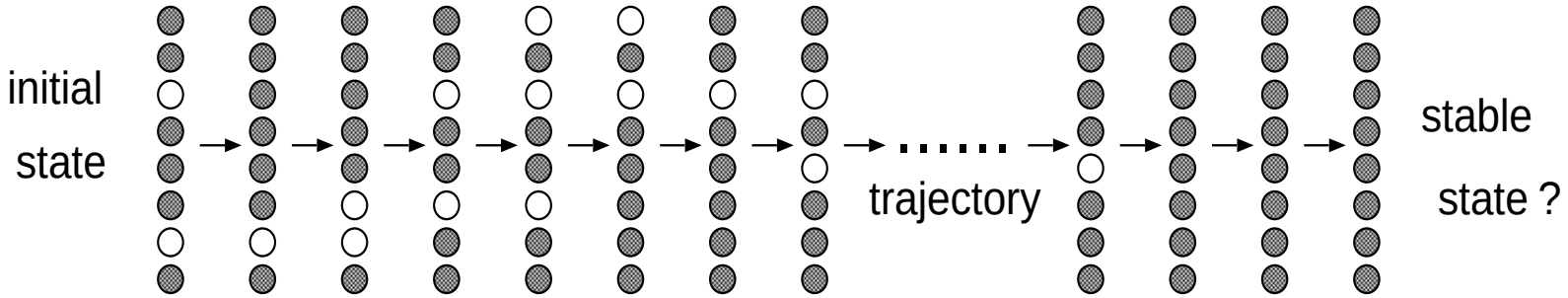
of

a multimodal Genetic Algorithm.

□ **Contents:**

- What is Associative Memory?
- How we evolve it?
- Fitness Landscape.
- Results and Discussion.
- Conclusion.

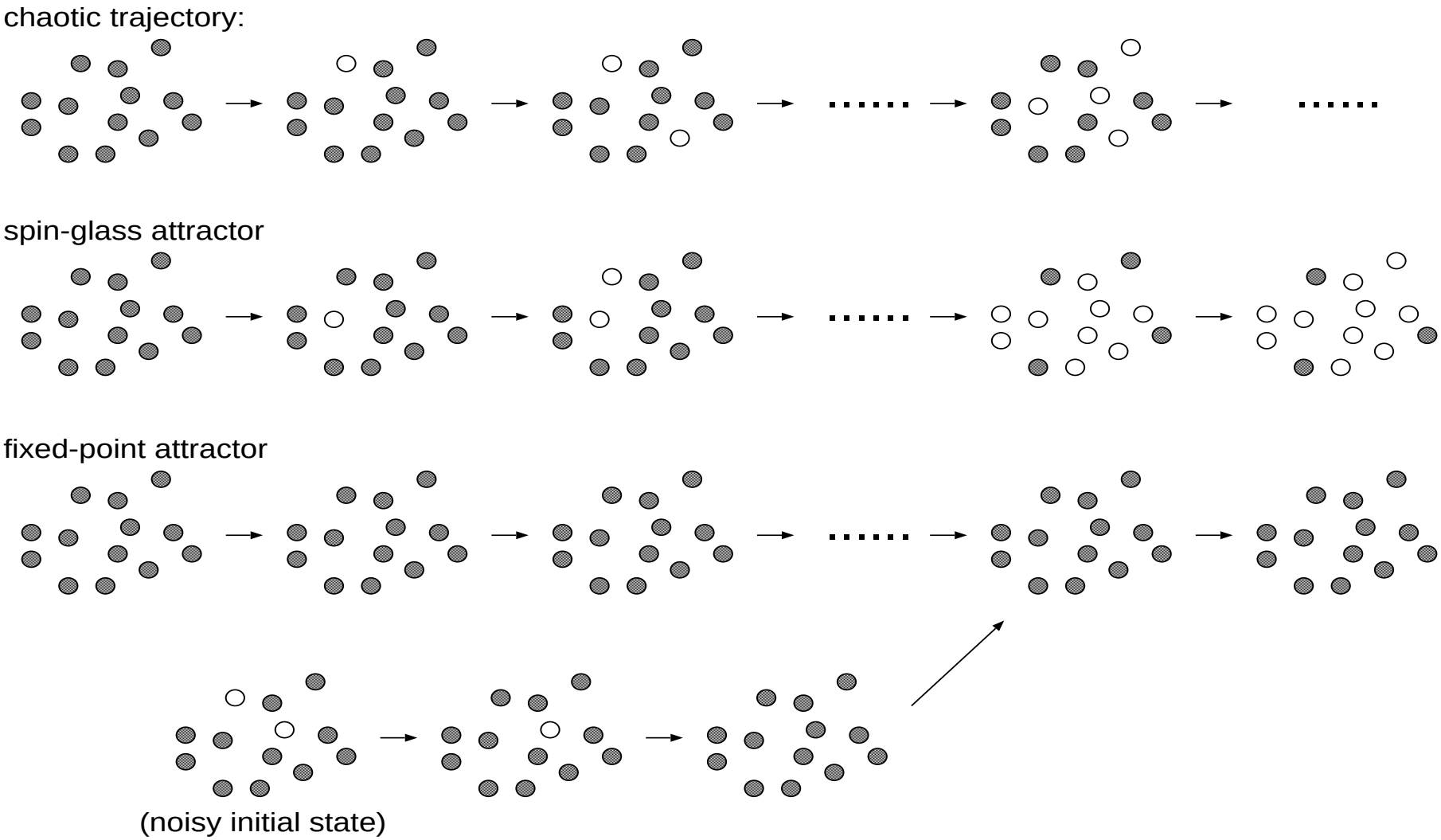
associative memory = dynamical system



converge

initial state
$u+\delta u$

equilibrium state
u

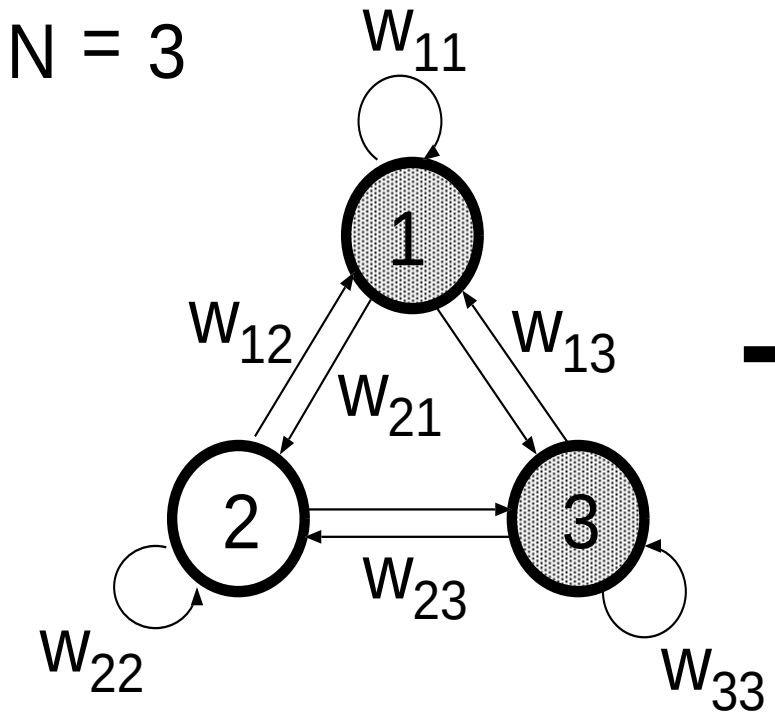

chaotic trajectory:
spin-glass attractor
fixed-point attractor
(noisy initial state)

The Associative Memory can be realized with:

- **fully-connected neural network model**
  (Hopfield model)

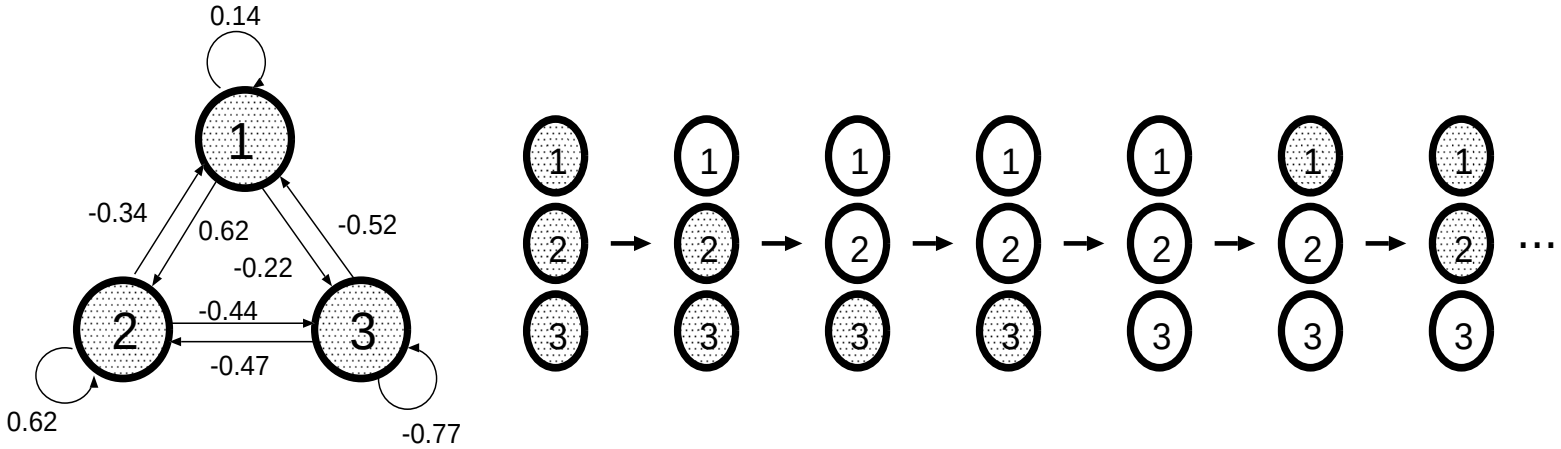
- **immune system model**

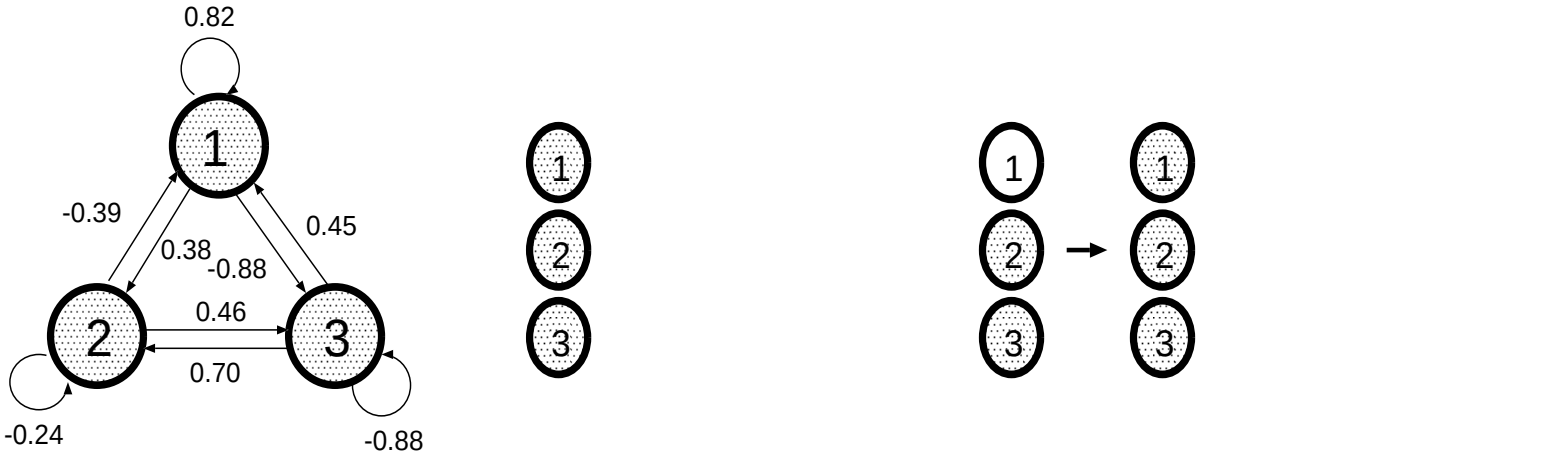
- **etc**



weight matrix

$$\begin{pmatrix} w_{11} & w_{12} & w_{13} \\ w_{21} & w_{22} & w_{23} \\ w_{31} & w_{32} & w_{33} \end{pmatrix}$$


0.14
1
-0.34
0.62
-0.52
-0.22
2
-0.44
3
-0.47
0.62
-0.77
1
2
3
1
2
3
1
2
3
1
2
3
1
2
3
1
2
3
1
2
3
...


0.82
1
-0.39
0.38
0.45
-0.88
2
0.46
3
0.70
-0.24
-0.88
1
2
3
1
2
3
1
2
3

□ **State Transition**

$$s_i(t+1) = sgn\left(\sum_j^N w_{ij} s_j(t)\right)$$

Example:

$$\begin{aligned} s_1(1) &= sgn\,(\,w_{11}\cdot s_1(0) + w_{12}\cdot s_2(0) + w_{13}\cdot s_3(0)\,) \\ &= sgn\,(\,0.14\cdot 1 + (-0.34)\cdot 1 + (-0.52)\cdot 1\,) \\ &= -1 \end{aligned}$$

□ Hebbian Weights

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

$$\begin{pmatrix} 0 & 0.11 & 0.11 \\ 0.11 & 0 & 0.11 \\ 0.11 & 0.11 & 0 \end{pmatrix}$$

- Thus we have many solutions in weight space.

$\Downarrow$

- A large network has an infinitely large number of these solutions.

$\Downarrow$

- How we can obtain them exhaustively?

$\Downarrow$

- Evolutionary Computations.

To apply an EC to a problem:

$\Downarrow$

a number of **candidate solutions** are produced (mostly) at random,

$\Downarrow$

representing them as strings (**chromosomes**).

$\Downarrow$

population

$$\begin{pmatrix} 0.14 & -0.34 & -0.52 \\ 0.62 & 0.62 & -0.47 \\ -0.22 & -0.04 & 0.77 \end{pmatrix} = (0.14, -0.34, -0.52, 0.62, 0.62, -0.47, -0.22, -0.04, 0.77)$$

$$\begin{pmatrix} 0.82 & -0.39 & -0.45 \\ 0.38 & -0.24 & 0.70 \\ -0.58 & 0.46 & -0.88 \end{pmatrix} = (0.82, -0.39, -0.45, 0.38, -0.24, 0.70, -0.58, 0.46, -0.88)$$

$$\begin{matrix} \cdots \\ \cdots \\ \cdots \end{matrix}$$

$$\begin{pmatrix} -0.62 & 0.28 & -0.61 \\ 0.92 & -0.24 & 0.06 \\ -0.11 & -0.85 & 0.49 \end{pmatrix} = (-0.62, 0.28, -0.61, 0.92, -0.24, 0.06, -0.11, -0.85, 0.49)$$

□ Fitness Evaluation:

- The degree to how a weight configuration works as an associative memory.

$$f = \max\{ \frac{1}{t_0} \sum_{t=1}^{t_0} m^{\nu}(t) \mid \mu = 1, 2, \cdots, p\}.$$

- At the start:
  - low fitness,
  - different from each other.

□ **Evolution proceeds by:**

- **Selection**
- **recombination**
- **mutation**

(-0.28, -0.51, 0.29, 0.36, -0.82, -0.35, -0.63, 0.13, 0.77)

(0.93, 0.64, -0.27, -0.51, -0.62, 0.49, 0.51, -0.22, -0.74)

(-0.71, -0.35, -0.82, 0.41, 0.52, -0.89, 0.66, -0.44, -0.36)

.....

(-0.42, -0.83, 0.38, -0.44, 0.62, 0.51, -0.38, -0.95, 0.48)

(0.93, 0.64, -0.27, -0.51, -0.62, 0.49, 0.51, -0.22, -0.74)

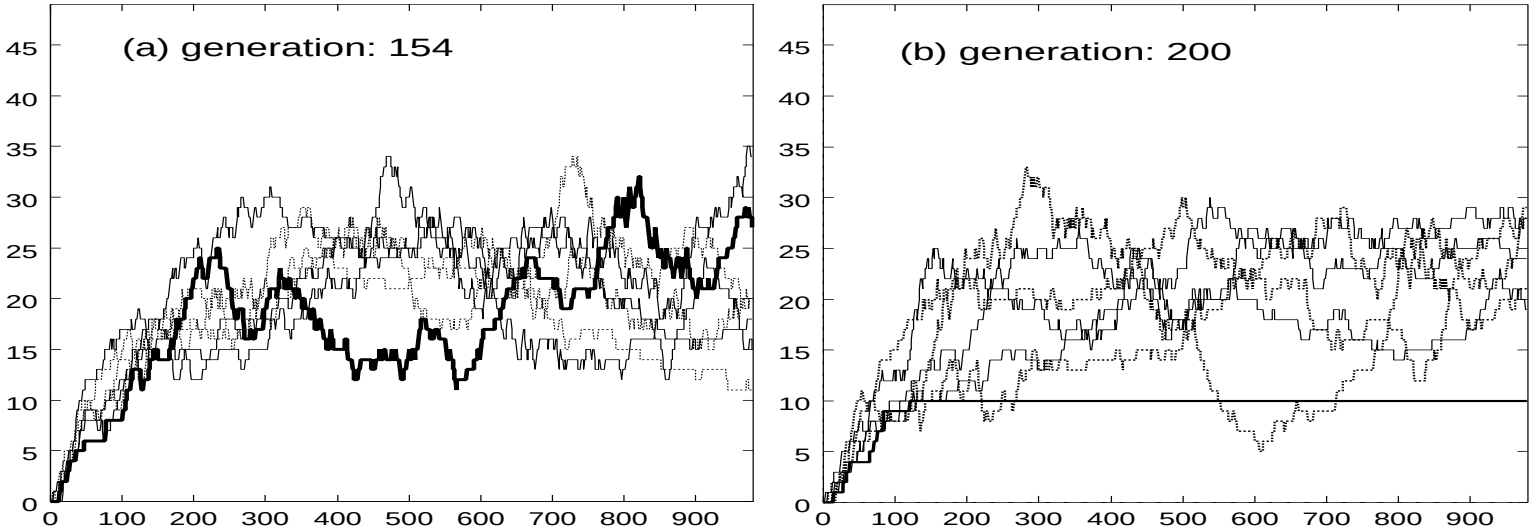
(-0.71, -0.35, -0.82, 0.41, 0.52, -0.89, 0.66, -0.44, -0.36)

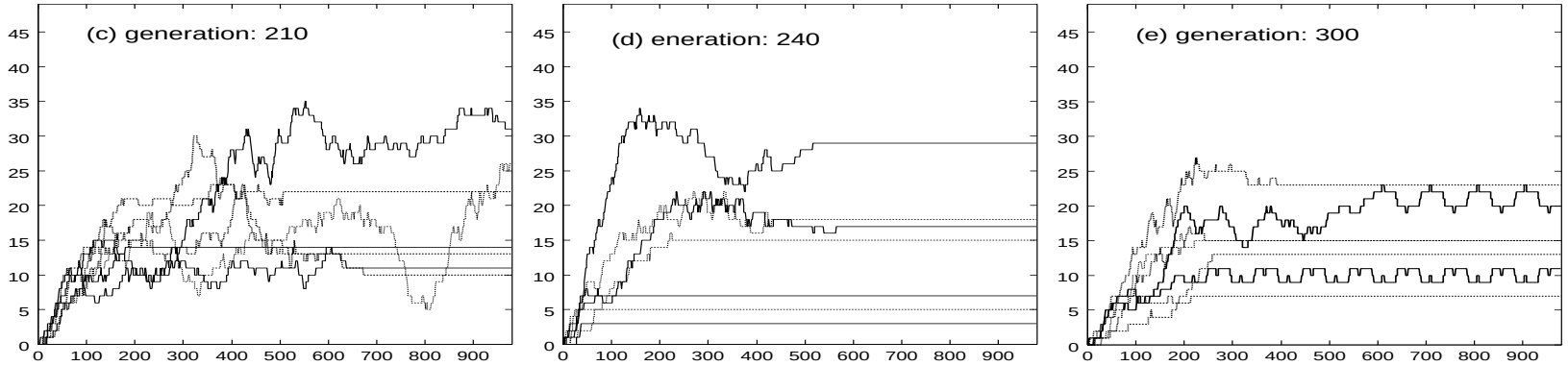
(0.93, -0.35, -0.27, 0.41, 0.52, 0.49, 0.66, -0.44, -0.74)

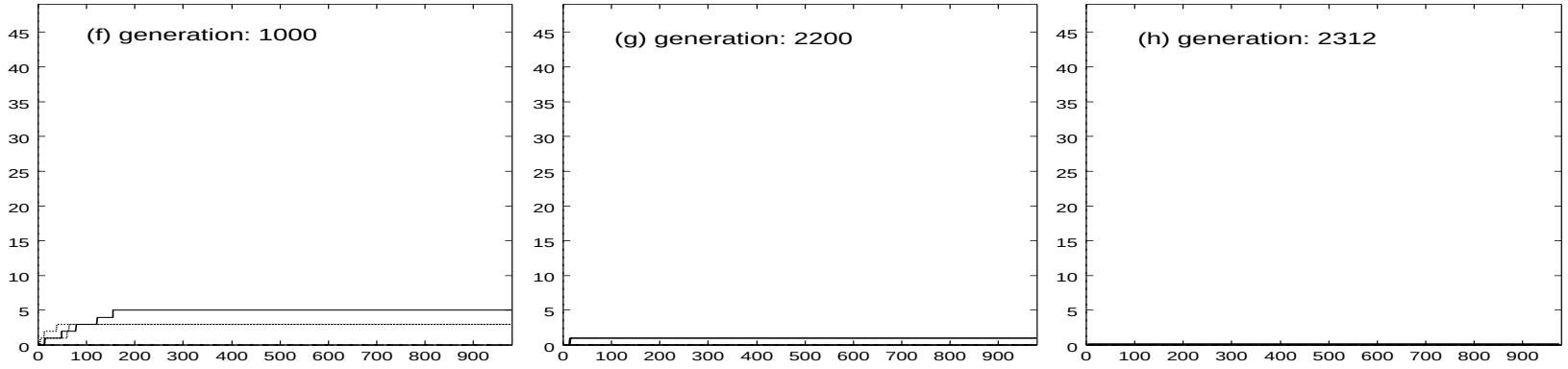
(0.93, -0.35, -0.27, 0.41, 0.54, 0.49, 0.66, -0.44, -0.74)

□ How to observe an evolution?

(1) State trajectories.

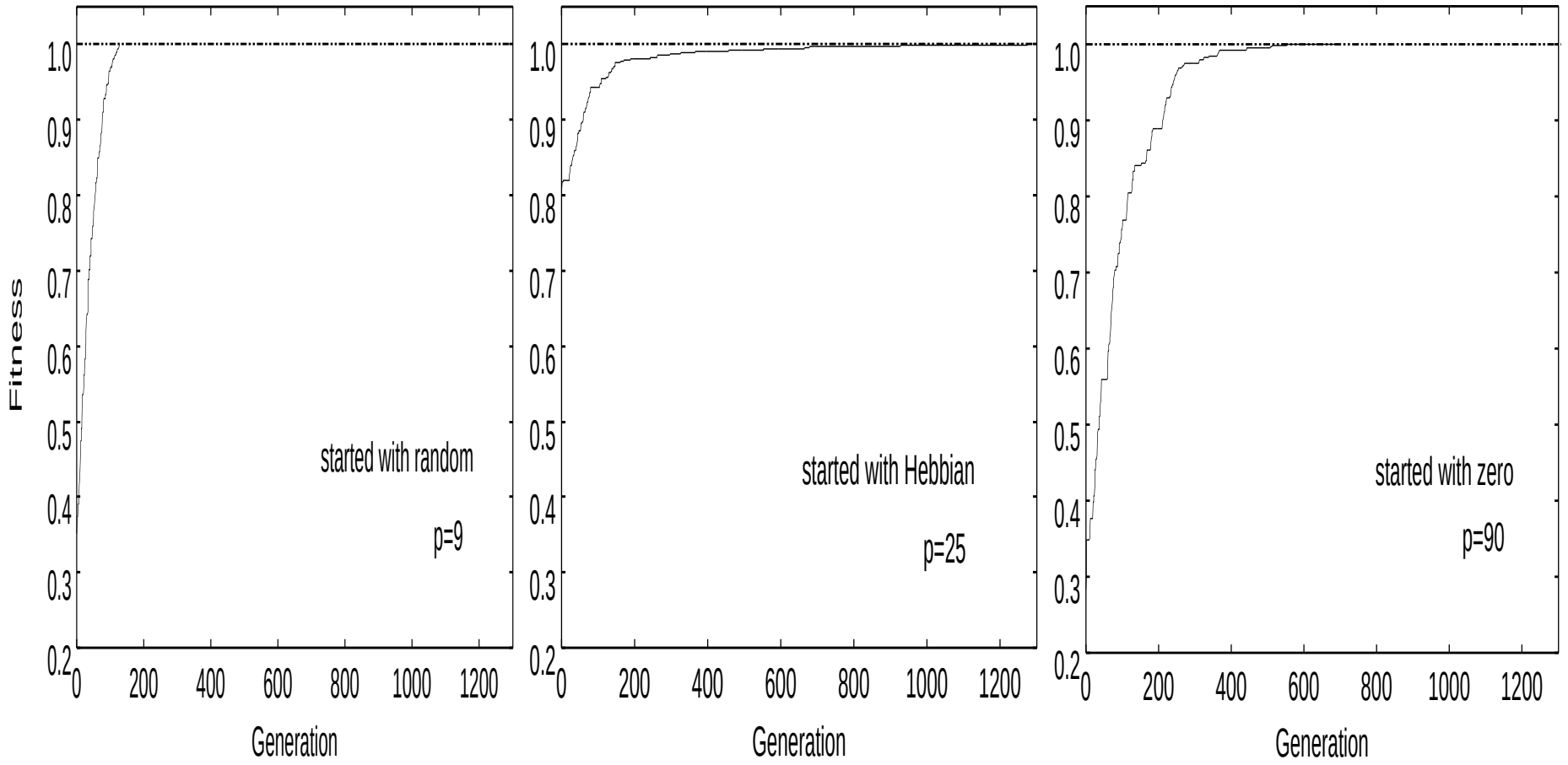

(a) generation: 154
(b) generation: 200
0
5
10
15
20
25
30
35
40
45
0
100
200
300
400
500
600
700
800
900


(c) generation: 210
(d) eneration: 240
(e) generation: 300
0
5
10
15
20
25
30
35
40
45
0
100
200
300
400
500
600
700
800
900


(f) generation: 1000
(g) generation: 2200
(h) generation: 2312
0
5
10
15
20
25
30
35
40
45
0
100
200
300
400
500
600
700
800
900

□ **How to observe an evolution?**

(2) Fitness of the best individual in the population


Fitness
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0
200
400
600
800
1000
1200
Generation
started with random
p=9
started with Hebbian
p=25
started with zero
p=90

Fitness Landscapes on Weight Space

- all the possible configurations of w_{ij} of a network with N

$\Downarrow$

- an N^2-dimensional Euclidean space

$\Downarrow$

- a specific configuration is represented as a point in the space

$\Downarrow$

- "fitness landscape" can be defined on the space

$\Downarrow$

- where we can imagine the optima as peaks

□ **The preliminary step:**

- Many different solutions by EC's.

□ **The next step:**

- What does the Fitness Landscape look like?
  - How is its geometry?
- Exploration of Weight Space.
  - The number and distribution of weight solutions?

□ Weight Space and String Space:

- weight space
  - a given set of patterns to be memorized: fixed
  - N^2-dimensional; real-valued genes

- string space
  - a configuration of weights: fixed
  - N-dimensional; binary genes

□ **The dimensionality of weight space:**

- N-neurons $\Rightarrow$ N^2-dimensional.
- 49-neurons $\Rightarrow$ $49^2 = 2401$-dimensional.

A Multimodal GA

explores

string space

searching for

patterns embedded by the Hebb rule

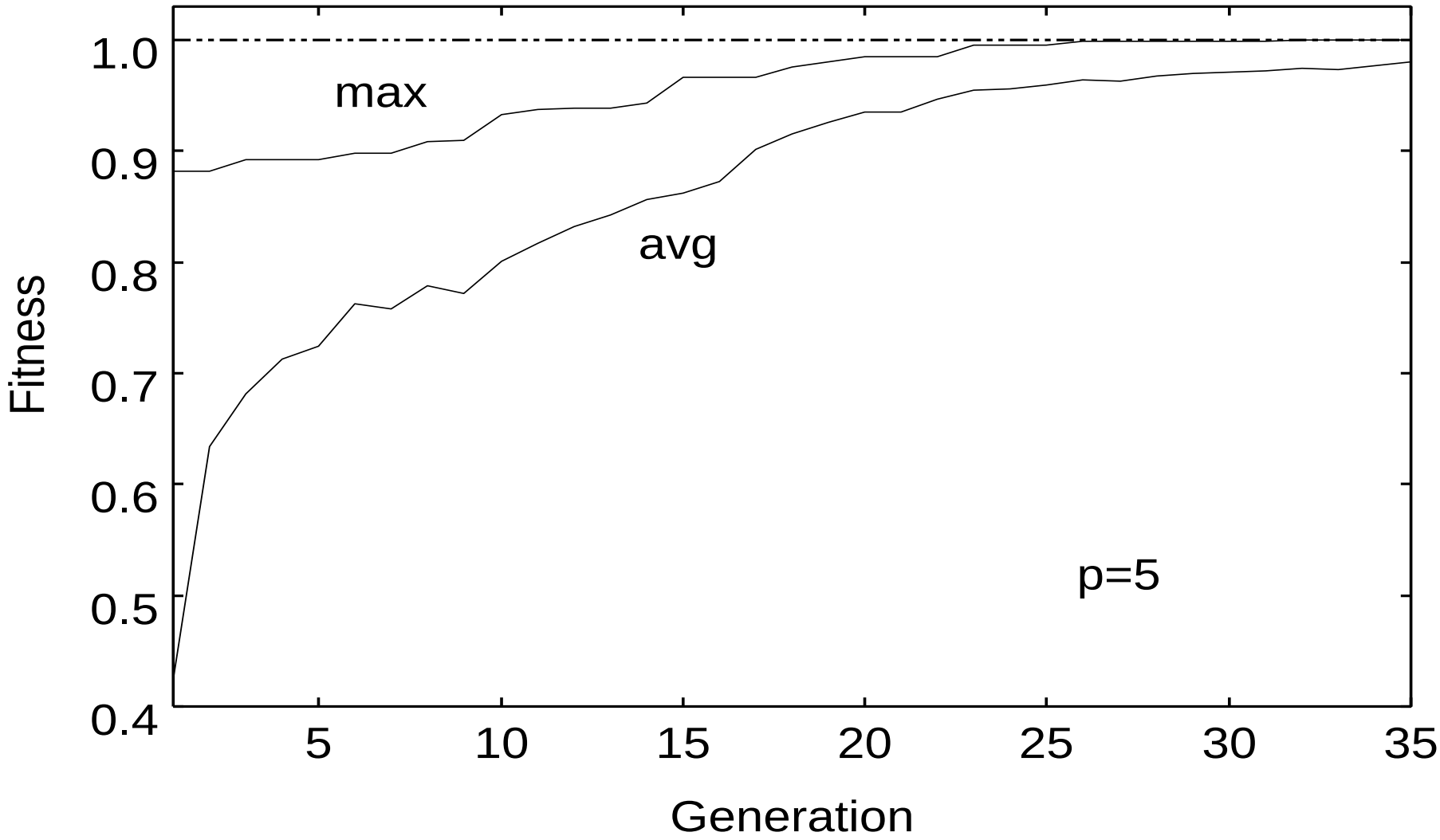
starting at

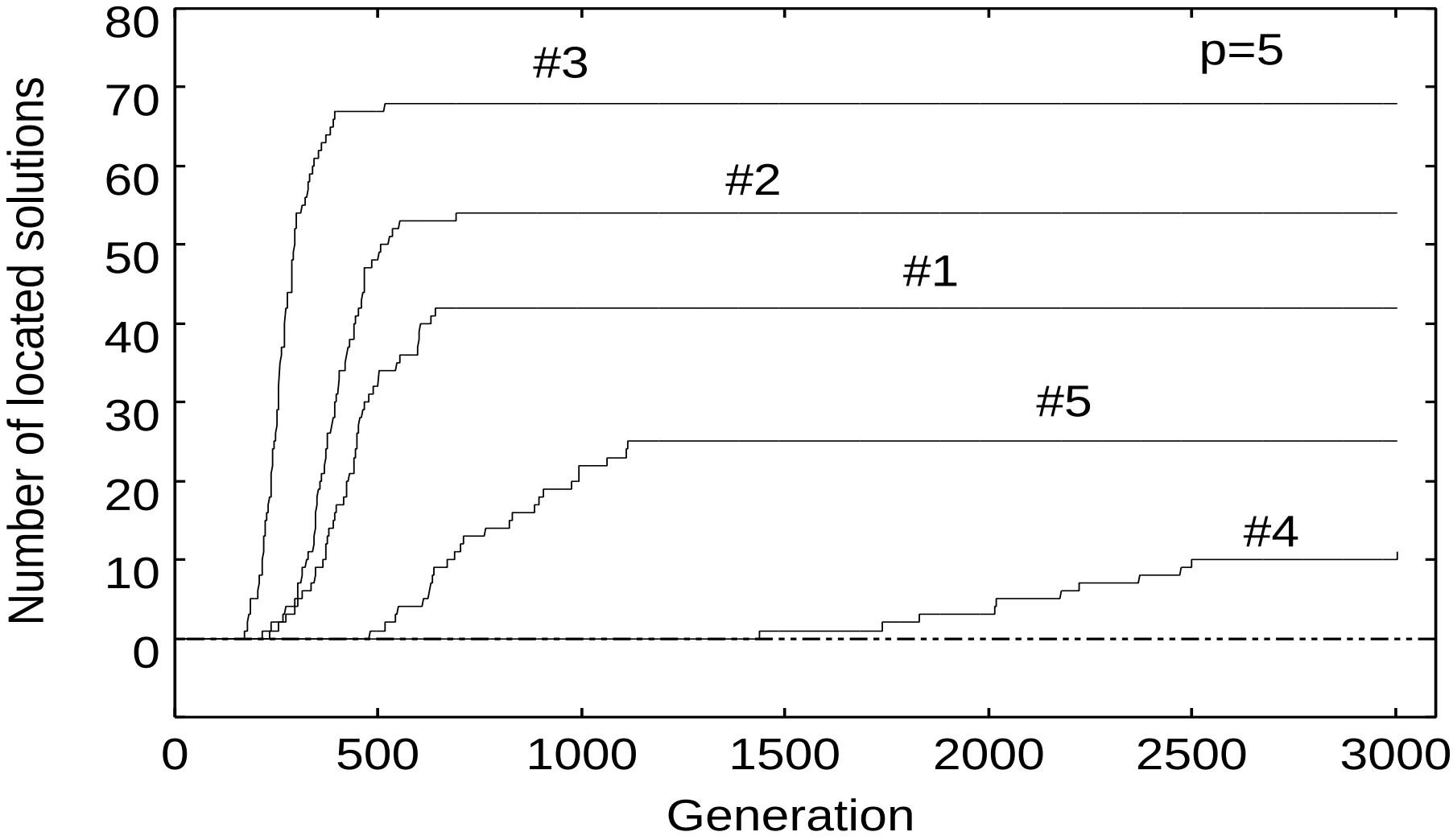
random points.

□ **A multimodal Genetic Algorithm**

⇓

locates multiple solutions simultaneously at a run.


1.0
0.9
0.8
0.7
0.6
0.5
0.4
Fitness
max
avg
p=5
5
10
15
20
25
30
35
Generation


Number of located solutions
80
70
60
50
40
30
20
10
0
0
500
1000
1500
2000
2500
3000
Generation
#3
p=5
#2
#1
#5
#4

Deterministic Crowding (Mahfoud, 1992)

(1) Choose two parents, p_1 and p_2, at random, without repetition.

(2) Produce two children, c_1 and c_2, with uniform crossover.

(3) Mutate the children by flipping bit chosen at random with probability p_m, yielding c_1' and c_2'.

(4) Replace parent with child as follows:

- IF $d(p_1, c_1') + d(p_2, c_2') > d(p_1, c_2') + d(p_2, c_1')$
  - * IF $f(c_1') > f(p_1)$ THEN replace p_1 with c_1'
  - * IF $f(c_2') > f(p_2)$ THEN replace p_2 with c_2'
- ELSE
  - * IF $f(c_2') > f(p_1)$ THEN replace p_1 with c_2'
  - * IF $f(c_1') > f(p_2)$ THEN replace p_2 with c_1'

□ **Techniques of observing high dimensional space:**

⇓

visualization

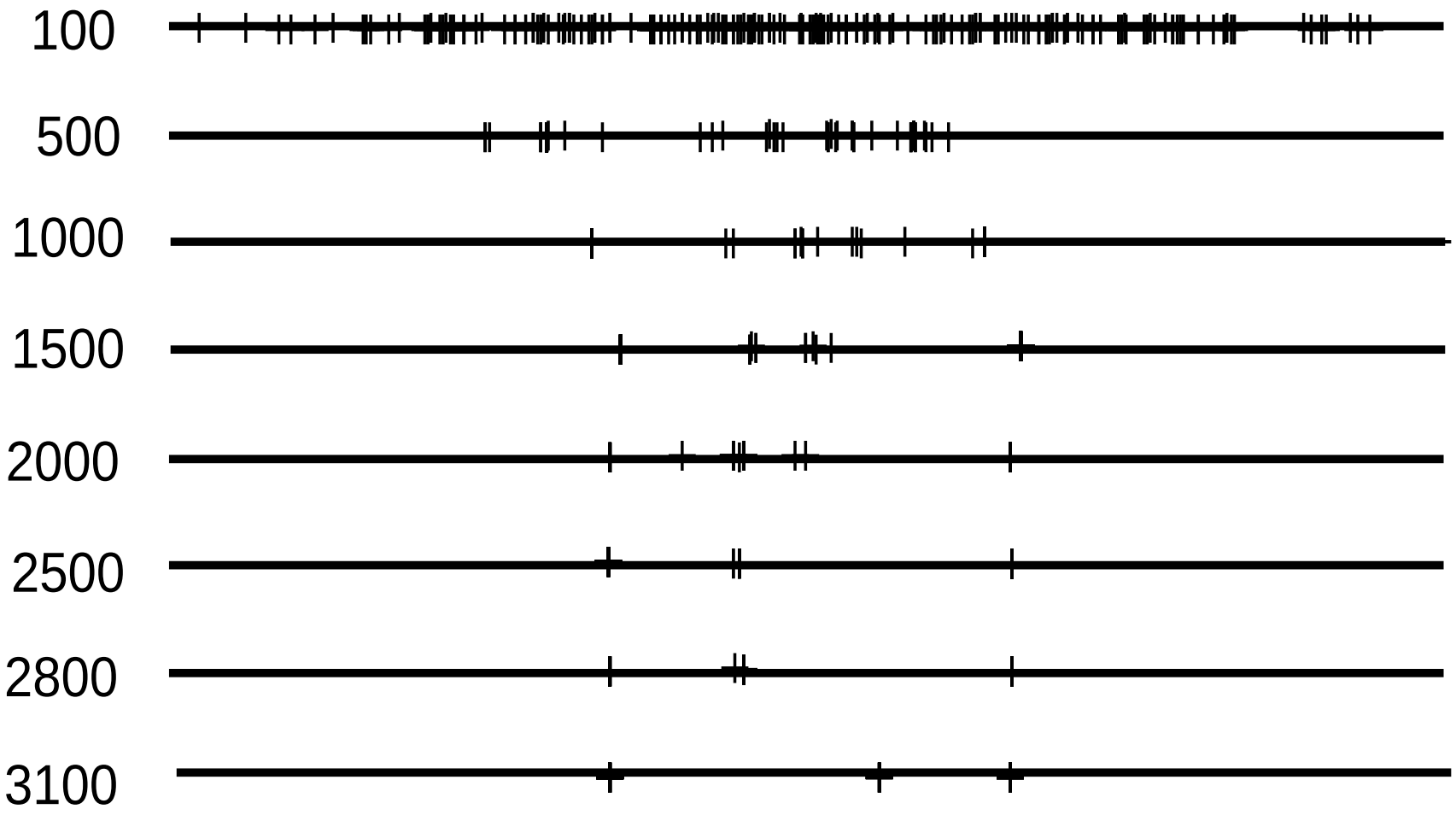
by

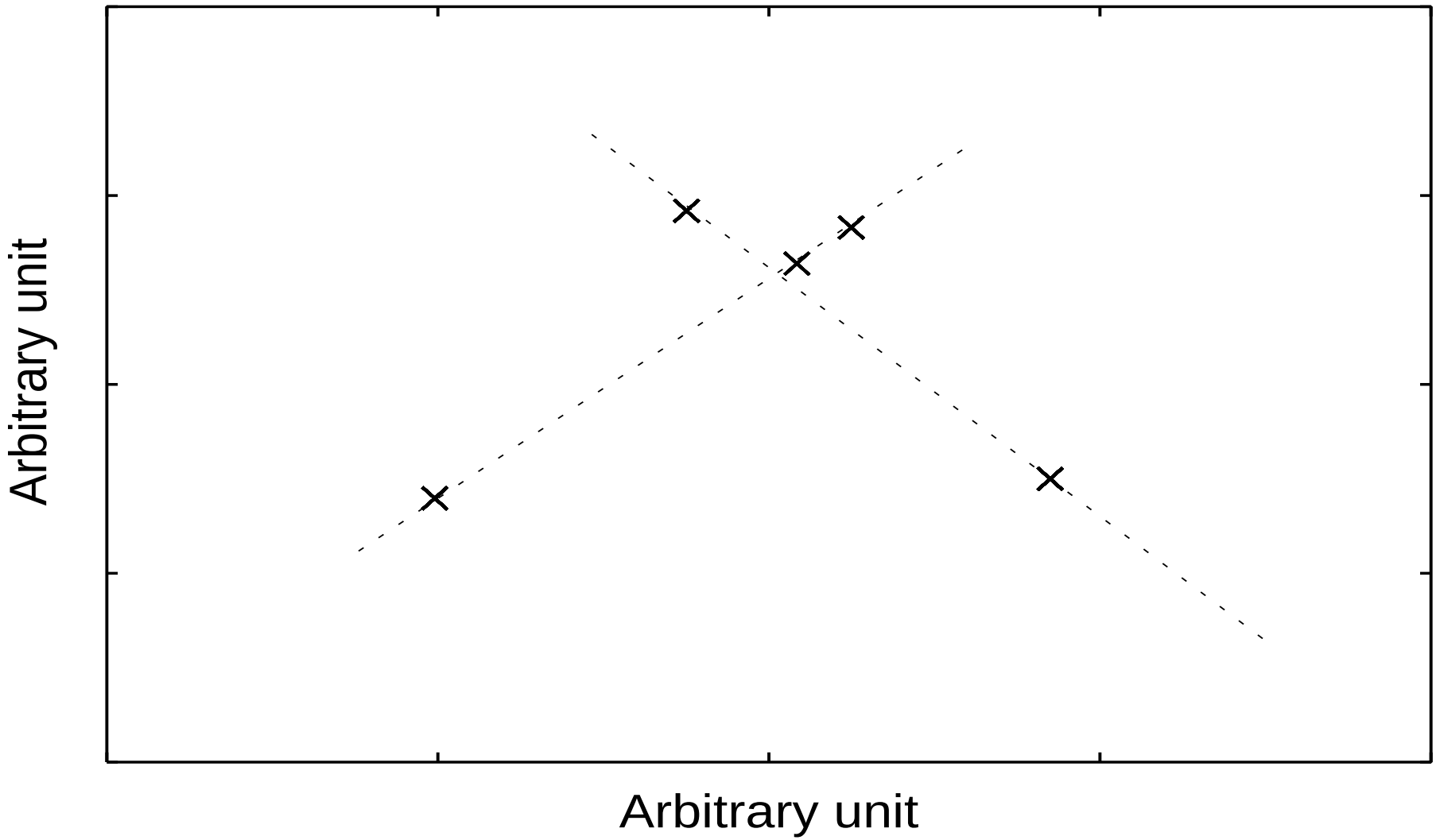
dimension reduction

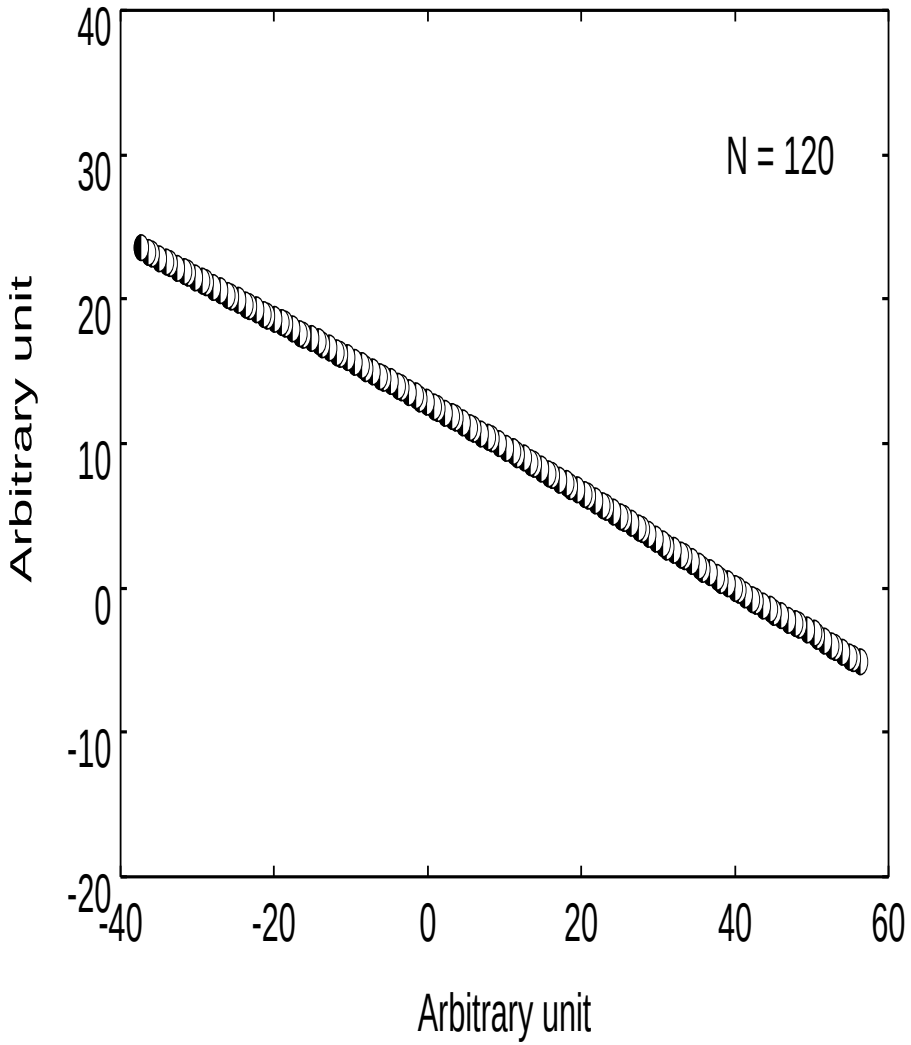
⇓

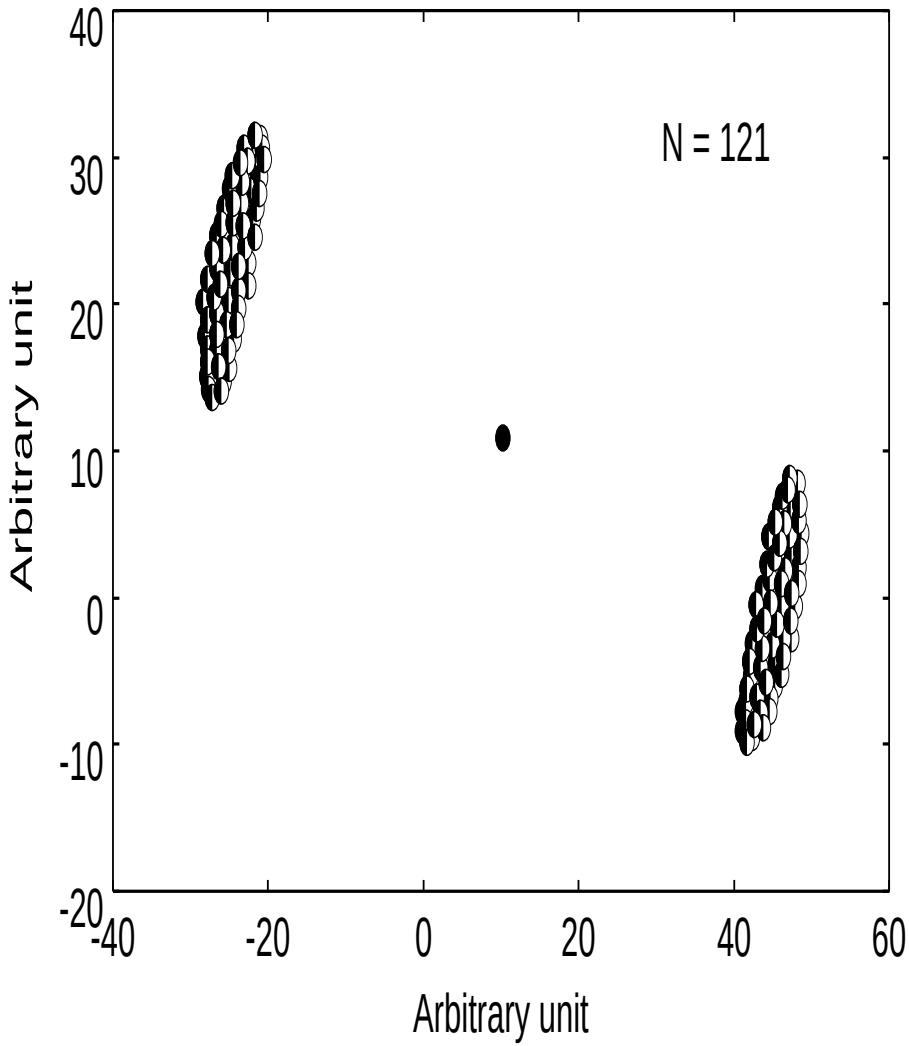
- Principal Component Analysis (PCA)
- Genetic Algorithm (GA)

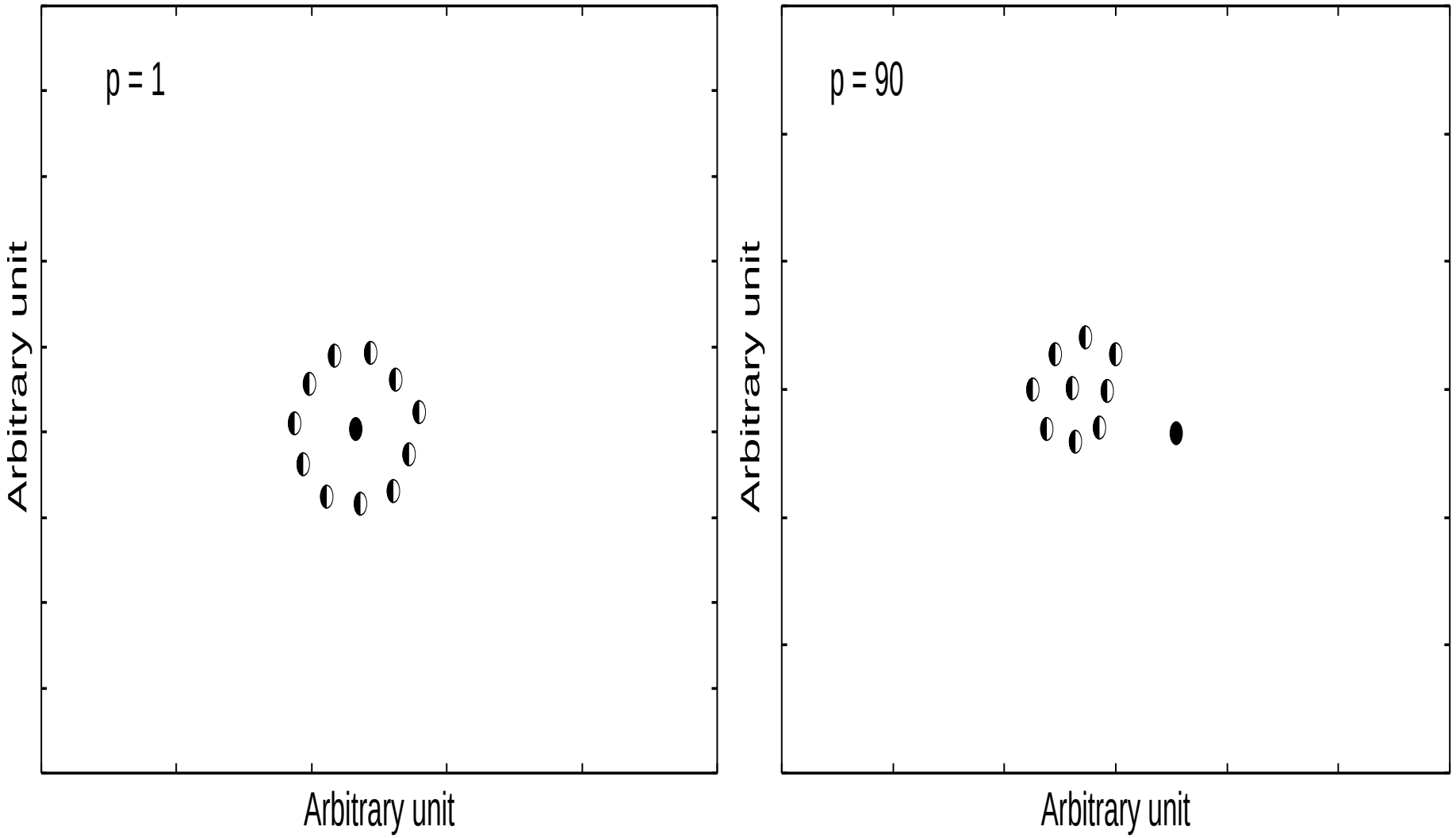
at generation:




Arbitrary unit
Arbitrary unit


N = 120
Arbitrary unit
Arbitrary unit
40
30
20
10
0
-10
-20
-40
-20
0
20
40
60


N = 121
Arbitrary unit
Arbitrary unit
40
30
20
10
0
-10
-20
-40
-20
0
20
40
60


p = 1
Arbitrary unit
Arbitrary unit
p = 90
Arbitrary unit
Arbitrary unit

□ Conclusions:

- We have many different weight solutions for Associative Memory.

  The Number and distribution of these solutions?

- PCA & GA visualize weight space.
- We shed new light on the analysis.