

Bioinspiración en I.A.

Métodos y modelos de la I.A.
Inspirados en la Naturaleza

Bioinspiración en I.A.

- La aplicación del ordenador al estudio de la Naturaleza ha conducido desde sus orígenes a espectaculares simulaciones.

Bioinspiración en I.A.

Sin embargo, y desde los orígenes de la simulación, los científicos e ingenieros emprendieron otro camino: aquel en el que es la propia Naturaleza la que se convierte en fuente de inspiración, diseñándose desde entonces un número cada vez mayor de procedimientos computacionales bioinspirados de probada eficacia y utilidad.

Bioinspiración en I.A.

- La simulación de procesos naturales es un campo de investigación muy amplio en Inteligencia Artificial.
- Si ha funcionado bien en la naturaleza, ¿porqué una simulación de estos procesos no iba a proporcionar buenos resultados en un computador?

Bioinspiración en I.A.

Este área de la I.A. es conocida por varios nombres que dependen un tanto de la orientación y el uso que se haga de la “bioinspiración”:

- **computación evolutiva,**
- **biocomputación,**
- **algoritmos bioinspirados,**
- **etc.**

Bioinspiración en I.A.

- Ejemplos
 - Algoritmos genéticos,
 - Algoritmos basados en inteligencia de enjambres,
Algoritmos basados en Colonias de Hormigas,
 - Redes Neuronales Artificiales,
 - Algoritmos basados en el Sistema inmunológico,
 - Computación molecular,
 - etc

Algoritmos genéticos

- La teoría de la evolución (que no es tal teoría, sino una serie de hechos observados/ probados), fue descrita por Charles Darwin y Alfred Russell Wallace de modo casi independiente en 1858.
- Los cambios heredables en los seres vivos y la selección son los dos hechos que provocan el cambio en la Naturaleza y la generación de nuevas especies.

Algoritmos genéticos

- Pero Darwin desconocía cual es la base de la herencia.
- Pensaba que los rasgos de un ser vivo eran como un fluido, y que los "fluidos" de los dos padres se mezclaban en la descendencia.

Algoritmos genéticos

- Fue Gregorio Mendel, a mediados del siglo XIX, quien descubrió que los caracteres se heredaban de forma discreta (factores/genes), y que se tomaban del padre y de la madre.

<http://museovirtual.csic.es/salas/mendel/m10.htm>

- En realidad, las teorías de Mendel, que trabajó solo, se olvidaron y no se volvieron a redescubrir hasta principios del siglo XX.

Algoritmos genéticos

- En 1930 el genetista inglés Robert Aylmer relacionó ambas teorías, demostrando que los genes de Mendel eran los que proporcionaban el mecanismo necesario para la evolución.
- Más o menos por la misma época, el biólogo alemán Walther Flemming describió los cromosomas.

Algoritmos genéticos

- Poco más adelante se descubrió que las células de cada especie viviente tenían un número fijo y característico de cromosomas.
- En los años 50, Watson y Crick descubrieron que la base molecular de los genes está en el ADN, ácido desoxirribonucleico.

Algoritmos genéticos

- A principios de los 60, John Holland en la Universidad de Michigan en Ann Arbor, “descubre” la teoría genética de la selección natural y concluye que la evolución era una forma de adaptación más potente que el simple aprendizaje, y tomó la decisión de aplicar estas ideas para desarrollar programas bien adaptados para un fin determinado.

Algoritmos genéticos

- En esa universidad, Holland impartía un curso titulado Teoría de sistemas adaptativos.

Dentro de este curso, fue donde se crearon las ideas que más tarde se convertirían en los algoritmos genéticos.

Algoritmos genéticos

En un proceso de evolución, existe una **población de individuos**. Los más adecuados a su entorno **se reproducen y tienen descendencia** (a veces **con mutaciones** que mejoran su idoneidad al entorno).

Los más adecuados sobreviven para la siguiente generación.

Algoritmos genéticos

- Hoy los algoritmos genéticos son métodos sistemáticos para la resolución de problemas de búsqueda y optimización que aplican a estos los mismos métodos de la evolución biológica:
 - selección basada en la población,
 - reproducción sexual y
 - mutación.

Algoritmos genéticos

Desde un punto de vista general, el objetivo de cualquier A.G. es encontrar una solución óptima para una cierta función objetivo:

$$\text{Max } F(x)$$

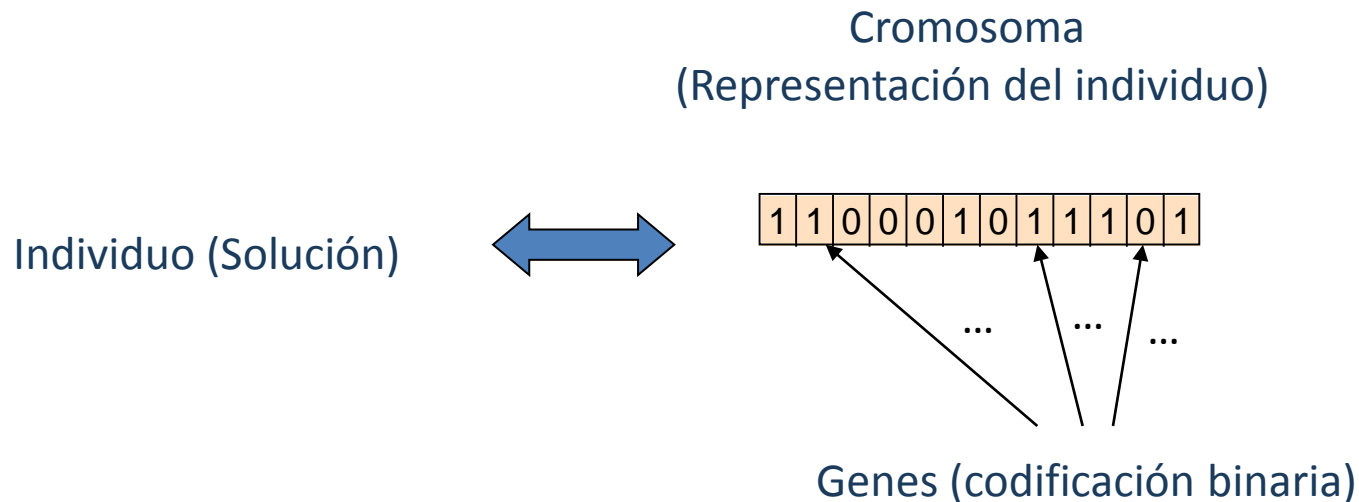
$$x \in X$$

Algoritmos genéticos

- **Cromosoma** \leftrightarrow Vector representación de una solución al problema.
- **Gen** \leftrightarrow Característica/Variable/Atributo concreto del vector de representación de una solución
- **Población** \leftrightarrow Conjunto de soluciones al problema.
- **Adecuación al entorno** \leftrightarrow Valor de función objetivo (fitness).
- **Selección natural** \leftrightarrow Operador de selección.
- **Reproducción sexual** \leftrightarrow Operador de cruce.
- **Mutación** \leftrightarrow Operador de mutación.
- **Cambio generacional** \leftrightarrow Operador de reemplazamiento.

Algoritmos genéticos

- **Ejemplo:** Cromosoma que codifica una solución a un problema. Cada característica del problema es un valor 0/1.



Algoritmos genéticos

- **Ejemplo:** Población. Conjunto de individuos (cada uno con su fitness).

Población

Conjunto de soluciones

1	1	0	0	0	1	0	1	1	1	0	1
---	---	---	---	---	---	---	---	---	---	---	---

 Fitness = 6

0	1	1	1	1	0	0	1	0	1	1	0
---	---	---	---	---	---	---	---	---	---	---	---

 Fitness = 4

1	0	1	1	0	0	1	1	1	1	0	0
---	---	---	---	---	---	---	---	---	---	---	---

 Fitness = 7

0	1	0	1	0	1	0	1	1	1	1	1
---	---	---	---	---	---	---	---	---	---	---	---

 Fitness = 1

0	0	0	0	0	1	1	0	0	1	0	0
---	---	---	---	---	---	---	---	---	---	---	---

 Fitness = 9

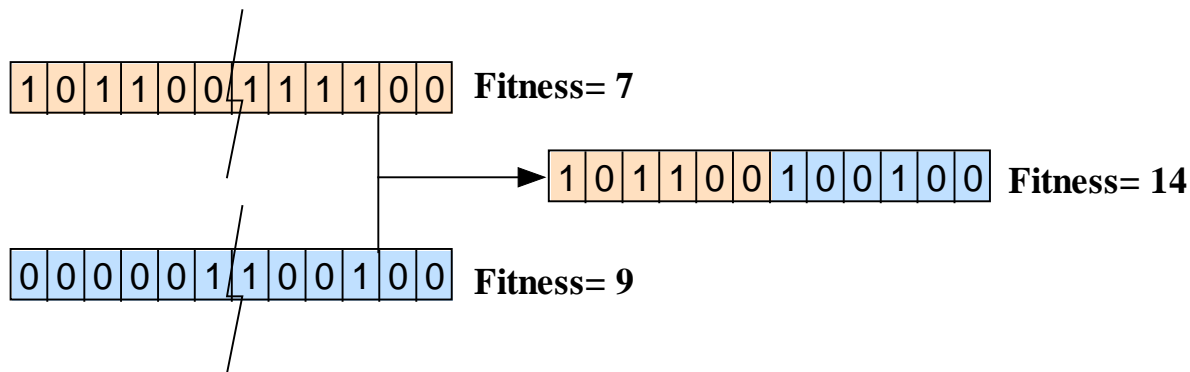
...

0	0	1	1	1	1	0	1	0	1	1	0
---	---	---	---	---	---	---	---	---	---	---	---

 Fitness = 8

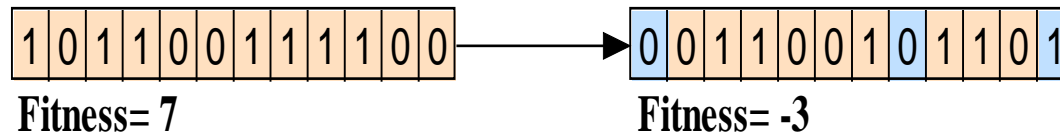
Algoritmos genéticos

- **Ejemplo:** Cruce. Combinación de soluciones de la población para generar descendientes.



Algoritmos genéticos

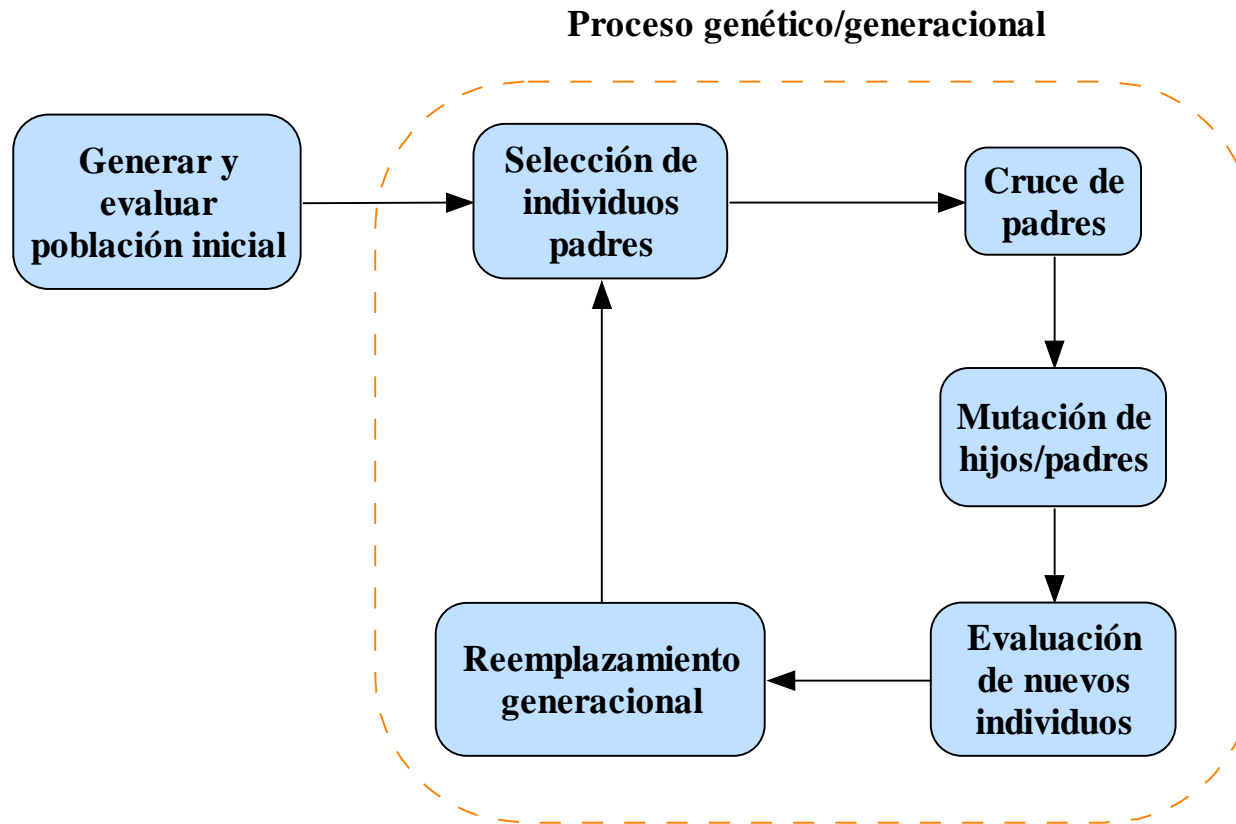
- **Ejemplo:** Mutación. Uno o más genes de un individuo pueden mutar para generar una nueva solución.



- En la población, hay una probabilidad dada a priori de que un individuo pueda mutar. A su vez, cuando un individuo muta, existe otra probabilidad de que cada gen mute o no.

Algoritmos genéticos

- Proceso de un algoritmo genético:



Ejemplo A.G.

- Vamos a partir de una función $f(x)$ muy sencilla: $f(x) = x^2$
- Se busca el valor de x que hace máximo $f(x)$, restringiendo a la variable x a tomar valores comprendidos entre 0 y 31.
Obviamente el máximo se tiene para $x = 31$, donde f vale 961.
- A x sólo le vamos a permitir tomar valores enteros, es decir: 0, 1, 2, 3, ..., 30, 31.

No necesitamos saber algoritmos genéticos para resolver este problema, pero su sencillez hace que el algoritmo sea más fácil de comprender.

Ejemplo A.G.

- Lo primero es encontrar una manera de codificar en forma de “cromosomas” las posibles soluciones (posibles valores de x).
- Con codificación binaria cada valor entre 0 y 31 se puede codificar con 5 bits (cromosoma).
- El cromosoma que representa al valor 11 será (0, 1, 0, 1, 1).

Ejemplo A.G.

- Una vez que tenemos codificada la solución, debemos escoger un tamaño de población.
-
- Para este ejemplo trabajaremos con poblaciones de 6 individuos.
- Debemos partir de una población inicial. Una manera de generarla es aleatoriamente.

Ejemplo A.G.

- Se coge una moneda y se lanza, si sale cara anotamos un 0 y en caso contrario un 1.
- Con 30 lanzamientos “troceados” en 6 cromosomas de 5 genes (bits) tenemos la población inicial.

Ejemplo A.G.

- El siguiente paso es hacer competir a los individuos entre sí. Este proceso se conoce como selección.
- Una forma de selección es mediante un **torneo**:
 1. A cada individuo de la población se le asigna una pareja ambos se valoran con $f(x)$ y se comparan (torneo).
 2. El mejor es duplicado y el peor se desecha. Con esto se obtiene una nueva población de 6 individuos
- Existen muchas variantes de este proceso de selección.

- La tabla siguiente resume el proceso.
- Las columnas 1, 2, 3, 4 identifican cada cromosoma (solución).
- La columna (5) indica la pareja asignada, aleatoriamente en este caso, a cada individuo.
- La asignación aleatoria o por muestreo es lo usual

Ejemplo A.G.

Población Inicial/Selección				
(1)	(2)	(3)	(4)	(5)
1	(0,1,1,0,0)	12	144	6
2	(1,0,0,1,0)	18	324	3
3	(0,1,1,1,1)	15	225	2
4	(1,1,0,0,0)	24	576	5
5	(1,1,0,1,0)	26	676	4
6	(0,0,0,0,1)	1	1	1

Ejemplo A.G.

- El mejor individuo es el 5 con $f = 676$.
- La media de f es para esta población es $f_{med} = 324.3$.

Ejemplo A.G.

Después de realizar el proceso de selección, la población que tenemos es la mostrada en la columna (2) de la tabla 2.

Observe, por ejemplo, que en el torneo entre el individuo 1 y el 6 de la población inicial, el primero de ellos ha recibido dos copias, mientras que el segundo cae en el olvido.

Ejemplo A.G.

- Tras realizar la selección, se realiza el cruce.
- Una manera de hacerlo es mediante el cruce 1X: se forman parejas entre los individuos aleatoriamente de forma similar a la selección.
- Dada una pareja se establece un punto de cruce aleatorio, que no es más que un número aleatorio entre 1 y 4 (la longitud del individuo menos 1).

Ejemplo A.G.

CRUCE/Tabla 2

(1)	(2)	(3)	(4)
-----	-----	-----	-----

1	(0,1,1,0,0)	5	1
---	-------------	---	---

2	(0,1,1,0,0)	3	3
---	-------------	---	---

3	(1,0,0,1,0)	2	3
---	-------------	---	---

4	(1,0,0,1,0)	6	1
---	-------------	---	---

5	(1,1,0,1,0)	1	1
---	-------------	---	---

6	(1,1,0,1,0)	4	1
---	-------------	---	---

Ejemplo A.G.

Por ejemplo, en la pareja 2-3 el punto de cruce es 3:

- un hijo de la pareja hereda los tres primeros bits del padre y los dos últimos de la madre,
- el otro hijo hereda los tres primeros bits de la madre y los dos últimos del padre.

La población resultante se muestra en la columna (2) de la tabla 3.

Ejemplo A.G.

POBLACION TRAS EL CRUCE/Tabla 3

(1)	(2)	(3)	(4)
1	(0,1,0,1,0)	10	100
2	(1,1,1,0,0)	28	784
3	(0,1,1,1,0)	14	196
4	(1,0,0,0,0)	16	256
5	(1,1,0,1,0)	26	676
6	(1,0,0,1,0)	18	324

Ejemplo A.G.

Observemos que

- el valor máximo de f es 784 (para el individuo 2), mientras que antes de la selección y el cruce era de 676.
- f_{med} ha subido de 324.3 a 389.3.

¿Qué quiere decir esto?

Los individuos después de la selección y el cruce son mejores que antes de estas transformaciones.

Ejemplo A.G.

- El siguiente paso es volver a realizar la selección y el cruce tomando como población inicial la de la tabla 3.
- Esta manera de proceder se repite tantas veces como número de iteraciones se desee.

Ejemplo A.G.

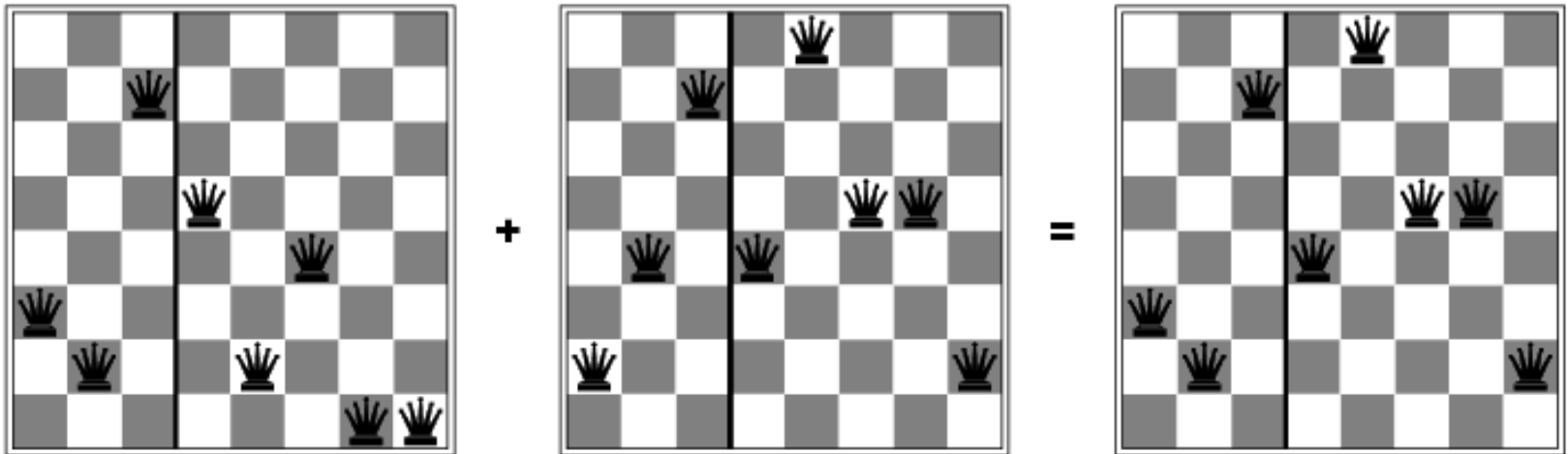
- Y ¿cuál es el óptimo?

En realidad un algoritmo genético no garantiza la obtención del óptimo pero, si está bien construido, proporcionará una solución razonablemente buena.

Ejemplo A.G.

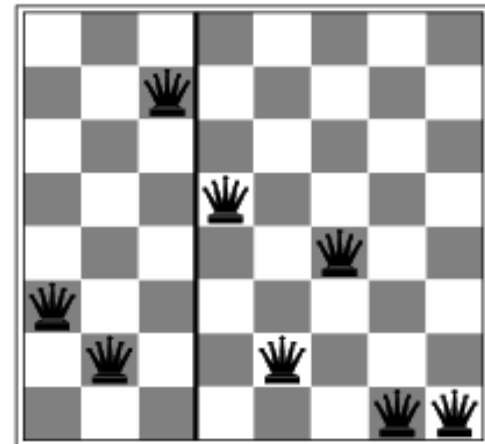
- Puede que se obtenga el óptimo, pero el algoritmo no confirma que lo sea. Así que es aconsejable quedarse con la mejor solución de la última iteración.
- También es buena idea ir guardando la mejor solución de todas las iteraciones anteriores y al final quedarse con la mejor solución de las exploradas.

Problema de las 8 reinas



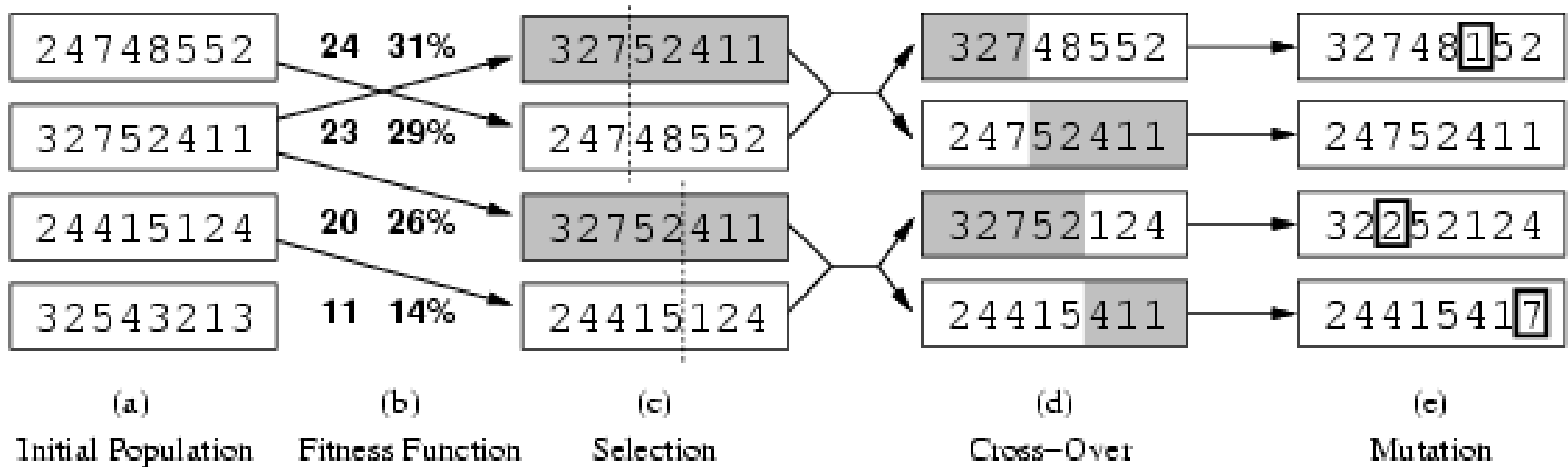
Problema de las 8 reinas

- La forma inmediata de codificarlo es mediante un vector de 8 posiciones, numeradas de 0 a 7, donde cada posición representa una columna y el valor de dicha posición la fila donde está colocada la reina.



- [5,6,1,3,6,4,7,7]
- [(101),(110),(001),(011),(110),(100),(111),(111)]

Ejemplo



Función de evaluación (8 reinas) = número de pares de reinas no atacadas

28=7+6+5+4+3+2+1 valor máximo correspondiente a una solución

Inteligencia de Enjambres

Hay muchos comportamientos de insectos sociales que son interesantes y no bien conocidos:

- Encontrar el camino (mas corto) a la comida,
- División del trabajo y trabajo cooperativo,
- Gestión de cementerios,
- Gestión de las crías,
- Etc.

Inteligencia de Enjambres

Comportamiento de los insectos sociales: muy “inteligente” y sorprendente

Causas:

- Condicionamiento genético
- Autoorganización

Autoorganización

- Conjunto de mecanismos dinámicos por los cuales un sistema presenta una estructura de alto nivel como consecuencia de la integración de componentes de bajo nivel.

Mecanismos de autoorganización

- Amplificación (positive feedback)
- Amortiguación (negative feedback)
- Fluctuaciones aleatorias
- Interacciones múltiples

Características de la autoorganización:

- Creación de estructuras espacio- temporales en un medio homogéneo
- Posible existencia de varios estados estables
- Cambios drásticos cuando se modifican los parámetros

Formas de interacción entre unidades:

- Directas (comunicación)
- Indirectas (señales en el medio ambiente: stimergia)

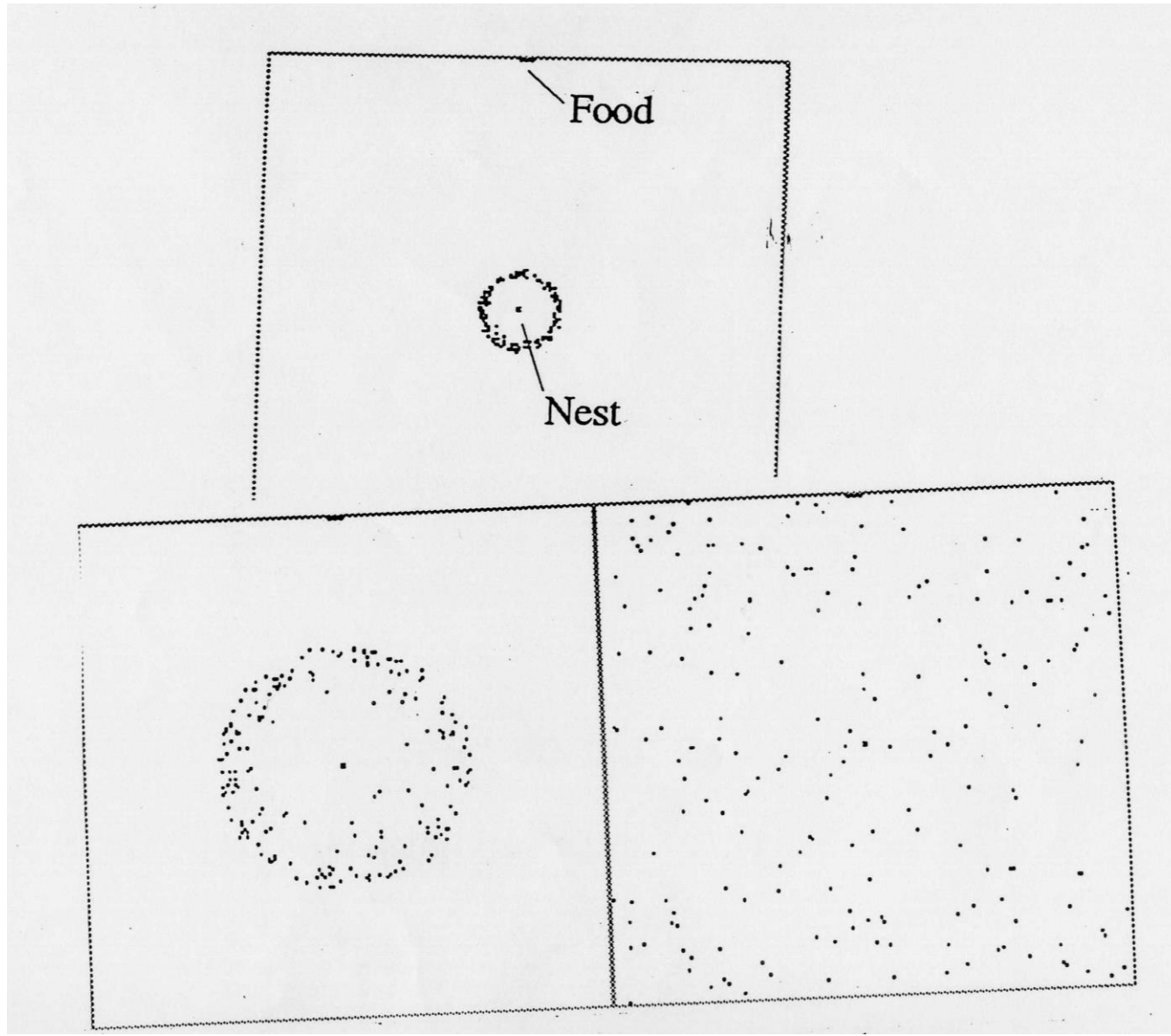
Algoritmos basados en el comportamiento de las hormigas

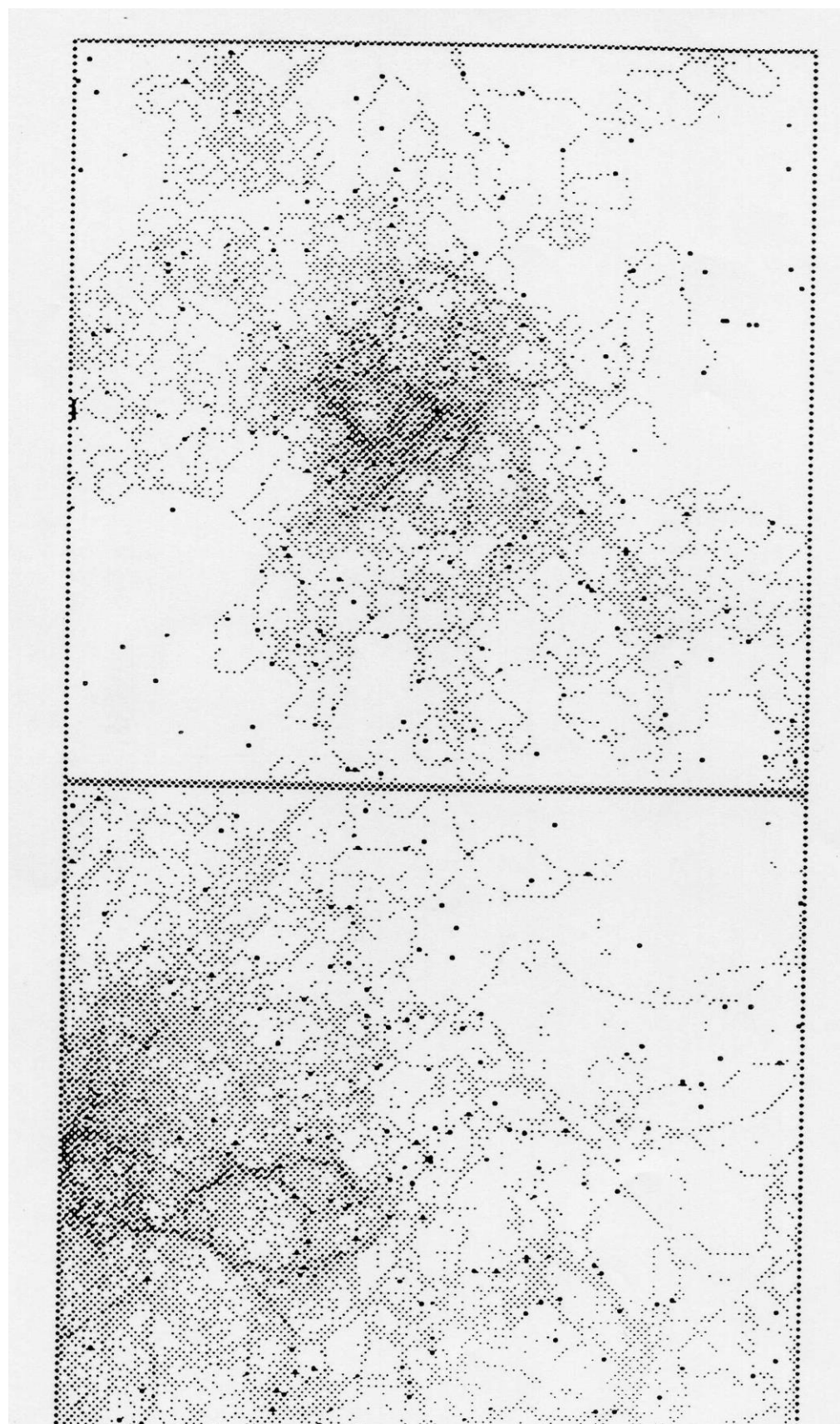
- Son uno de los primeros modelos de Inteligencia de enjambres.
- Se basan en el comportamiento colectivo de las hormigas en la búsqueda de alimentos.
- Resulta fascinante que animales casi ciegos, moviéndose prácticamente al azar, pueden encontrar el camino más corto desde su nido hasta los alimentos y regresar.

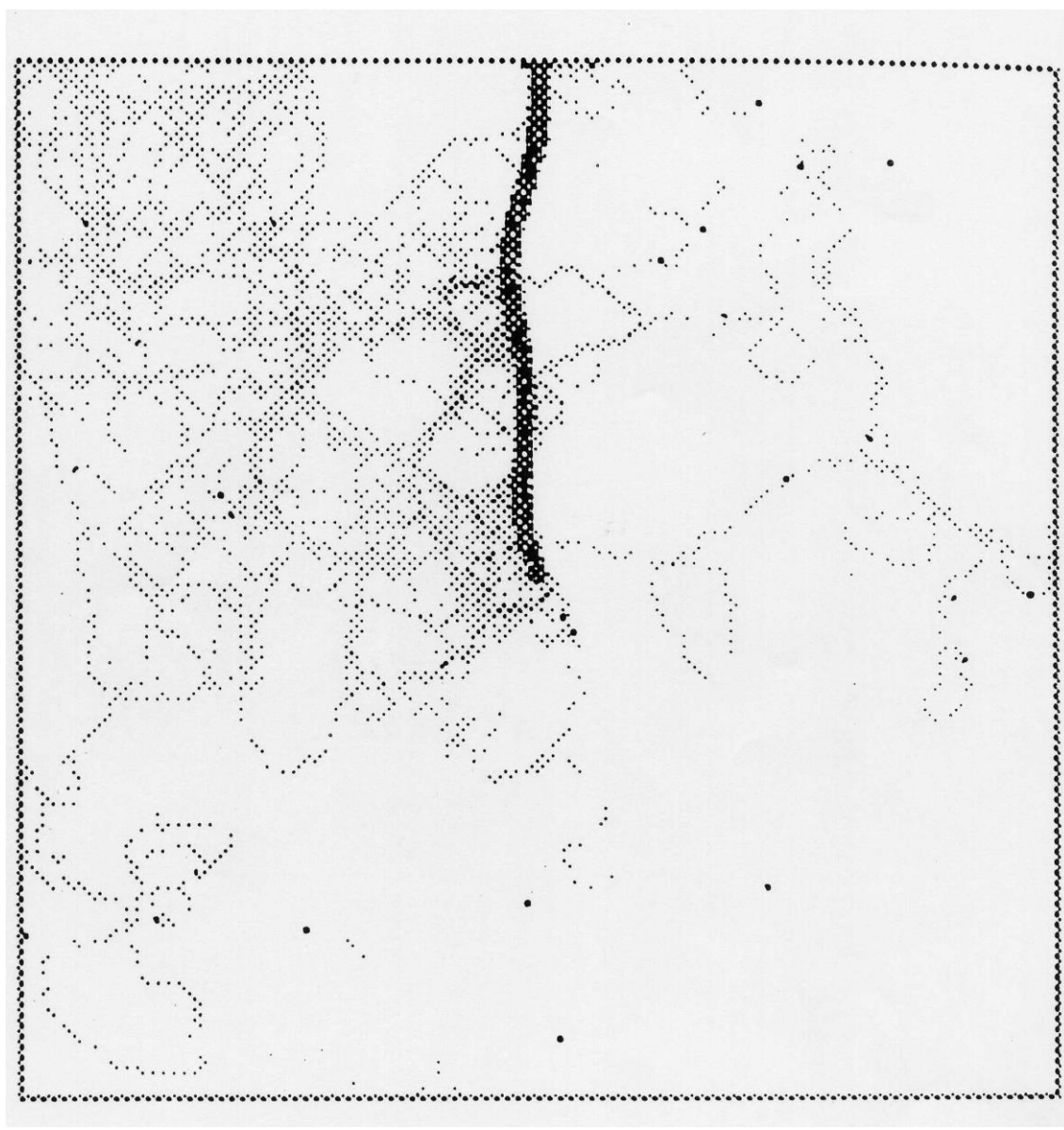
Algoritmos basados en el comportamiento de las hormigas

- Cuando una hormiga se mueve, deja una señal odorífera, depositando una sustancia denominada feromona, para que las demás puedan seguirla.
- En principio, una hormiga aislada se mueve esencialmente al azar, pero las siguientes deciden con probabilidad proporcional a la cantidad de feromonas.

ALGORITMOS BASADOS EN HORMIGAS

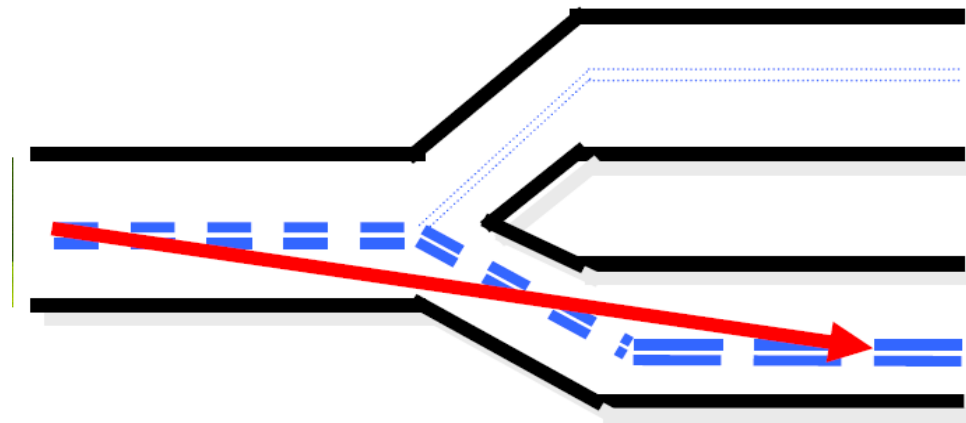






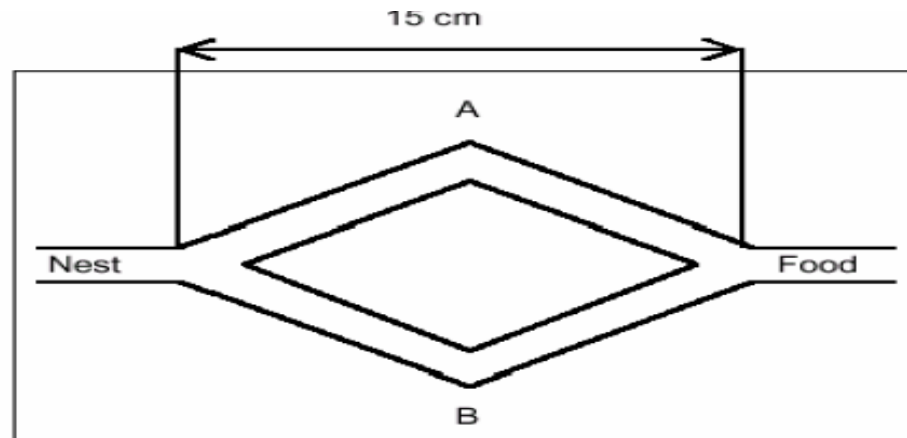
Algoritmos basados en el comportamiento de las hormigas

- Las hormigas depositan feromona
- En una bifurcación escogen al zar com mayor probabilidad en función de la feromona encontrada



Algoritmos basados en el comportamiento de las hormigas

- Experimento de puente simétrico



- Al cabo de un tiempo todas las hormigas van por una rama o la mitad por una y la mitad por la otra.

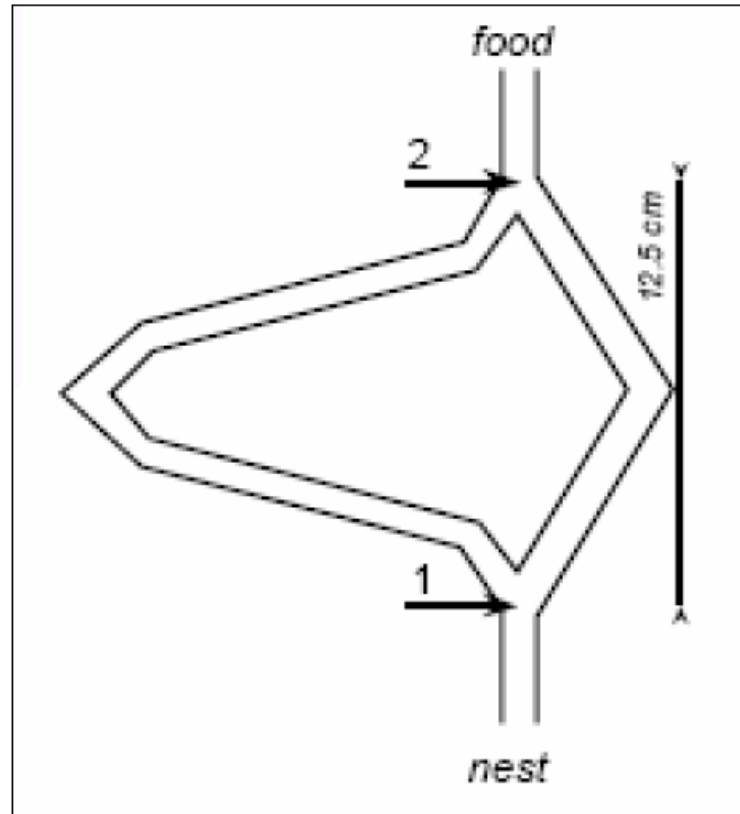
Algoritmos basados en el comportamiento de las hormigas

$$P_A = \frac{(k + A_i)^n}{(k + A_i)^n + (k + B_i)^n} \quad A_i + B_i = i$$

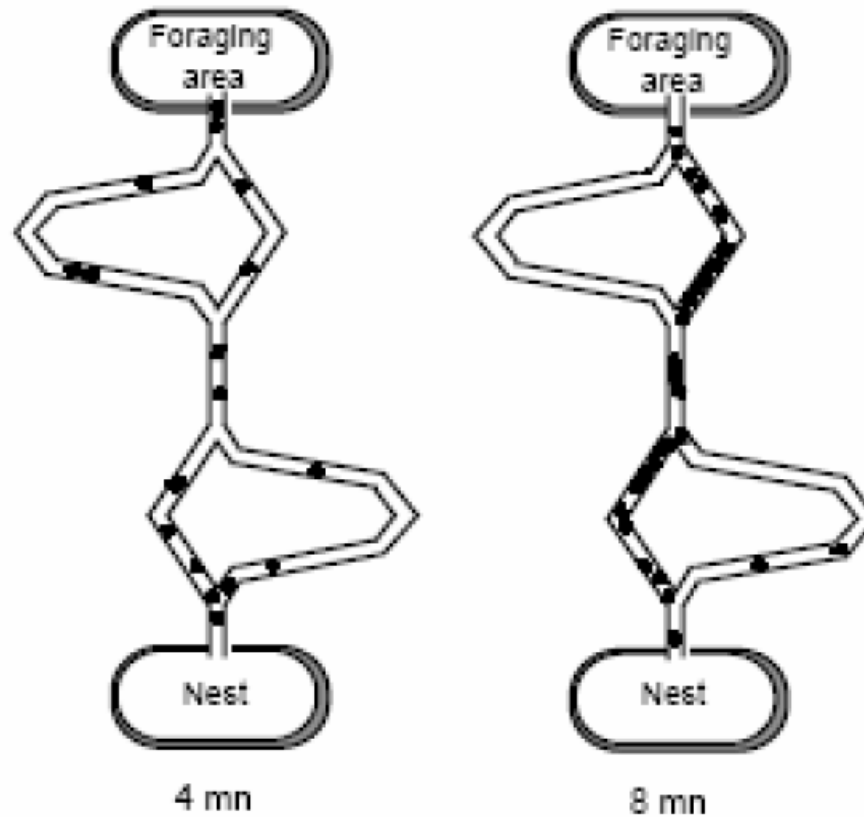
$n=2$ $k \approx 20$ (Parámetros experimentales)

Algoritmos basados en el comportamiento de las hormigas

Algoritmos basados en el comportamiento de las hormigas



Algoritmos basados en el comportamiento de las hormigas



Algoritmos basados en el comportamiento de las hormigas

Hormiga Artificial (agente simple):

- Mecanismo de deposición de feromona,
- Mecanismo de evaporación de feromona,
- Mecanismo probabilístico de elección en una bifurcación.,
- Información heurística acerca del problema concreto

Algoritmos basados en el comportamiento de las hormigas

- Se emplean para resolver problemas que se puedan formularse como problemas de camino mínimo en un grafo.
- Los métodos basados en el comportamiento de las hormigas se engloban en la INTELIGENCIA DE ENJAMBRES.

ALGORITMOS BASADOS EN EL COMPORTAMIENTO DE LAS HORMIGAS

AS: Ant System

ACS: Ant Colony System

MMAS; Max-Min AS

.....

Todos ellos formulan matemáticamente el comportamiento de las hormigas en busca de comida.

ALGORITMOS BASADOS EN EL COMPORTAMIENTO DE LAS HORMIGAS

Aplicación: Problemas combinatoriales.

- Problema del viajante de comercio,
- Problema de asignación cuadrática,
- Problema de asignación de tareas,
- Problema de coloreado de un grafo,
- Problemas de enrutamiento: canales, de distribución, Telecomunicaciones.

ALGORITMOS BASADOS EN EL COMPORTAMIENTO DE LAS HORMIGAS

Ingredientes:

1. Representación del problema
2. Regla de transición probabilística
3. Medida de deseabilidad de una solución
4. Método de satisfacción de restricciones
5. Regla de cambio de la feromona.

OTROS COMPORTAMIENTOS DE LOS INSECTOS SOCIALES

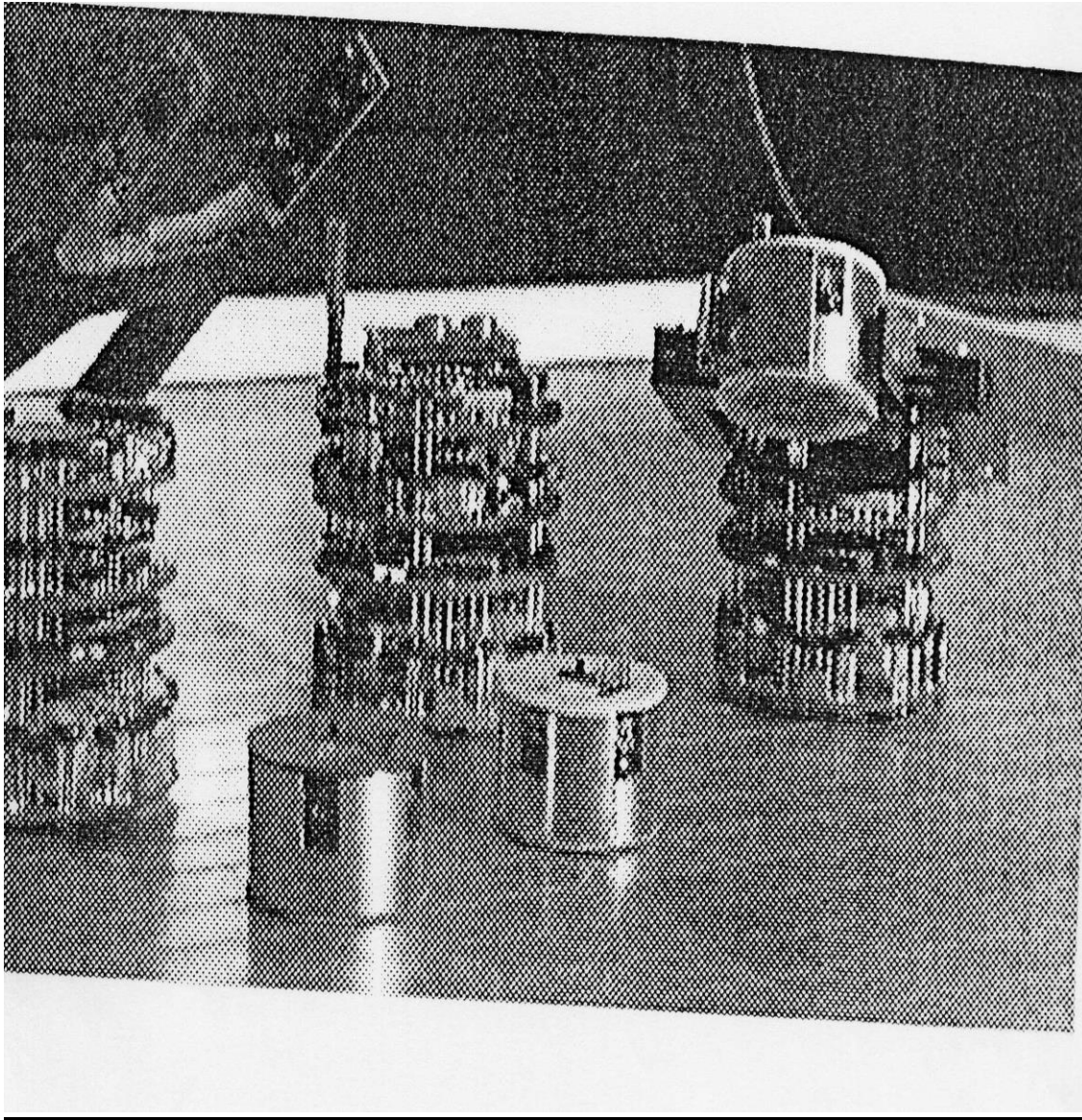
- División del trabajo y asignación de tareas,
- Organización de cementerios,
- Gestión de las crías,
- Transporte cooperativo

ROBOTICA EN ENJAMBRES

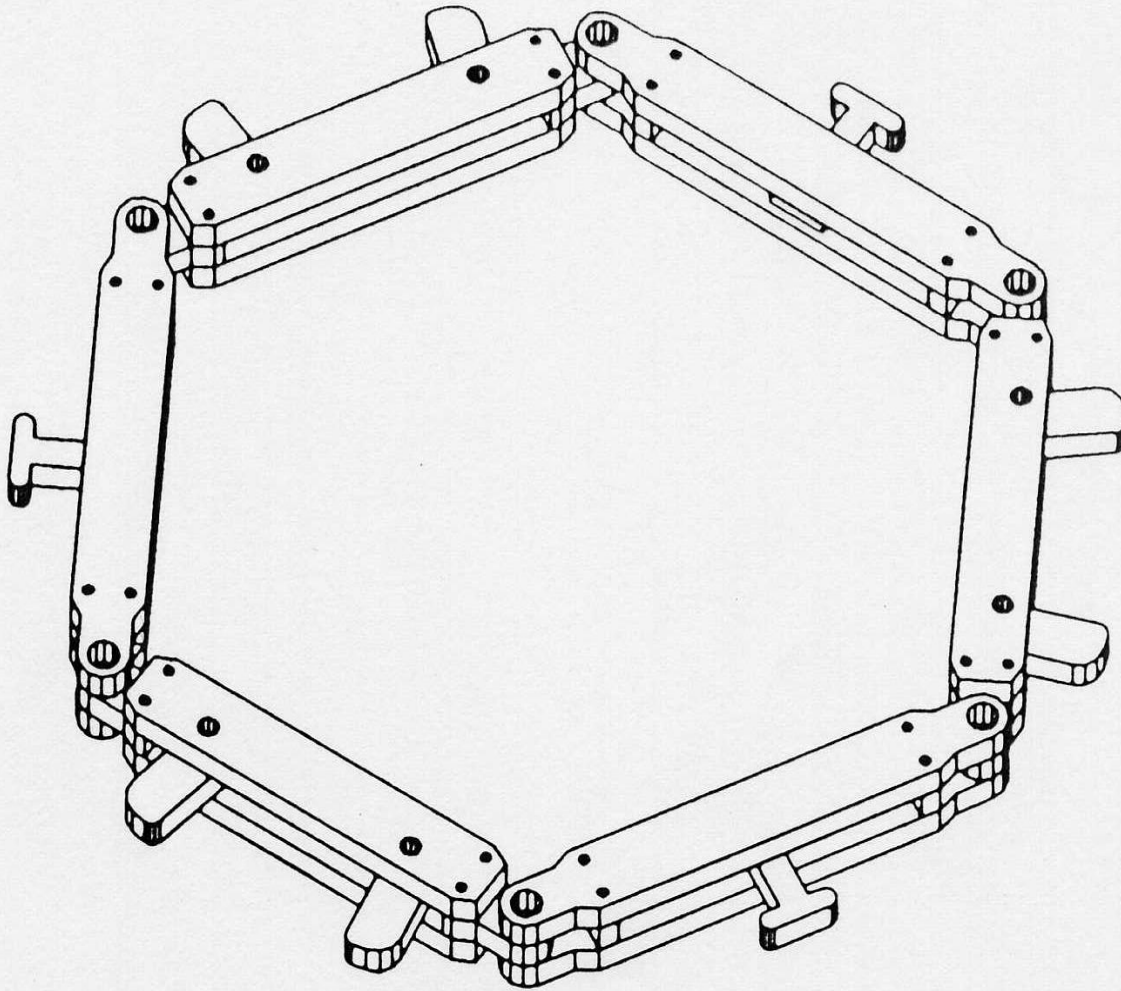
Combinación de la Autoorganización con la Robótica Colectiva.

- “Fallo” de la Inteligencia Artificial “macroscópica” en ciertos problemas,
- Miniaturización del hardware,
- Desarrollo del campo de la “Vida Artificial”,
- Vanguardia actual: Nanotecnología

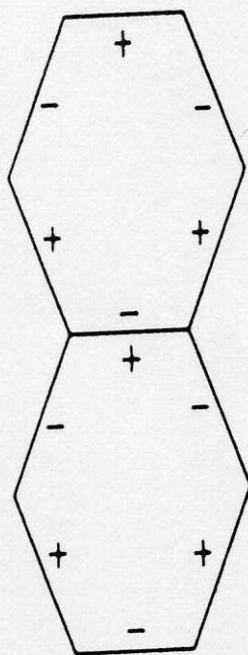
ROBOTICA EN ENJAMBRES



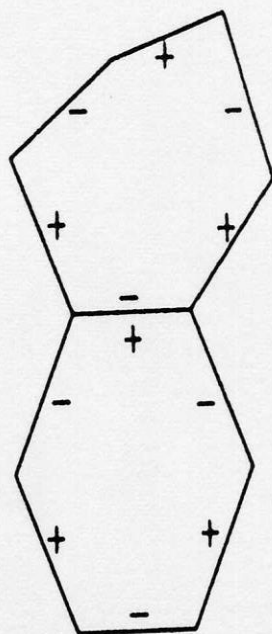
AUTOORGANIZACIÓN



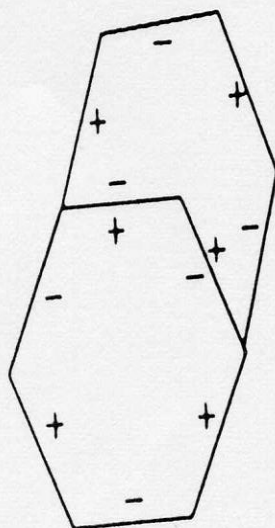
or hexagonal module. After Pamecha et al.



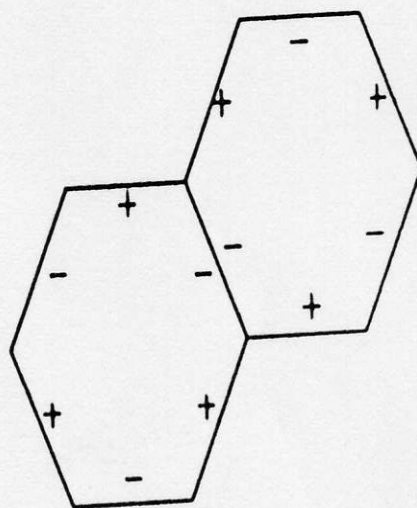
(a)



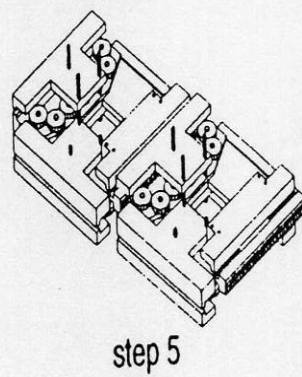
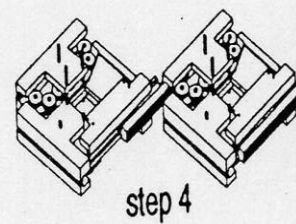
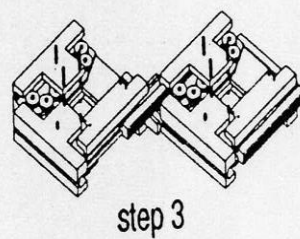
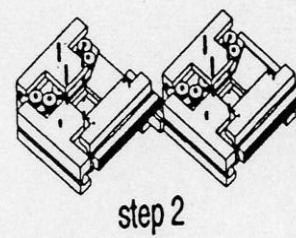
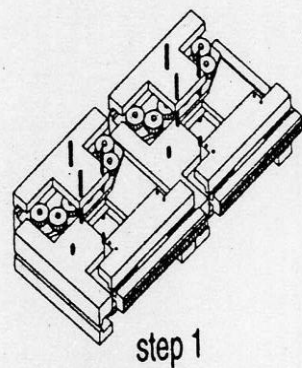
(b)



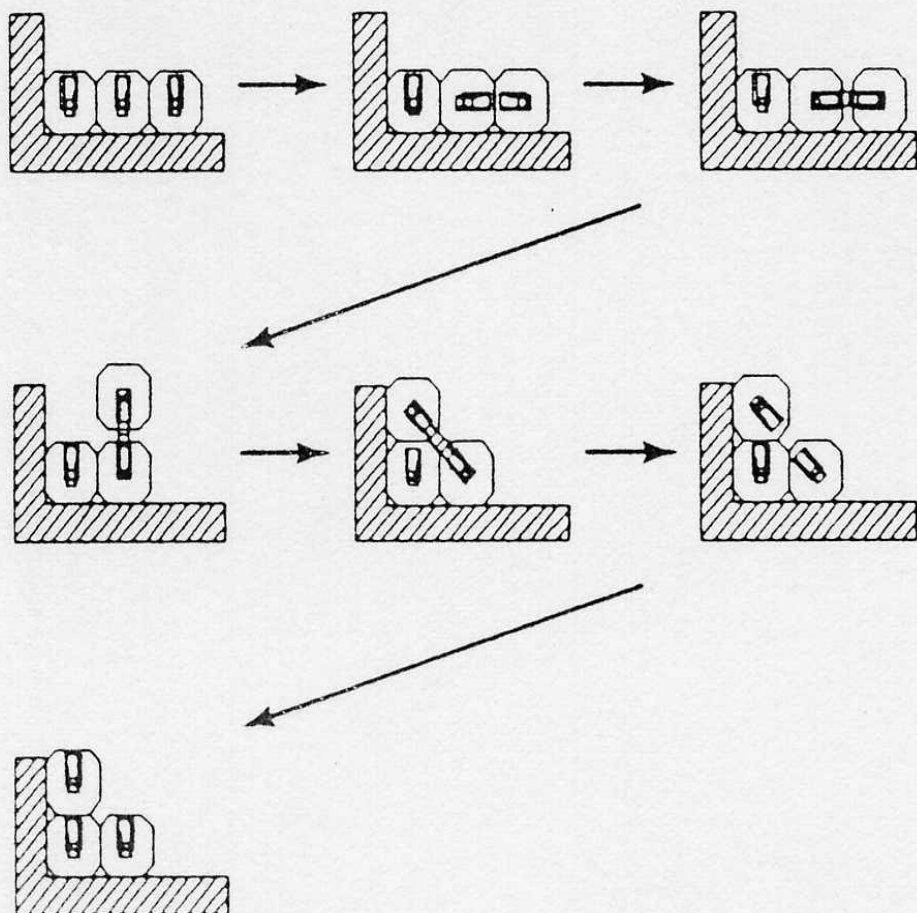
(c)

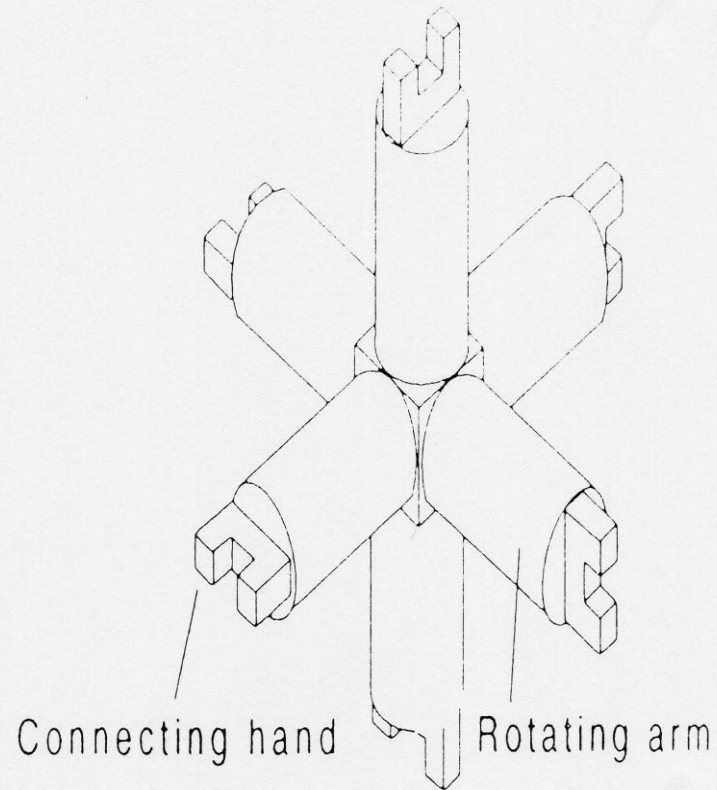


(d)



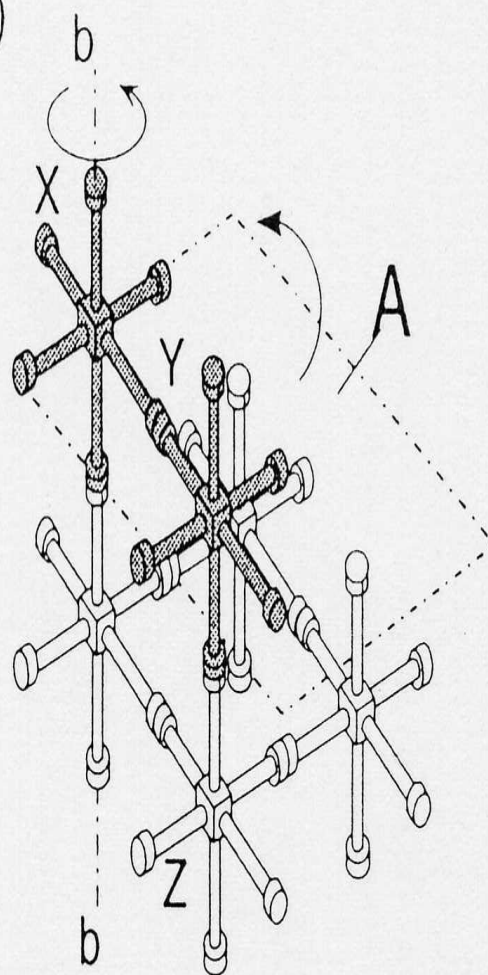
(a)



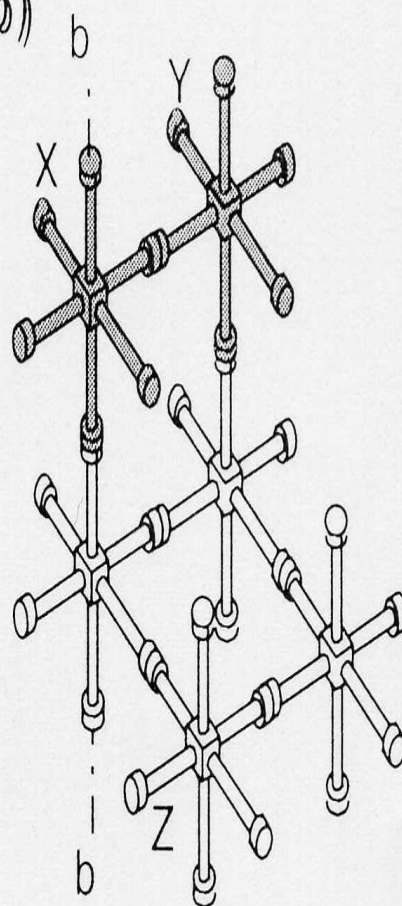


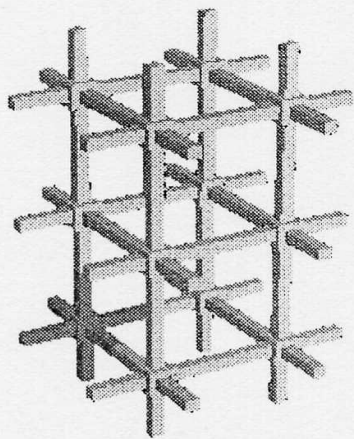
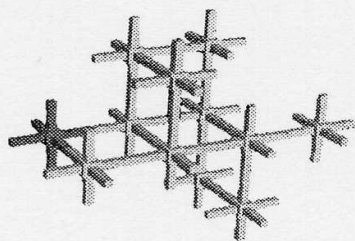
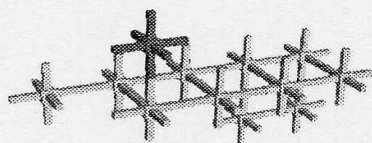
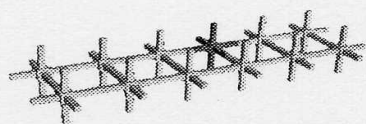
Schematic representation of a unit. After Yoshida et al.
© *IEEE Press*.

(a)



((b))





Redes Neuronales Artificiales

- (Modelos Conexionistas)
- Son modelos de aprendizaje y aproximación inspirados en el comportamiento del cerebro biológico.

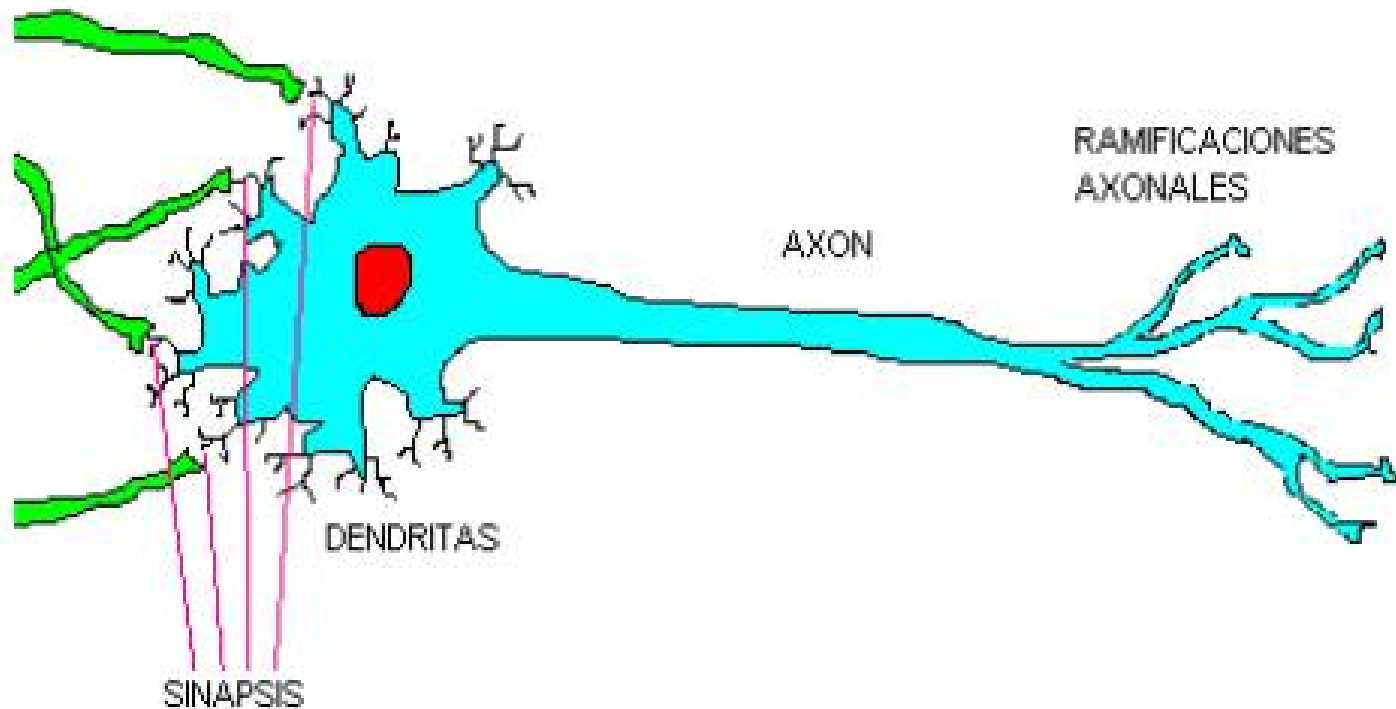
Redes Neuronales Artificiales

- El cerebro humano se compone de millones de neuronas interconectadas entre sí.
- Una neurona típica recoge señales procedentes de otras neuronas a través de unas estructuras llamadas dendritas.
- La neurona emite impulsos eléctricos a lo largo de una fibra axón, que se escinde en millares de ramificaciones.

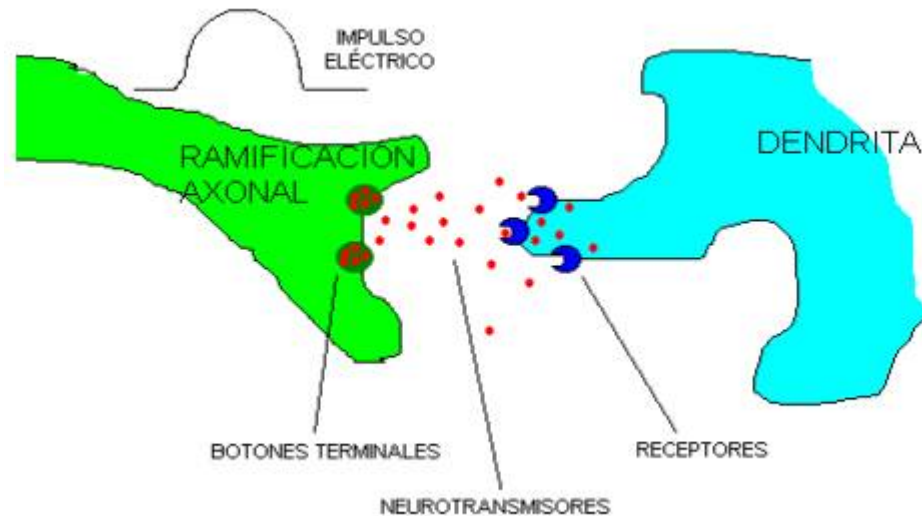
Redes Neuronales Artificiales

- Estas ramificaciones llegan hasta las dendritas de otras neuronas y establecen unas conexiones llamadas sinapsis.
- De esta manera la información se transmite de unas neuronas a otras y va siendo procesada a través de las conexiones sinápticas y las propias neuronas.

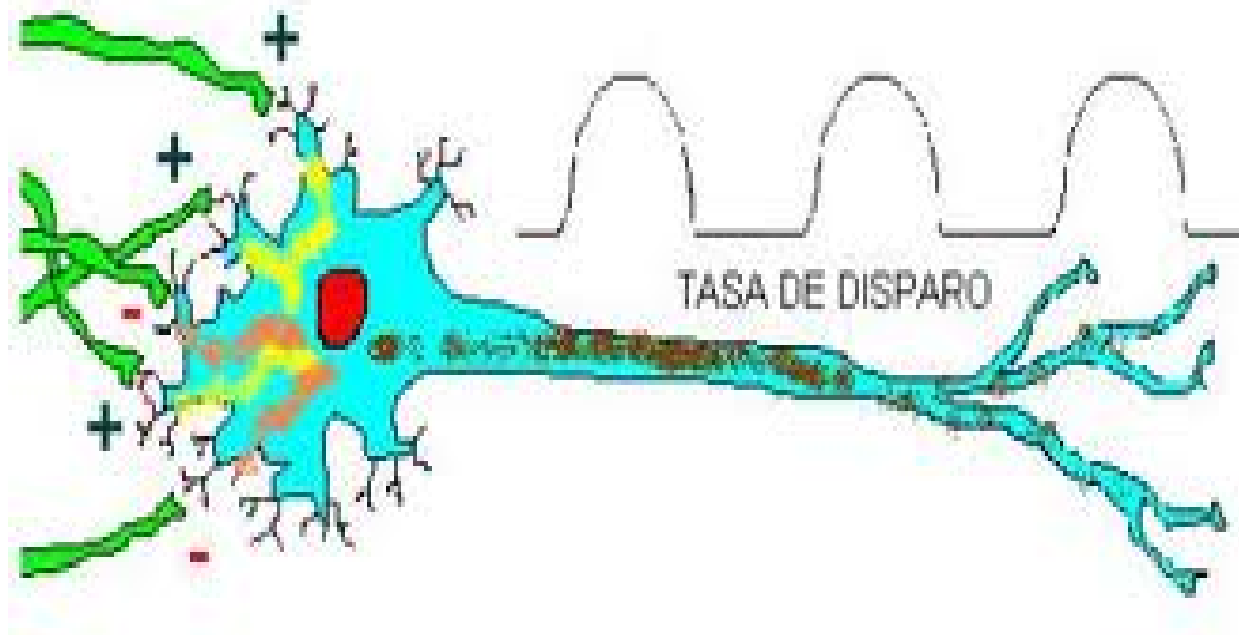
Redes Neuronales Artificiales



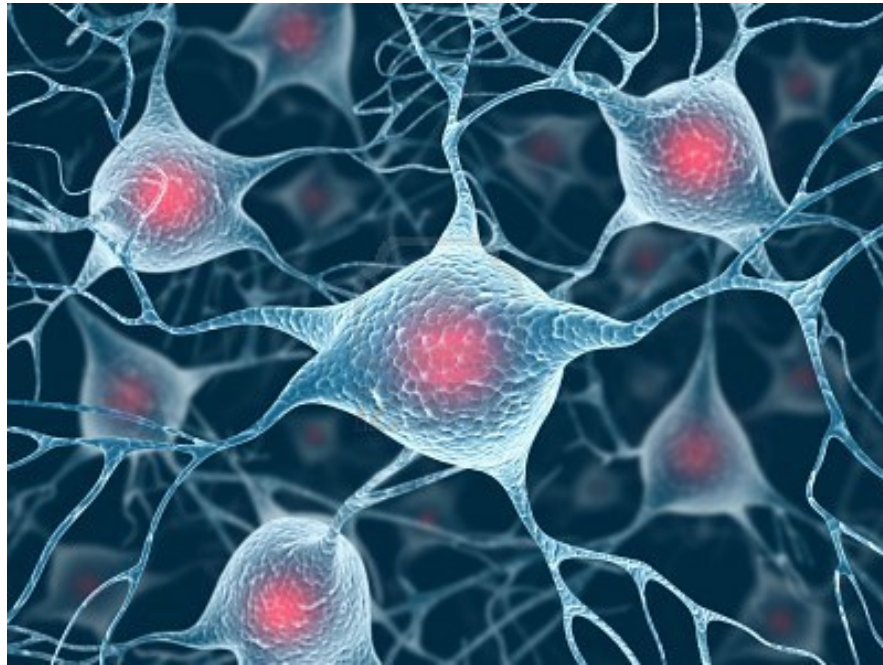
Redes Neuronales Artificiales



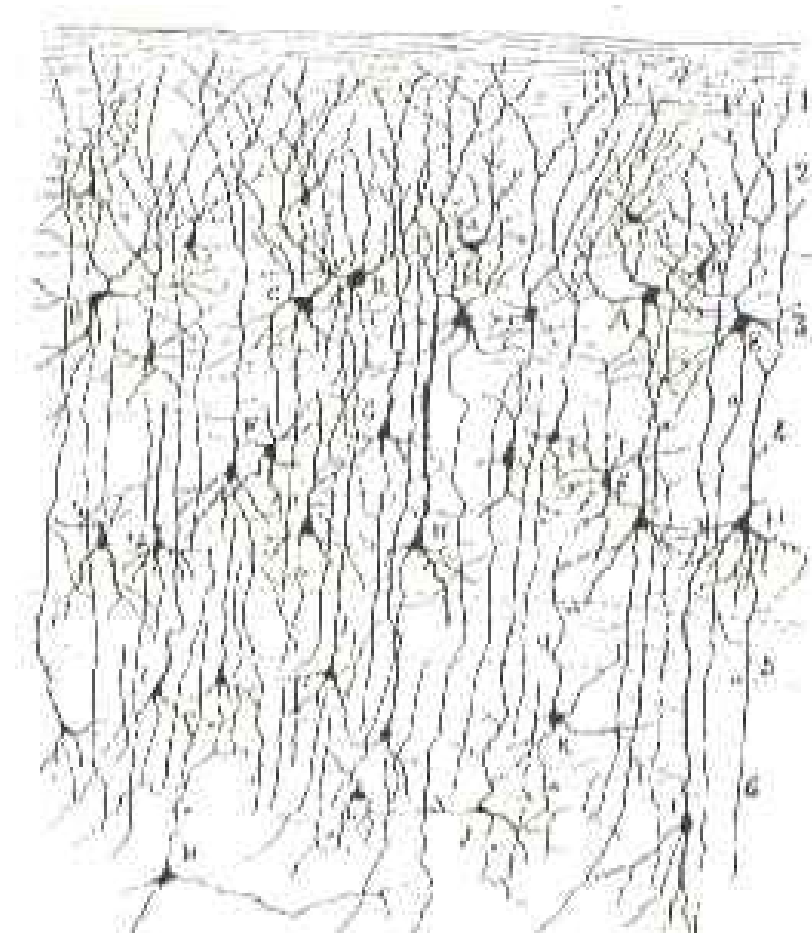
Redes Neuronales Artificiales



Redes Neuronales Artificiales

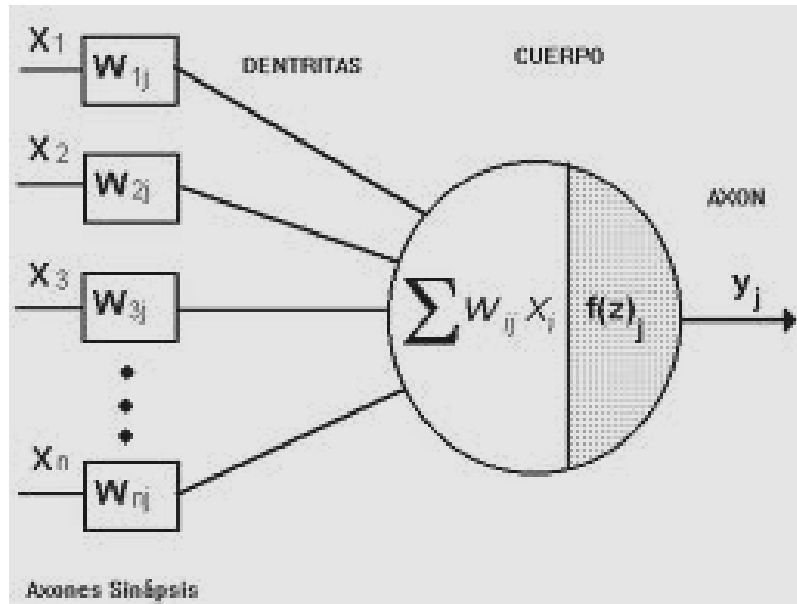


Redes Neuronales Artificiales



Redes Neuronales Artificiales

- El origen de los modelos conexionistas se situa en la definición de la neurona formal dada por McCulloch y Pitts en 1943 como un autómata con umbral



Redes Neuronales Artificiales

- Neural Network Study (1988, AFCEA International Press):

... a neural network is a system composed of many simple processing elements operating in parallel whose function is determined by:

- *network structure, connection strengths, and*
- *the processing performed at computing elements or nodes.*

Redes Neuronales Artificiales

- Haykin, S. Neural Networks: A Comprehensive Foundation (1994, Macmillan NY) :

A neural network is a massively parallel distributed processor that has a natural propensity for storing experiential knowledge and making it available for use.

Redes Neuronales Artificiales

It resembles the brain in two respects:

- 1. Knowledge is acquired by the network through a learning process.*
- 2. Interneuron connection strengths known as synaptic weights are used to store the knowledge.*

Redes Neuronales Artificiales

- Las RNAs son « máquinas numéricas» compuestas de gran cantidad de procesadores (neuronas) conectados entre sí y actuando en paralelo.
- Los modelos de RNs biológicos son mucho más complejos que los modelos de RNAs actuales.

Redes Neuronales Artificiales

- El comportamiento de una RNA está determinado por:
- las características de las neuronas,
- la topología de la red y
- los pesos de las conexiones.

Redes Neuronales Artificiales

Existen muchos tipos de modelos para las neuronas artificiales:

- Automatas con umbral
- Adaptiva Linear Neuron (ADALINE)
- Neuronas binarias sin bucle (Hopfield),
- Etc.

Redes Neuronales Artificiales

En cuanto a la topología también hay mucha variedad con tres formas básicas:

- 1. Conexiones hacia delante.*
- 2. Conexiones laterales.*
- 3. Conexiones hacia atrás (o recurrentes).*

Redes Neuronales Artificiales

Neuronas Biológicas	Neuronas Artificiales
Neuronas	Unidades de proceso
Conexiones sinápticas	Conexiones ponderadas
Efectividad de las sinápsis	Peso de las conexiones
Efecto excitatorio o inhibitorio de una conexión	Signo del peso de una conexión
Efecto combinado de las sinápsis	Función de propagación o de red
Activación -> tasa de disparo	Función de activación -> Salida

Redes Neuronales Artificiales

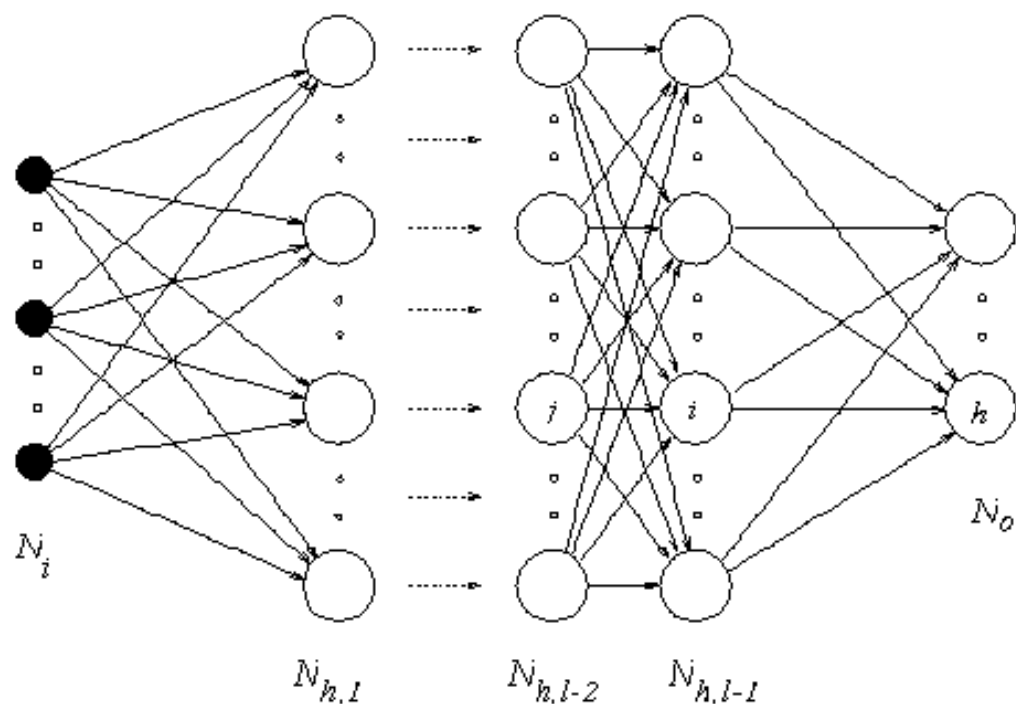
- Aprendizaje (entrenamiento)
- Determinación de los pesos de modo que la relación de entrada-salida de la red capture la información contenida en una tabla (X,Y) de pares de entrada salida.
- Propósitos: interpolación y generalización.
- Métodos: Minimización del error cuadrático

Redes Neuronales Artificiales

El paradigma de RNA es el perceptron multicapa:

- Automatas con umbral,
- Dos o mas capas de neuronas,
- Conexiones hacia adelante.

Redes Neuronales Artificiales



Redes Neuronales Artificiales

Un perceptrón multicapa encapsula una función

$$y=F(x)$$

$$x \in \mathbb{R}^{n_i} ; y \in \mathbb{R}^{n_o}.$$

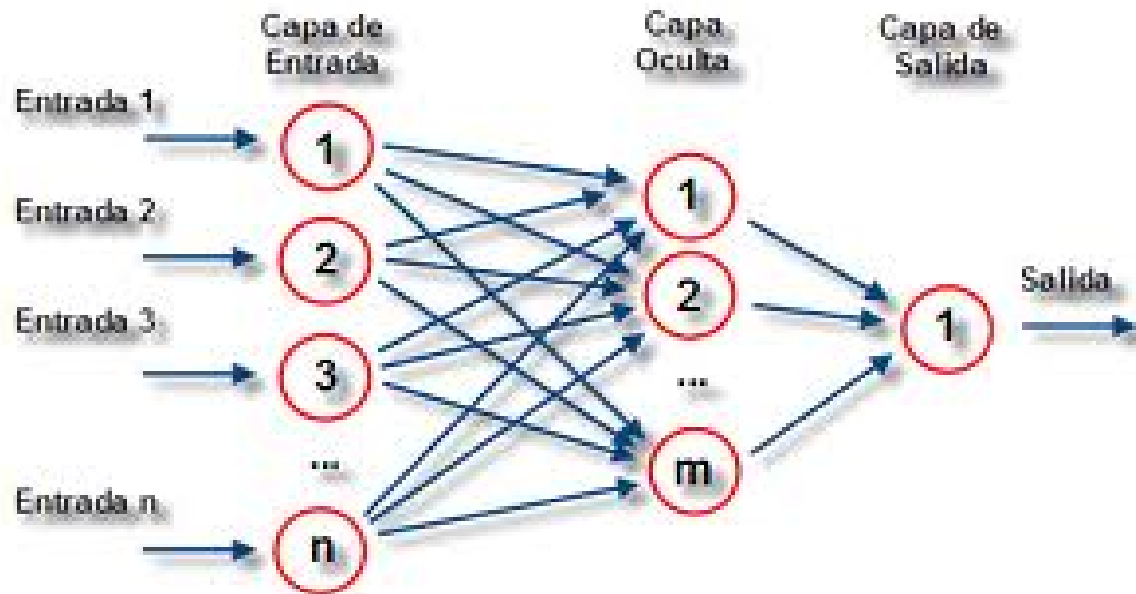
Un perceptron puede “aprender”, ajustando los pesos de las conexiones, cualquier relación de entrada salida $y=F(x)$ a partir de una tabla de ejemplos con un cierto margen de error.

Redes Neuronales Artificiales

- Teorema:

Un perceptron con dos capas y un numero “adecuado” de neuronas en la capa oculta, puede “aprender”, ajustando los pesos de las conexiones, cualquier relación de entrada salida $y=F(x)$ a partir de una tabla de ejemplos con un margen de error tan pequeño como se desee.

Redes Neuronales Artificiales



Modos de interconexión

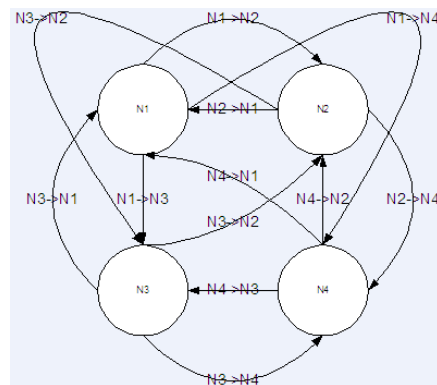
Feedforward: La red se organiza por capas de modo que la salida de una capa constituye la entrada de la siguiente.

Recurrentes: estáticas/ dinámicas

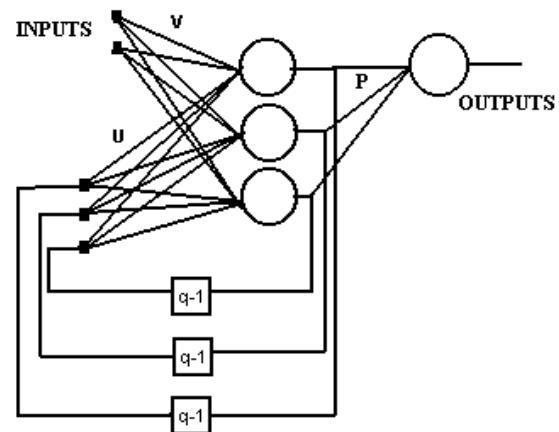
Existe algún tipo de recurrencia.

• **PARTICULARIDAD:** PRESENTAN ALGÚN TIPO DE REALIMENTACIÓN.

• **EJEMPLOS:** HOPFIELD, SOM, ELMAN, RECURRENTES DE SEGUNDO ORDEN...

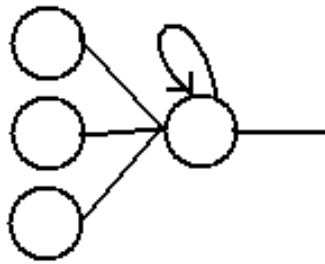


RNR
HOPFIELD

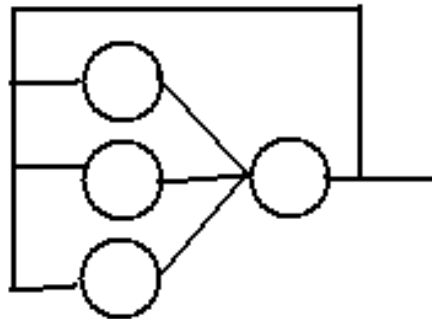


RNR ELMAN

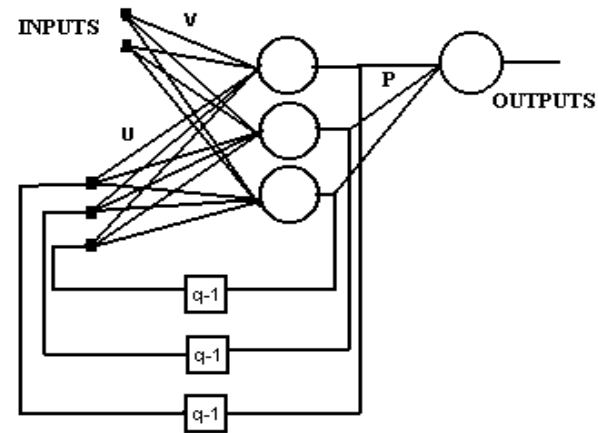
TIPOS DE RECURRENCIA



**RECURRENCIA
LOCAL**



**RECURRENCIA
GLOBAL**



**RECURRENCIA
MIXTA**

Enfoques actuales de los modelos conexionistas

- Enfoque computacional: se intenta desarrollar modelos de computación eficientes,
- Enfoque cognitivo: se interesa sobre todo por las capacidades cognitivas de estos modelos,
- Enfoque biocognitivo: Parecido al anterior pero toma como premisa la plausibilidad biológica de los modelos.

AUTOMATAS CELULARES

Un *Autómata Celular*, (AC), es un modelo formal que está compuesto por un conjunto de *células*, *entes* o *agentes*, cada uno de ellos susceptible de encontrarse en un cierto estado y de cambiarlo de un instante al siguiente, asumiendo que el tiempo transcurre de forma discreta.

La regla que gobierna la transición de estados en los entes es sensible a los estados de los demás elementos en su vecindad, siendo por tanto una *regla de transición local*.

El aspecto que más caracteriza a los ACs es su capacidad para dotar al conjunto del sistema, visto globalmente, de una serie de propiedades emergentes inducidas por la propia dinámica local.

Un Autómata Celular tiene las siguientes características:

1. Un espacio formado por un conjunto finito de celdas distribuidas en una rejilla regular n-dimensional.
2. Cada celda puede estar en un único estado en determinado instante de tiempo.
3. El estado de las celdas cambia de un instante a otro de acuerdo a un conjunto de reglas de evolución comunes a todas las celdas.
4. El vecindario de una celda esta formado por las celdas adyacentes y se define igual para todas las celdas del autómata. El vecindario influye en el cambio de estado.



El primer AC estudiado se debe a John von Neumann (1903-1957), interesado por definir una estructura formal capaz de auto-replicarse, al modo como lo hacen los organismos biológicos.

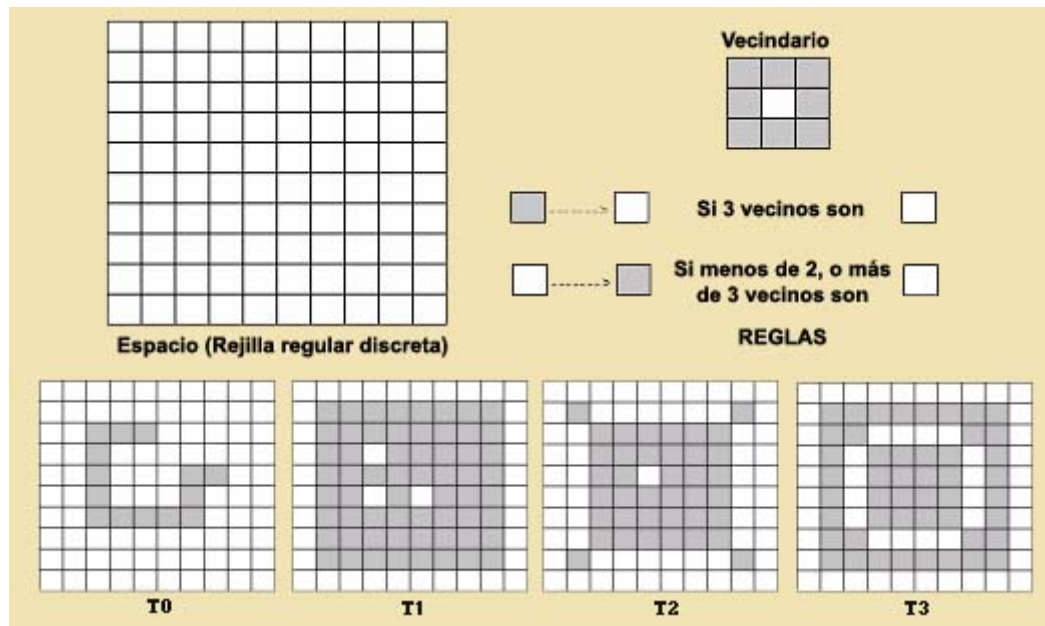
Cuando el número de celdas es finito, las vecindades en los bordes o extremos deben ser tratadas de forma especial. Se suelen considerar las siguientes condiciones:

- Bordes periódicos. Células opuestas se consideran vecinas, de forma que en un AC de una dimensión, el segmento se convierte en una circunferencia y en un retículo plano la superficie se convierte en un toro.
- Bordes absorbentes. Las células de los bordes no tienen vecinos más allá de los límites del retículo.

En 1970 se publicó como una curiosidad de matemática recreativa el *Juego de la Vida* (JV), ideado por John Conway.

Se trata de un AC donde las células pueden estar en un estado de vivas (1) o muertas (0).

En la siguiente figura se presenta un ejemplo del autómata que define el "Juego de la Vida".



Autómata Celular del "Juego de la Vida".

Partiendo de cierta configuración inicial, los estados de las células evolucionan en el tiempo.

Algunos grupos de células forman agregados impredecibles y extraños objetos aparecen, se mueven y desaparecen por todo el plano.

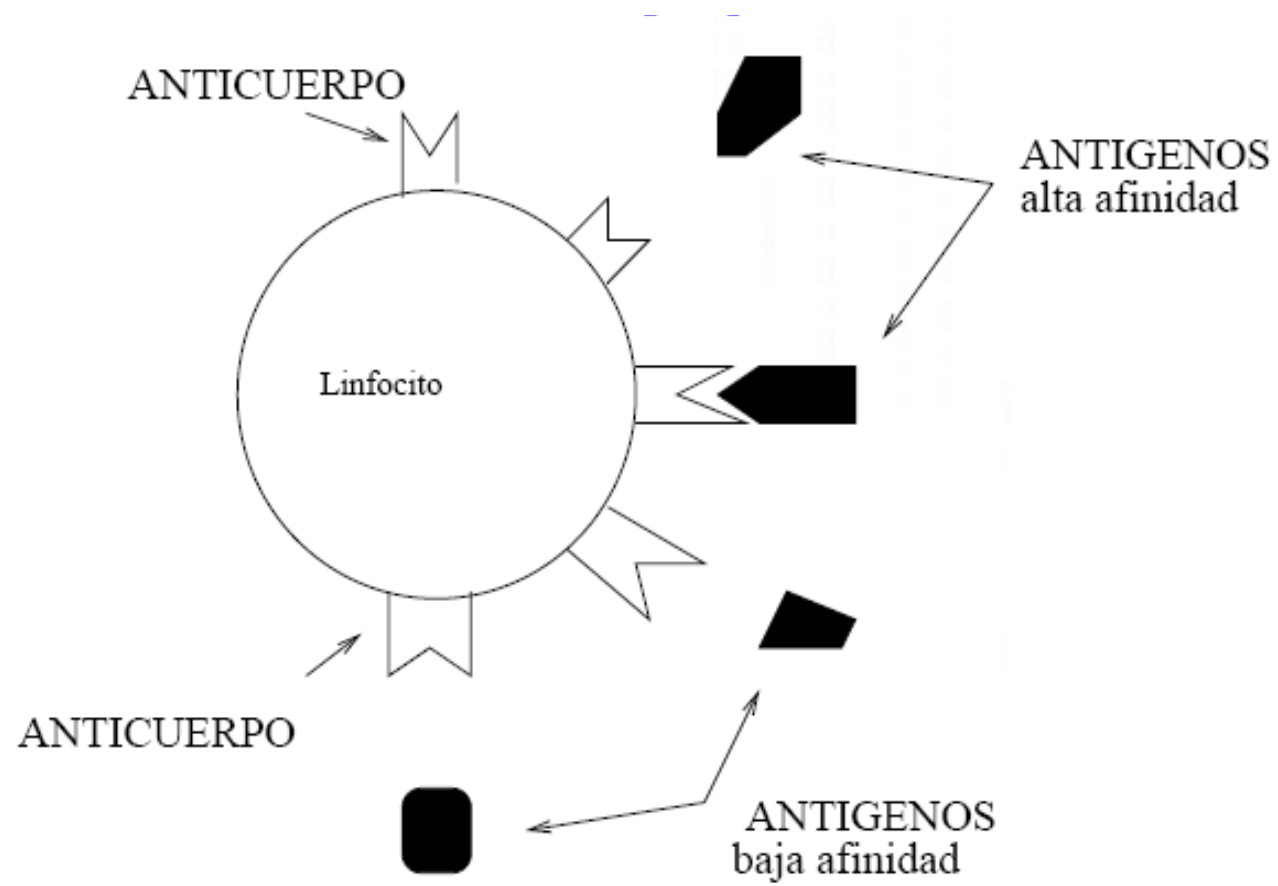
Mucha gente empezó a jugar al JV buscando configuraciones iniciales que evolucionasen hacia estados más o menos curiosos.

Se consiguió demostrar que el JV es equivalente a la Máquina de Turing Universal, idea que justificó el interés por la construcción de máquinas de cómputo paralelo o celulares.

Los años 80 fueron la gran década de todos estos modelos de procesamiento en paralelo. Brevemente, citar el exhaustivo trabajo de Stephen Wolfram sobre los *Autómatas Celulares Lineales*.

SISTEMA INMUNE ARTIFICIAL

Los sistemas inmunes artificiales son sistemas adaptativos inspirados por la teoría de la inmunología y funciones, principios y modelos observados en el sistema inmune, que se emplean para solucionar problemas.



Características del sistema inmune

Aprendizaje

Memoria

Descentralizado

Reconocimiento de patrones

Paralelo

Diversidad

Autoregulatorio

Detección distribuida

Aunque la cantidad de modelos y aplicaciones del sistema inmune va en ascenso, solo existe un esquema muy general de cuáles son los elementos esenciales que un sistema inmune artificial debe poseer.

- Representación de los componentes del sistema,
- Un conjunto de mecanismos para evaluar las interacciones,
- Un proceso de adaptación que gobierne las dinámicas del sistema.

Existen varios modelos del sistema inmune artificial, quizá debido a que su campo de aplicación es muy amplio y a la gran complejidad del sistema inmune biológico.

- _ Modelos de timo y de médula ósea (bone marrow)
- _ Algoritmo de selección clonal
- _ Teoría de red inmune

Los modelos de timo son aquellos basados en el proceso del sistema inmune mediante el cual se crea la población de células detectoras.

Los modelos de médula ósea se refieren a la creación y maduración de los detectores del sistema inmune.

Los modelos que utilizan el algoritmo de selección clonal emulan el proceso mediante el cual el sistema inmune, ante la presencia de un antígeno específico, estimula únicamente a aquellos linfocitos que sean más afines, para después ser clonados y mutados.

Los modelos que se basan en la teoría de red inmune, emulan las interrelaciones que existen entre anticuerpos, aún en una ausencia de antígenos, es decir, la estimulación y supresión entre anticuerpos.

Los campos de aplicación del sistema inmune artificial son muy variados:

- _ Robótica
- _ Seguridad en sistemas de cómputo y redes
- _ Optimización
- _ Reconocimiento de patrones
- _ Planificación de horarios
- _ Aprendizaje Automático
- _ Control

COMPUTADORAS DE ADN

El 11 de Noviembre de 1994, Leonard M. Adleman publicó un artículo en "Science" donde describía la "Computación Molecular de Soluciones a Problemas Combinatorios".

Se considera la primera implementación de una computadora basada en ADN.

En particular, el experimento de Adleman logró resolver el problema de la Ruta Hamiltoniana para una pequeña cantidad de nodos.

Este problema consiste en hallar una ruta que recorra todos los nodos de un grafo, pasando sólo una vez por cada uno de ellos.

El problema resulta muy difícil para las computadoras convencionales porque es de tipo NP.

El problema de la Ruta Hamiltoniana se resuelve con el siguiente algoritmo:

1. Generar rutas aleatorias a través del grafo.
2. Conservar sólo aquellas rutas que inicien en el nodo inicial y concluyan en el nodo final.
3. Si el grafo tiene n nodos, conservar sólo aquellas rutas que contengan n nodos.
4. Conservar sólo aquellas rutas que tocan todos los nodos al menos una vez.
5. Cualesquiera rutas restantes son soluciones al problema.

El elemento central de la solución usando ADN fué establecer los equivalentes bioquímicos adecuados de los pasos correspondientes al algoritmo especificado.

Las operaciones que se describen a continuación pueden realizarse con ADN en los laboratorios y se denominan "Modelo no restringido de cómputo con ADN":

Síntesis de una cadena genética deseada.

Separación de cadenas considerando su longitud.

Mezcla, vertiendo dos tubos de ensayo en uno para realizar la unión.

Extracción, tomando aquellas cadenas que contengan un patrón determinado.

Fundir y/o templar, rompiendo o ligando dos moléculas de ADN con secuencias complementarias.

Amplificación, usando un compuesto denominado PCR para hacer copias de cadenas de ADN.

Corte, separando el ADN con enzimas de restricción.

Ligación, enlazando cadenas de ADN con límites complementarios "adherentes" usando un compuesto denominado ligasa.

Detección, confirmando la presencia o ausencia de ADN en un determinado tubo de ensayo.

Las operaciones mencionadas pueden usarse para "programar" una "computadora de ADN".

Adleman vislumbra la posibilidad de que una molécula simple de ADN pueda usarse para codificar la "descripción instantánea" de una MT, y que los protocolos bioquímicos y enzimas disponibles actualmente podrían, al menos bajo condiciones ideales, usarse para inducir modificaciones sucesivas en una secuencia de ADN, modificaciones que serían el equivalente de la ejecución de una MT.

La ejecución del experimento de Adleman llevó aproximadamente una semana.

Aunque este problema específico puede resolverse en papel en menos de una hora, cuando el número de nodos se incrementa a 70, el problema se vuelve excesivamente complejo aún para una super-computadora.

Actualmente, las super-computadoras más veloces pueden ejecutar 1000 millones de instrucciones por segundo (1000 MIPS).

Una molécula simple de ADN necesita aproximadamente 1000 segundos para ejecutar una instrucción, por lo cual su velocidad sería inferior a 0.001 MIPS.

Obviamente, si se desea realizar un solo cálculo a la vez (arquitectura secuencial), las computadoras de ADN no son una opción viable.

Sin embargo, si se desea ejecutar muchos cálculos simultáneamente (arquitectura paralela), una computadora como la descrita puede ejecutar fácilmente 10^{14} MIPS.

Las computadoras de ADN también requieren menos energía y espacio.

Mientras que las computadoras actuales ejecutan 10^9 operaciones por Joule de energía consumida, las computadoras de ADN podrían ejecutar 2×10^{19} operaciones.

Esto significa 10^{10} veces más eficiencia.

Los datos pueden almacenarse en el ADN a una densidad aproximada de 1 bit por nanómetro cúbico (nm^3), mientras que los medios actuales de almacenamiento requieren 10^{12} nm^3 para cada bit.

Al ver al ADN como elemento de cómputo, los bioquímicos pueden generar moléculas mediante nuevos procesos, que serían similares a algoritmos computacionales, con lo cual su nivel de control sería quizá mejor que el de los procesos bioquímicos tradicionales.

Algunas aplicaciones recientes incluyen, por ejemplo, la construcción de pseudo-enzimas.

A pesar de las actuales limitaciones físicas y lógicas del hardware de ADN, en el futuro el posible hardware biológico podría ser quizá más veloz que el electrónico para aplicaciones que requieran paralelismo.

Se tendría la ventaja de que lo vivo puede reproducirse por sí mismo, y eso es algo que las computadoras electrónicas actuales todavía no pueden hacer. La aplicación de un posible hardware biológico depende en gran medida de su posibilidad de automatización, que quizás no esté muy lejana.

