

# *Informática Evolutiva para la Optimización de Problemas*

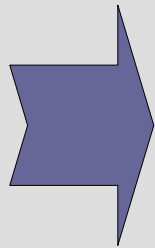
*Prof. Wílmer Pereira*

*Universidad Católica Andrés Bello  
Universidad Simón Bolívar*

<http://www.idc.usb.ve/~wpereira>

# Resolución de problemas complejos ...

- Hilbert formula 23 problemas en París. El segundo (mostrar la consistencia de las matemáticas a partir de su axiomática) era esperanzador ... pero ... Russel&Whitehead se topan con problemas ... además existen, según el teorema de incompletitud de Gödel, problemas indecibles ... IGNORABIMUS ...
- Entre los problemas decidibles, tenemos problemas cuya exigencias en tiempo de cálculo o peticiones de memoria crece exponencialmente, es decir, problemas intratables:



NP-Exp  
NP-Duros  
NP-Completos ...

- Soluciones posibles:



Dividir el problema y paralelizarlo pero ... sería necesario un crecimiento exponencial en la cantidad de procesadores disponibles :- ( ...



Introducir heurísticas particulares al problema que se pretende resolver pero ... no es general (*deep blue*) y no hay garantía de que las intuiciones conduzcan a una estrategia que permita llegar a las soluciones óptimas :- ( ...

# Ejemplo de Intratabilidad: satisfactibilidad

- Dada una tabla de verdad con 5 o 6 variables  $\rightarrow 2^5 = 32$  o  $2^6 = 64$  filas. Un humano con suficiente paciencia no tendría mayores problemas. Hasta cierto límite, un computador también puede realizar la tarea sin mayores inconvenientes ...
- Imaginemos un computador con el cual podemos procesar el valor de verdad de cada fila de la tabla en 1 nseg ( $10^{-9}$  seg). Una tabla de 40 variables implica  $2^{40}$  filas es decir  $10^{12}$  – alrededor de un billon de filas –  
Con un cálculo sencillo, el tiempo estimado sería de

$$10^{12} * 10^{-9} = 1000 \text{ seg}$$

... lo cual representa alrededor de 16 minutos de procesamiento.

- Con 100 variables, implica aproximadamente un quintillon de filas ( $10^{30}$ ).

$$10^{30} * 10^{-9} = 10^{21} \text{ seg}$$

lo que representa  $3.17 * 10^{11}$  siglos de cálculo ... solo para llenar la tabla ...  
Esto sin tomar en cuenta el espacio necesario para contener la tabla de un quintillon de filas en el disco del computador ...

# Genética y Evolución

## Base para la herencia y variabilidad de los seres vivos

- Dentro del núcleo se encuentra el ADN que permite ensamblar las proteínas.
- El ADN formado por cadenas de bases nitrogenadas:
  - Adenina
  - Citosina
  - Guanina
  - Timina
- La fecundación se realiza mediante el cruce (en la meiosis) y la mutación

- 
- Selección natural debida a la competencia (Darwin) y la presión ecológica (Wallace)
  - Gradualismo vs Equilibrio puntuado
  - Coevolución
    - Competitiva
      - Huesped-Parásito, Presa-Depredador o Conflicto Sexual
    - Cooperativa
    - Simbiosis

# Algoritmos Genéticos

## John Holland (1960)

### ● Codificación:

Gen: son las variables del problema codificadas en binario

Cromosoma: individuos o soluciones del problema

### ● Función de Aptitud:

Heurística para indicar la bondad de una solución o que tan apto es un individuo

### ● Criterios de Selección:

Ruleta: la probabilidad de seleccionar cada individuo es proporcional

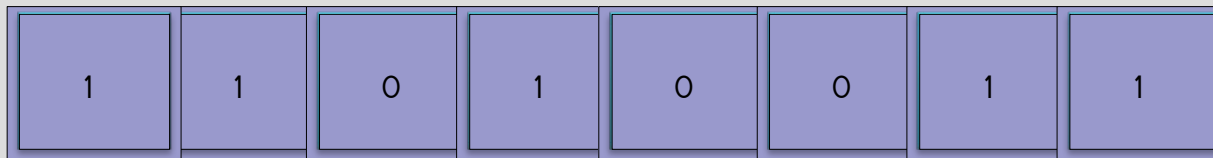
Torneo: selección por turnos de soluciones para escoger las dos mejores

Elitista: los individuos con mejor aptitud son seleccionados

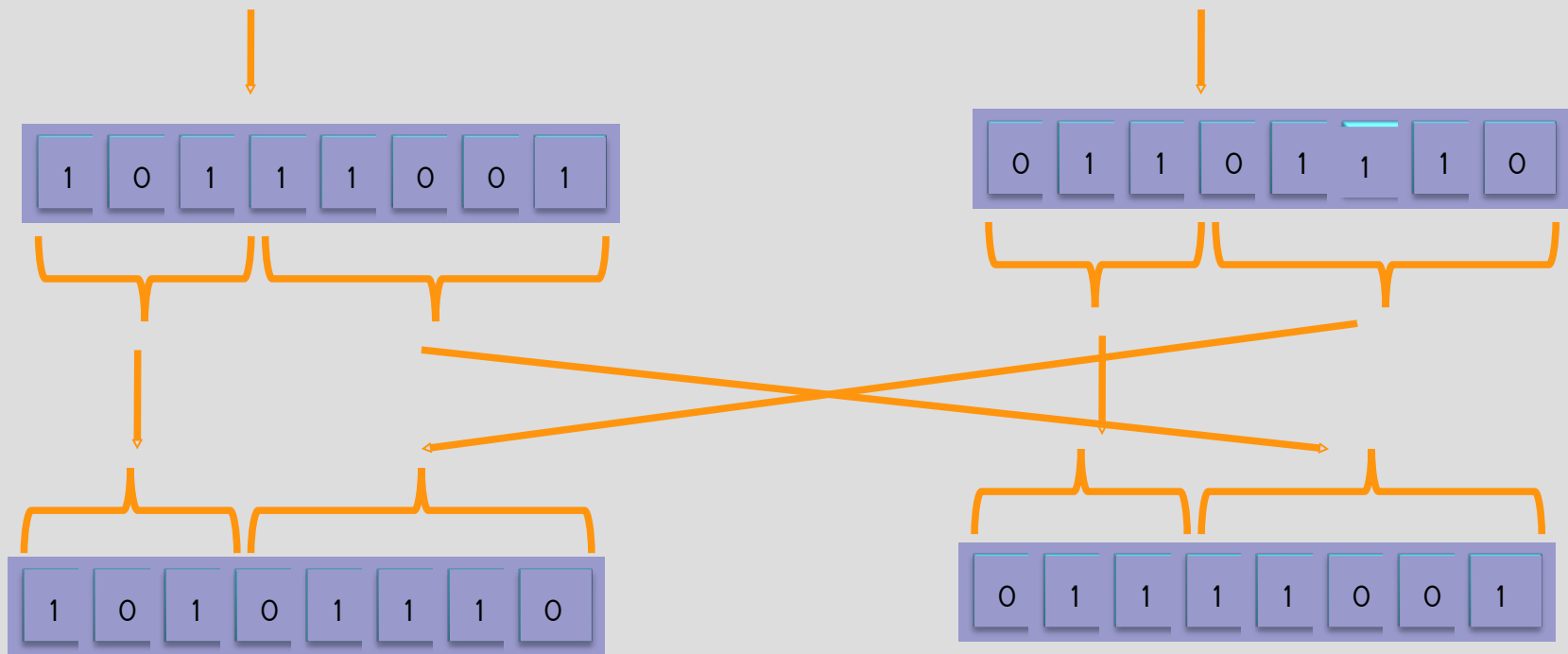
### ● Operadores Genéticos:

Cruce: nuevas soluciones a partir de dos padres

Mutación: probabilidad de cambiar aleatoriamente un bit de una variable



# Cruce



# Algoritmo Genético de Base

Generar una población de soluciones inicial

Seleccionar los individuos de la población  
(Soluciones del problema)

Cruzar los individuos

Mutar los individuos con una  
baja probabilidad

Insertar los nuevos individuos de la  
próxima generación

Parar según un criterio de satisfacción



# Condiciones de Aplicación de un Algoritmo Genético

- ¿Cómo codificar el problema, es decir, cuales son las variables relevante para resolverlo?
- ¿Qué método de selección es el adecuado para obtener los mejores resultados?
- ¿Cómo definir la función de aptitud para determinar que es una buena solución?
- ¿Cómo combinar la carga genética de los individuos para aumentar el valor de la función de aptitud?
- ¿Cuántas generaciones son necesarias para el mejor sub-óptimo posible?

*La respuesta a cada pregunta es dependiente del problema y no hay manera de saber con certeza la mejor opción ... afortunadamente se demostró la convergencia de la estrategia genética ... al menos ...*

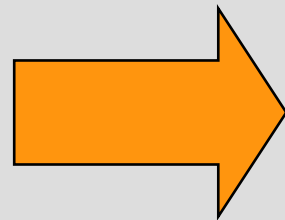


# Redes Neurales Naturales

Estructura celular del cerebro donde residen las capacidades intelectuales del hombre. Desde  $100 \times 10^9$  hasta  $10 \times 10^{12}$  neuronas ...

<u>Neurona:</u>	Célula nerviosa
<u>Soma:</u>	Núcleo celular
<u>Dendritas:</u>	Ramificaciones entre neuronas
<u>Sinapsis:</u>	Punto de unión entre dendritas 10.000 en promedio por neurona

Reacciones  
Electroquímicas



Impulsos Inhibidores o  
Impulsos Excitatorios



- Interneuronas
- Neuronas motoras (directo al músculo)
- Neuronas receptoras (directo desde el órgano sensor)

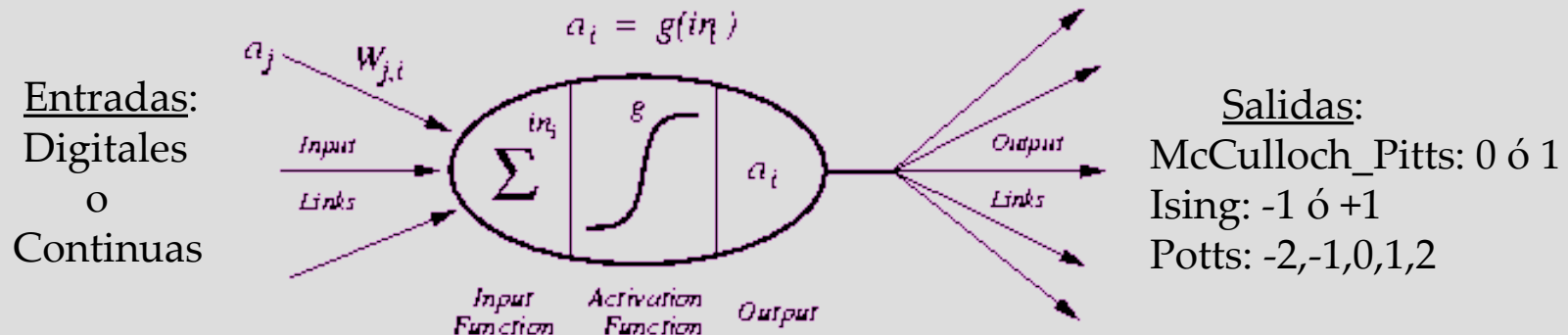
# Cerebro vs Computador

- Almacenamiento: Más neuronas que bits aunque la evolución computacional es vertiginosa (mucho mayor que la evolución de cerebro)
- Velocidad. Computador orden de los  $\eta$ seg  
Cerebro del orden de los mseg  
*pero ... el cerebro es masivamente paralelo y en definitiva el cerebro es  $10^{10}$  veces más rápido*
- Tolerancia a fallas: Una neurona natural dañada afecta de manera marginal el comportamiento del cerebro  
Cualquier mínimo error altera todo el procesamiento a nivel del computador
- Complejidad de ejecución: El cerebro realiza tareas muy complejas que son sencillas al humano pero difíciles para cualquier computador
- Procesamiento: Centralizado vs Distribuido  
*Computador Cerebro*

# Redes Neuronales Artificiales

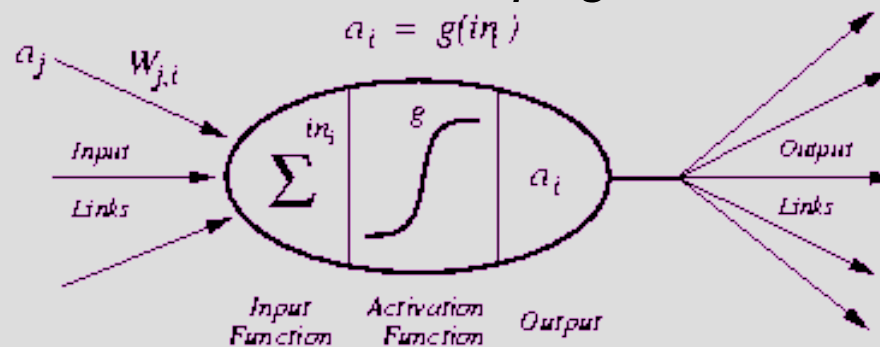
Unidades enlazadas a través de conexiones cargadas por pesos numéricos

- El aprendizaje se basa en la actualización de esos pesos que se inicializan en la fase de entrenamiento de la red
- Está formada por unidades de entrada y unidades de salida (neuronas de entrada y neuronas de salida)
- El nivel de activación de la neurona artificial (equivalente al impulso excitatorio) es un cálculo individual en cada neurona, sin control global



# Componentes de la Neurona

## Función de Propagación



- Suma Ponderada

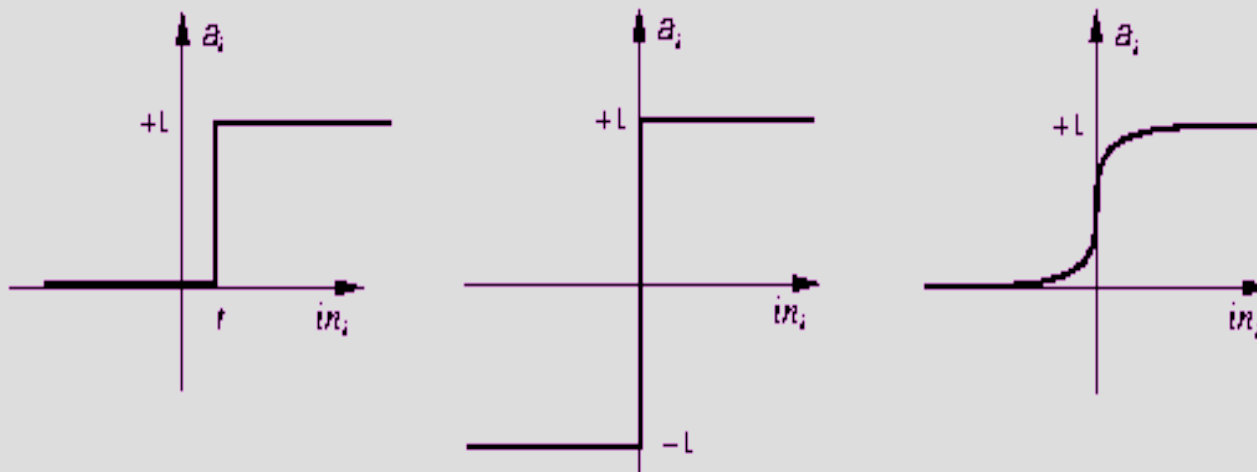
$$in_i = \sum W_{j,i} a_j$$

- Distancia Eucladiana

$$in_i = \sum a_j - W_{j,i}$$

- Manhattan, Sigma-Pi, ...

## Funciones de Activación



- Función de salida normalmente es la identidad aunque puede ser Estocástica

- La activación de toda la red puede ser síncrona o asíncrona

(a) Step function

(b) Sign function

(c) Sigmoid function

# Condiciones de *Estabilización*

- Las redes unidireccionales no tiene problemas de estabilidad pero Las retroalimentadas si ... deben cumplir ciertas condiciones para Converger a un estado estable o punto fijo ...
- Condiciones y función de Lyapunov:
  - a) El sistema está en reposo sólo en el origen
  - b) Existen variables de entrada que describen todo el dominio
  - c) las variables están acotadas

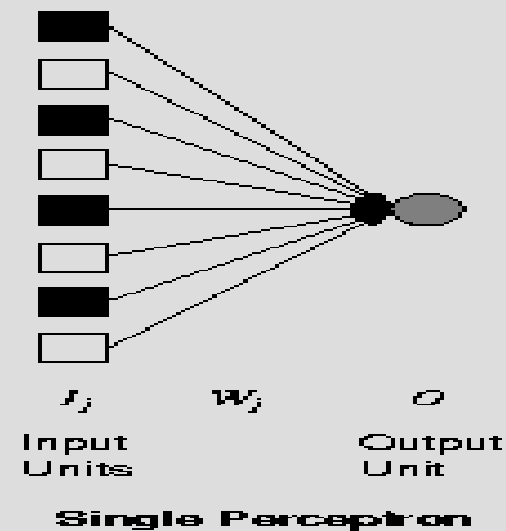
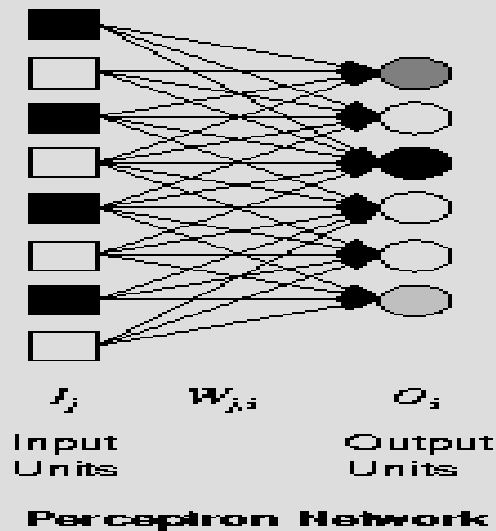
Sean  $V$  las variables  $x_i$  de entrada  $V: \mathbb{R}^n \rightarrow \mathbb{R}$

$$\dot{V} = \sum_{i=1}^n dV / dx_i \leq 0, \text{ para todo } x_i$$

- Además computacionalmente como una red neural puede representar un NAND el cual a su vez puede representar cualquier función booleana significa que son capaces de representar cualquier problema

# Perceptrón (feed forward)

Red neural lineal a dos capas  
(sólo neuronas de entrada y salida)

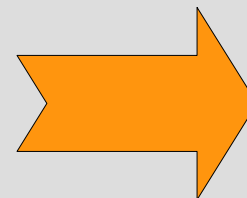


El perceptrón aprende comenzando con pesos aleatorios ajustandolos mientras se entrena  
(sencillo pues las neuronas de entrada van conectadas directamente con las de salida)

$$Err = T - O$$

Si  $Err > 0$  aumentar  $O$  sino disminuir  $O$

$$W_j = W_j + \alpha \cdot I_j \cdot Err$$



$O$  : ejemplo predicho

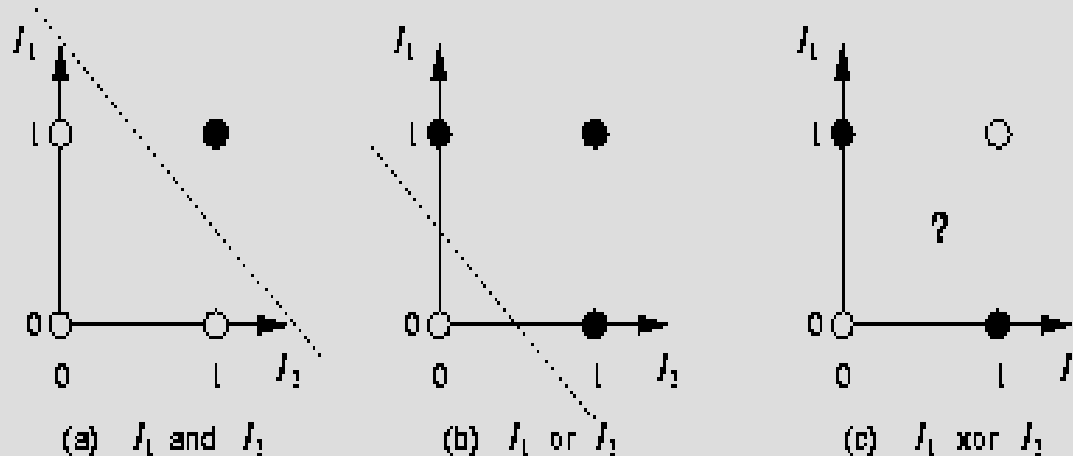
$T$  : ejemplo correcto

$\alpha$  : velocidad de aprendizaje

# Problemas del Perceptrón

Minsky y Papert publicaron en 1969, un artículo donde mostraron las limitaciones de los perceptrones

El problema está en que el perceptrón sólo puede representar funciones linealmente separables ya que el perceptrón es una función lineal de las neuronas de entrada



Las funciones linealmente separables son muy escasas y además, según Minsky y Papert, aún las redes neurales multicapas no resuelven el problema pues son una extensión del perceptrón

# Redes Neuronales Multicapas

Bryson y Ho publicaron también en 1969, un artículo sobre la retropropagación (*back propagation*) que valorizaba el uso de las redes neuronales multicapas

*No obstante sus trabajos no fueron tomados en cuenta y no fueron considerados sino hasta 1980 con el resurgimiento de las Redes Neuronales*

- Por cada neurona en una capa oculta se genera un hiperplano
- La intersección de cada uno de ellos pueda producir crestas y hasta montículos
- Con dos capas ocultas se puede generar hasta una función discontinua



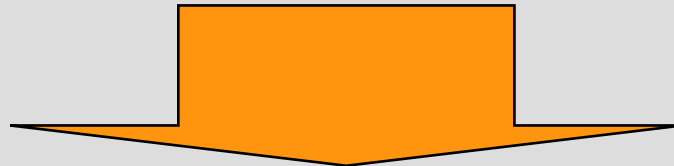
# Retropropagación (Backpropagation)

El problema radica en como ajustar los pesos de las neuronas intermedias mientras se está en fase de entrenamiento

La idea es que la neurona oculta  $j$  es responsable de alguna fracción proporcional del error  $\Delta_i$

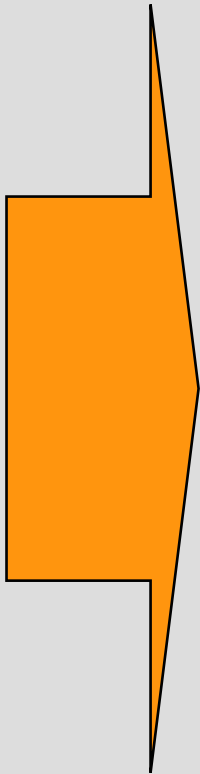
$$W_{j,i} = W_{j,i} + \alpha \cdot a_j \cdot \text{Err}_i \cdot g(in_i) \quad \text{donde } \Delta_i = \text{Err}_i \cdot g(in_i)$$

La fórmula propaga hacia atrás, capa por capa, hasta las neuronas de entrada



Este método también tiene sus limitaciones pues está demostrado que es una tarea intratable (NP-completo)

# Consideraciones de una Redes Neuronales Artificial



- ¿ Cuantas unidades o neuronas artificiales ?  
*Tanteo ...*
- ¿ Tipo de neurona ?  
*Por problemas similares*
- ¿ Topología de la red ?  
*Tanteo ...*
- ¿ Inicialización de los pesos ?  
*Aleatorio*
- ¿ Número de ejemplos para el entrenamiento ?  
*Depende de la sobrecompensación ...*
- ¿ Cómo codificar los datos de entrada y salida ?  
*Binario es lo común ...*

# Limitaciones de las Redes Neuronales Artificiales

- ¿ Cuántas capas y neuronas se deben considerar en un diseño ?



Se hace empíricamente lo cual es muy cuestionable desde el punto de vista científico

- El tiempo de aprendizaje crece exponencialmente
- No hay transparencia pues usa enfoque caja negra que impiden saber con certeza como trabaja la red neural una vez entrenada



No tienen capacidad de explicación

- Conocimiento a priori no puede ser bien aprovechado

# Artículos de Ejemplos Usando Informática Evolutiva

- G. Garcia, A. Lujan, W. Pereira, R. Paladino, IBC: Individual Based Choice, The 2nd International Conference on Advanced Computer Theory and Engineering (ICACTE 2009), Egypt, Sept/2009.
- J. Rada, R. Parma y W. Pereira, Path Optimization for Multiple Objectives in Directed Graphs using Genetic Algorithms, IEEE World Congress on Computational Intelligence, 1-6 Jun/2008, Hong Kong, China.
- R. Parma, W. Pereira y J. Rada, Ant Colony Optimization to an Autonomous Multiagent Game, 10th International Conference on Computer Games: AI, Animation, Mobile, Educational & Serious Games, 25<sup>th</sup>-28th Jul/2007, Louisville, Kentucky, USA.
- G. García, W. Pereira y G. Ron, Strength By Objective: Una nueva estrategia de asignación de fitness para Algoritmos Evolutivos Multi-Objetivos, XXXI Conferencia Latinoamericana de Informática (CLEI2005), Octubre/2005, Cali, Colombia.
- A. Luján y W. Pereira, Desarrollo de un Jugador Artificial de GO Basado en Redes Neurales Evolutivas, 55<sup>ava</sup> Convención Anual de Avance para la Ciencia (ASOVAC2005), Nov/2005, UCV, Caracas.
- C. Leiva, G. Puma y W. Pereira, Sistema Autónomo Robótico para Intersectar Objetivos Móviles de Comportamiento Evasivo, 3<sup>eras</sup> Jornadas de Investigación UCAB, Nov/2005, Caracas.
- D. Dos Santos, R. Peñalver y W. Pereira, Autonomous Navigation Robotic System to Recognize Irregular Patterns, 1<sup>st</sup> International Conference on Informatics in Control, Automation and Robotics, August 25-28, 2004, Setúbal, Portugal.
- G. Loerincs, S. Zabala, W. Pereira , Identifying Languages using Support Vector Machines, XXV Conferencia Latinoamericana de Informática (PANEL99), Agosto/1999, La Asunción, Paraguay