

# Introducción a la Inteligencia Computacional

## Materia: Fundamentos de Inteligencia Artificial

Prof. D.Sc. BARSEKH-ONJI Aboud

Facultad de Ingeniería  
Universidad Anáhuac México

12 de enero de 2026



# Agenda

- 1 Inteligencia Artificial e Inteligencia Computacional
- 2 Ramas Principales de la Inteligencia Artificial
  - Aprendizaje Automático de Máquina
  - Aprendizaje Profundo
  - Modelos de Lenguaje y Procesamiento del Lenguaje Natural (NLP)
- 3 Tipos de Inteligencia Computacional
  - Redes Neuronales Artificiales
  - Computación Evolutiva
  - Lógica Difusa y Sistemas Difusos
  - Inteligencia de Enjambre
  - Sistemas Inmunes Artificiales

# Definiendo la Inteligencia

## Punto de Partida: La Inteligencia Humana

La inteligencia humana es un espectro de capacidades cognitivas: razonamiento, aprendizaje, resolución de problemas, percepción, lenguaje y creatividad. Trasladar esto a una máquina es un desafío complejo.

## Definiciones de Inteligencia Artificial (IA)

Existen diferentes enfoques para definir la IA:

- **Enfoque Humano:** Sistemas que piensan o actúan como humanos.
- **Enfoque Racional:** Sistemas que piensan o actúan racionalmente. El enfoque del **agente racional**, que toma la mejor decisión posible dada la información disponible, ha ganado gran tracción en la investigación.



# La Prueba de Turing (1950)

## Un Punto de Inflexión Histórico

Propuesta por Alan Turing, busca responder a la pregunta: ¿pueden pensar las máquinas?

- **El Juego:** Un evaluador humano se comunica por texto con dos entidades ocultas: una máquina y un humano.
- **El Veredicto:** Si el evaluador no puede distinguir de manera fiable cuál es la máquina, se dice que esta ha superado la prueba y exhibe un comportamiento inteligente.

## Críticas y Debate

Aunque es un referente cultural, la prueba ha sido muy debatida. Argumentos como la

# Inteligencia Computacional (IC)

## Definición

La IC es 'el estudio de mecanismos **adaptativos** para permitir o facilitar el comportamiento inteligente en entornos complejos y cambiantes'. El énfasis está en la **adaptación** y el **aprendizaje**.

## Habilidades Centrales de los Sistemas de IC

Estos sistemas demuestran una habilidad intrínseca para:

- Aprender de la experiencia.
- Adaptarse a nuevas situaciones.
- Generalizar el conocimiento a casos no vistos.

# IA vs. IC: Un Enfoque Distintivo

## Inteligencia Artificial (General)

Es un campo amplio que también abarca enfoques simbólicos, lógicos y basados en el conocimiento explícito, como los sistemas expertos clásicos.

## Inteligencia Computacional (Específico)

Es una rama de la IA con un enfoque distintivo en algoritmos a menudo **inspirados en la naturaleza**. Se centra en paradigmas como:

- Redes Neuronales Artificiales
- Computación Evolutiva
- Lógica Difusa
- Inteligencia de Enjambre

# Agenda

- 1 Inteligencia Artificial e Inteligencia Computacional
- 2 Ramas Principales de la Inteligencia Artificial
  - Aprendizaje Automático de Máquina
  - Aprendizaje Profundo
  - Modelos de Lenguaje y Procesamiento del Lenguaje Natural (NLP)
- 3 Tipos de Inteligencia Computacional
  - Redes Neuronales Artificiales
  - Computación Evolutiva
  - Lógica Difusa y Sistemas Difusos
  - Inteligencia de Enjambre
  - Sistemas Inmunes Artificiales

# Aprendizaje Automático de Máquina (Machine Learning)

## Definición

El Aprendizaje Automático (ML) es una rama de la IA cuyo objetivo es desarrollar modelos que permitan a los sistemas informáticos mejorar su rendimiento en una tarea a través de la experiencia, es decir, **aprendiendo de los datos** sin necesidad de ser programados explícitamente para cada detalle.

## Concepto Clave: Generalización

Es la capacidad de un modelo de ML para funcionar bien y hacer predicciones precisas con **datos no vistos previamente**. Es el objetivo crucial del proceso de aprendizaje.



# Principales Categorías de Aprendizaje Automático

# Aprendizaje Supervisado

El algoritmo aprende de datos **etiquetados**, donde cada entrada tiene una salida correcta conocida. El objetivo es predecir la salida para nuevas entradas.

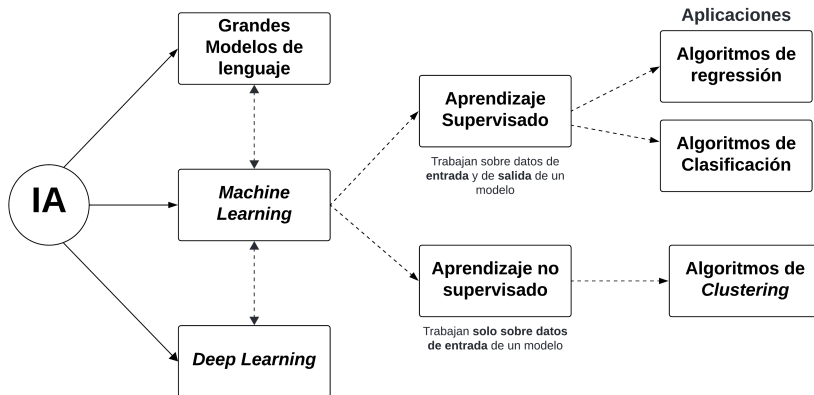
# Aprendizaje No Supervisado

El algoritmo trabaja con datos **sin etiquetar**. El objetivo es descubrir patrones, estructuras o agrupaciones inherentes en los datos.

## Aprendizaje por Refuerzo

Un agente aprende a tomar decisiones interactuando con un **entorno**. El objetivo es aprender una estrategia que maximice una **recompensa** acumulada.

# Principales Categorías de Aprendizaje Automático



# Aprendizaje Supervisado: Aprender con un Maestro

## Analogía

Es como enseñarle a un niño la diferencia entre un perro y un gato: le muestras fotos etiquetadas ('esto es un perro', 'esto es un gato') y supervisas su aprendizaje. El modelo aprende a mapear una entrada (imagen) a una salida (etiqueta).

## Tareas Típicas

- **Clasificación:** Predecir una etiqueta categórica.
  - Ej: 'spam' o 'no spam', tipo de animal en una foto.
- **Regresión:** Predecir un valor continuo.
  - Ej: precio de una casa, temperatura de mañana.

# Aprender por Prueba y Error

# Aprendizaje No Supervisado

El algoritmo explora datos sin etiquetar para encontrar patrones por sí mismo.

- **Agrupamiento (Clustering):**  
Descubrir grupos naturales en los datos (e.g., segmentar clientes por comportamiento de compra).
- **Reducción de Dimensionalidad:**  
Encontrar representaciones más compactas de los datos.

## Aprendizaje por Refuerzo

Un agente aprende interactuando con un entorno. No recibe 'respuestas correctas', sino señales de **recompensa** o **castigo** por sus acciones, buscando maximizar la recompensa a largo plazo.

- **Ejemplo Clásico:** Un programa que aprende a jugar ajedrez o Go.

# Aprendizaje Profundo (Deep Learning)

## Definición

El Aprendizaje Profundo (DL) es una rama del Aprendizaje Automático caracterizada por el uso de **Redes Neuronales Artificiales (ANN)** con múltiples capas de procesamiento, conocidas como arquitecturas 'profundas'.

## La Fuerza de la Profundidad: Aprendizaje Jerárquico

La 'profundidad' se refiere al número de capas a través de las cuales se transforman los datos. Esta estructura jerárquica permite que el modelo aprenda características complejas de forma incremental: las primeras capas aprenden rasgos simples (como bordes en una imagen) y las capas posteriores los combinan para formar conceptos más abstractos (como rostros).

El DL ha sido el motor de revoluciones en dominios como:

- **Visión por Computadora:** Clasificación de imágenes y detección de objetos con Redes Neuronales Convolucionales (CNNs).
- **Procesamiento del Lenguaje Natural (NLP):** Traducción y generación de texto con *Transformers*.
- **Reconocimiento de Voz.**
- **Juegos y Control:** Agentes con nivel sobrehumano en juegos como Go (AlphaGo).

# Aplicaciones y Desafíos del Aprendizaje Profundo

## El Gran Desafío: La 'Caja Negra'

A pesar de su efectividad, los modelos de DL a menudo son 'cajas negras' imposibles de interpretar. No podemos saber con certeza por qué toman una decisión específica.

Esta falta de interpretabilidad es un limitante en áreas críticas y ha dado origen a un nuevo campo de investigación: la **Inteligencia Artificial Explicable (XAI)**.

**Objetivo:** Permitir que las máquinas procesen, comprendan, generen y respondan al lenguaje humano de manera útil y significativa.

Pensemos en la función de autocompletar de nuestros teclados o editores de código.



- **LMs Tradicionales (n-gramas):** Capturan dependencias locales, pero fallan con el contexto a largo plazo.

# La Revolución del Aprendizaje Profundo en NLP

## Evolución de los Modelos de Lenguaje

- **LMs Tradicionales (n-gramas):** Capturan dependencias locales, pero fallan con el contexto a largo plazo.
- **Redes Neuronales Recurrentes (RNNs):** Primer gran avance para modelar secuencias y dependencias a más largo plazo.

# La Revolución del Aprendizaje Profundo en NLP

## Evolución de los Modelos de Lenguaje

- **LMs Tradicionales (n-gramas):** Capturan dependencias locales, pero fallan con el contexto a largo plazo.
- **Redes Neuronales Recurrentes (RNNs):** Primer gran avance para modelar secuencias y dependencias a más largo plazo.
- **Arquitectura Transformer:** Una revolución gracias a su mecanismo de 'auto-atención', que le permite ponderar la importancia de todas las palabras en una secuencia a la vez.

## Grandes Modelos de Lenguaje (LLMs)

# Usar un LLM vs. 'Hacer' Inteligencia Artificial

## Uso de LLMs

Estos modelos pueden ser ajustados para tareas específicas (*fine-tuning*) o utilizados directamente mediante **ingeniería de prompts** (*prompt engineering*), demostrando capacidades asombrosas en generación de texto, traducción, resumen y hasta programación.

## Una Distinción Importante

- **Usar un LLM:** Es ser un usuario de un producto final desarrollado con IA.
- **'Hacer' IA / Inteligencia Computacional:** Implica aplicar directamente un algoritmo (ML o DL) sobre una base de datos para crear un nuevo modelo inteligente. Significa tener contacto directo tanto con los datos como con los



# Agenda

- 1 Inteligencia Artificial e Inteligencia Computacional
- 2 Ramas Principales de la Inteligencia Artificial
  - Aprendizaje Automático de Máquina
  - Aprendizaje Profundo
  - Modelos de Lenguaje y Procesamiento del Lenguaje Natural (NLP)
- 3 Tipos de Inteligencia Computacional
  - Redes Neuronales Artificiales
  - Computación Evolutiva
  - Lógica Difusa y Sistemas Difusos
  - Inteligencia de Enjambre
  - Sistemas Inmunes Artificiales

# Paradigmas de la Inteligencia Computacional (IC)

## Enfoque Principal

La Inteligencia Computacional se distingue por su enfoque en paradigmas que simulan procesos o estrategias que existen en la **naturaleza** con la finalidad de lograr **adaptación y aprendizaje**.

## Características Comunes de los Algoritmos de IC

Todos los algoritmos que pertenecen a la IC son:

- **Iterativos:** Mejoran su solución a lo largo de múltiples pasos o 'generaciones'.
- **Optimizadores:** Buscan resolver un problema de optimización, como minimizar un error de predicción.

# Inspiración: El Cerebro Humano

## Concepto

Las Redes Neuronales Artificiales (ANNs) son uno de los pilares de la IC. Tratan de simular la red neuronal del cerebro humano, representando redes de neuronas artificiales interconectadas y organizadas en capas.

## Funcionamiento Básico de una Neurona

- 1 Recibe señales de entrada.
- 2 Calcula una suma ponderada de estas entradas.
- 3 Aplica una **función de activación**

## Proceso de Aprendizaje

El 'aprendizaje' consiste en un algoritmo iterativo que **ajusta los pesos** de las conexiones para minimizar el error entre la salida producida por la red y la salida

# Inspiración: La Evolución Biológica

## Concepto

La Computación Evolutiva es una familia de algoritmos de optimización que se inspiran directamente en la teoría de la evolución de Charles Darwin: **selección natural**, **herencia genética** y **mutación**.

## Funcionamiento General

Estos algoritmos operan sobre una '**población**' de posibles soluciones y buscan mejorar su calidad de forma iterativa a lo largo de múltiples '**generaciones**'.



# El Ciclo de un Algoritmo Evolutivo

- 1 **Población Inicial:** Se genera un conjunto inicial de posibles soluciones ('cromosomas'), a menudo de forma aleatoria.

# El Ciclo de un Algoritmo Evolutivo

- 1 **Población Inicial:** Se genera un conjunto inicial de posibles soluciones ('cromosomas'), a menudo de forma aleatoria.
- 2 **Evaluación de Aptitud (*Fitness*):** Cada solución es evaluada con una 'función de aptitud' que cuantifica qué tan buena es.

# El Ciclo de un Algoritmo Evolutivo

- 1 **Población Inicial:** Se genera un conjunto inicial de posibles soluciones ('cromosomas'), a menudo de forma aleatoria.
- 2 **Evaluación de Aptitud (*Fitness*):** Cada solución es evaluada con una 'función de aptitud' que cuantifica qué tan buena es.
- 3 **Selección:** Las mejores soluciones ('los más aptos') son seleccionadas para ser 'padres' de la siguiente generación.

# El Ciclo de un Algoritmo Evolutivo

- 1 **Población Inicial:** Se genera un conjunto inicial de posibles soluciones ('cromosomas'), a menudo de forma aleatoria.
- 2 **Evaluación de Aptitud (*Fitness*):** Cada solución es evaluada con una 'función de aptitud' que cuantifica qué tan buena es.
- 3 **Selección:** Las mejores soluciones ('los más aptos') son seleccionadas para ser 'padres' de la siguiente generación.
- 4 **Operadores Genéticos:** Se crean nuevas soluciones ('descendencia') combinando a los padres (**cruce**) e introduciendo pequeñas variaciones (**mutación**).

- 1 **Población Inicial:** Se genera un conjunto inicial de posibles soluciones ('cromosomas'), a menudo de forma aleatoria.
- 2 **Evaluación de Aptitud (*Fitness*):** Cada solución es evaluada con una 'función de aptitud' que cuantifica qué tan buena es.
- 3 **Selección:** Las mejores soluciones ('los más aptos') son seleccionadas para ser 'padres' de la siguiente generación.
- 4 **Operadores Genéticos:** Se crean nuevas soluciones ('descendencia') combinando a los padres (**cruce**) e introduciendo pequeñas variaciones (**mutación**).
- 5 **Reemplazo:** La nueva generación de soluciones reemplaza a la anterior.

# El Ciclo de un Algoritmo Evolutivo

- 1 **Población Inicial:** Se genera un conjunto inicial de posibles soluciones ('cromosomas'), a menudo de forma aleatoria.
- 2 **Evaluación de Aptitud (*Fitness*):** Cada solución es evaluada con una 'función de aptitud' que cuantifica qué tan buena es.
- 3 **Selección:** Las mejores soluciones ('los más aptos') son seleccionadas para ser 'padres' de la siguiente generación.
- 4 **Operadores Genéticos:** Se crean nuevas soluciones ('descendencia') combinando a los padres (**cruce**) e introduciendo pequeñas variaciones (**mutación**).
- 5 **Reemplazo:** La nueva generación de soluciones reemplaza a la anterior.
- 6 El ciclo se repite hasta que se cumple un criterio de terminación (e.g., número de generaciones, calidad de la solución).

# Principales Paradigmas de la Computación Evolutiva

- **Algoritmos Genéticos (GA):** Típicamente utilizan representaciones binarias de las soluciones y enfatizan el operador de cruce.

# Principales Paradigmas de la Computación Evolutiva

- **Algoritmos Genéticos (GA):** Típicamente utilizan representaciones binarias de las soluciones y enfatizan el operador de cruce.
- **Programación Genética (GP):** Evolucionan directamente programas de computadora o expresiones matemáticas, a menudo representados como árboles.



# Principales Paradigmas de la Computación Evolutiva

- **Algoritmos Genéticos (GA):** Típicamente utilizan representaciones binarias de las soluciones y enfatizan el operador de cruce.
- **Programación Genética (GP):** Evolucionan directamente programas de computadora o expresiones matemáticas, a menudo representados como árboles.
- **Estrategias Evolutivas (ES):** Se centran en la optimización de parámetros de valor real y a menudo auto-adaptan sus propias estrategias de mutación.

# Principales Paradigmas de la Computación Evolutiva

- **Algoritmos Genéticos (GA):** Típicamente utilizan representaciones binarias de las soluciones y enfatizan el operador de cruce.
- **Programación Genética (GP):** Evolucionan directamente programas de computadora o expresiones matemáticas, a menudo representados como árboles.
- **Estrategias Evolutivas (ES):** Se centran en la optimización de parámetros de valor real y a menudo auto-adaptan sus propias estrategias de mutación.
- **Programación Evolutiva (EP):** Pone un mayor énfasis en la mutación como principal motor de la evolución, modelando la evolución a nivel de especie.

# Lógica Difusa: Más Allá del Verdadero y Falso

## Lógica Clásica (Booleana)

- Opera con proposiciones que son estrictamente **verdaderas** o **falsas**.
- Utiliza valores binarios (0 o 1).

## Lógica Difusa (Fuzzy Logic)

- Permite hablar de '**grados de verdad**'.
- Una proposición puede ser parcialmente verdadera.
- Trata con la imprecisión y vaguedad del razonamiento humano.

Concepto Central: El Conjunto Difuso

# Sistemas de Inferencia Difusa

## ¿Cómo funciona un Sistema Difuso?

Utilizan reglas lingüísticas para modelar sistemas y tomar decisiones. Constan de cuatro componentes:

- 1 **Fuzificación:** Convierte las entradas numéricas ('nítidas') en grados de pertenencia a conjuntos difusos.

# Sistemas de Inferencia Difusa

## ¿Cómo funciona un Sistema Difuso?

Utilizan reglas lingüísticas para modelar sistemas y tomar decisiones. Constan de cuatro componentes:

- 1 **Fuzificación:** Convierte las entradas numéricas ('nítidas') en grados de pertenencia a conjuntos difusos.
- 2 **Base de Reglas Difusas:** Un conjunto de reglas del tipo '**SI-ENTONCES**'. Por ejemplo: 'SI la temperatura ES alta ENTONCES la velocidad del ventilador ES rápida'.

# Sistemas de Inferencia Difusa

## ¿Cómo funciona un Sistema Difuso?

Utilizan reglas lingüísticas para modelar sistemas y tomar decisiones. Constan de cuatro componentes:

- 1 **Fuzificación:** Convierte las entradas numéricas ('nítidas') en grados de pertenencia a conjuntos difusos.
- 2 **Base de Reglas Difusas:** Un conjunto de reglas del tipo '**SI-ENTONCES**'. Por ejemplo: 'SI la temperatura ES alta ENTONCES la velocidad del ventilador ES rápida'.
- 3 **Motor de Inferencia:** Combina las reglas para derivar las conclusiones difusas.

# Sistemas de Inferencia Difusa

## ¿Cómo funciona un Sistema Difuso?

Utilizan reglas lingüísticas para modelar sistemas y tomar decisiones. Constan de cuatro componentes:

- 1 **Fuzificación:** Convierte las entradas numéricas ('nítidas') en grados de pertenencia a conjuntos difusos.
- 2 **Base de Reglas Difusas:** Un conjunto de reglas del tipo '**SI-ENTONCES**'. Por ejemplo: 'SI la temperatura ES alta ENTONCES la velocidad del ventilador ES rápida'.
- 3 **Motor de Inferencia:** Combina las reglas para derivar las conclusiones difusas.
- 4 **Defuzificación:** Convierte la conclusión difusa de nuevo en una salida numérica y accionable.

# Inspiración: Comportamiento Colectivo

## Concepto

Es un paradigma de la IC que se inspira en el comportamiento colectivo de sistemas naturales como colonias de hormigas, bandadas de pájaros o bancos de peces.

## Funcionamiento

Estos sistemas están compuestos por un gran número de agentes individuales relativamente simples que siguen reglas básicas. De sus interacciones locales emerge un comportamiento global inteligente y autoorganizado, sin necesidad de un control centralizado.



# Características Fundamentales de los Enjambres

- **Control descentralizado:** No hay una entidad única que dirija el comportamiento del enjambre.

## Características Fundamentales de los Enjambres

- **Control descentralizado:** No hay una entidad única que dirija el comportamiento del enjambre.
- **Autoorganización:** Los patrones y estructuras globales surgen espontáneamente de las interacciones locales entre agentes.

## Características Fundamentales de los Enjambres

- **Control descentralizado:** No hay una entidad única que dirija el comportamiento del enjambre.
- **Autoorganización:** Los patrones y estructuras globales surgen espontáneamente de las interacciones locales entre agentes.
- **Robustez:** El sistema puede tolerar fallos de algunos individuos sin que el rendimiento global se vea afectado significativamente.

## Características Fundamentales de los Enjambres

- **Control descentralizado:** No hay una entidad única que dirija el comportamiento del enjambre.
- **Autoorganización:** Los patrones y estructuras globales surgen espontáneamente de las interacciones locales entre agentes.
- **Robustez:** El sistema puede tolerar fallos de algunos individuos sin que el rendimiento global se vea afectado significativamente.
- **Escalabilidad:** El comportamiento del sistema puede mantenerse efectivo incluso si aumenta el número de agentes.

- **Control descentralizado:** No hay una entidad única que dirija el comportamiento del enjambre.
- **Autoorganización:** Los patrones y estructuras globales surgen espontáneamente de las interacciones locales entre agentes.
- **Robustez:** El sistema puede tolerar fallos de algunos individuos sin que el rendimiento global se vea afectado significativamente.
- **Escalabilidad:** El comportamiento del sistema puede mantenerse efectivo incluso si aumenta el número de agentes.
- **Comunicación indirecta:** Los agentes a menudo interactúan modificando su entorno (e.g., el rastro de feromonas de las hormigas), y estas modificaciones sirven como señales para otros.

# Algoritmos Prominentes de Inteligencia de Enjambre

## Optimización por Enjambre de Partículas (PSO)

- **Inspiración:** El movimiento coordinado de las bandadas de pájaros o bancos de peces buscando alimento.
- **Mecanismo:** Una población de 'partículas' (soluciones) 'vuela' a través del espacio de búsqueda, ajustando su trayectoria en base a su

## Optimización por Colonia de Hormigas (ACO)

- **Inspiración:** El comportamiento de las hormigas para encontrar los caminos más cortos hacia una fuente de alimento.
- **Mecanismo:** 'Hormigas' artificiales construyen soluciones y depositan 'feromona' en los caminos que eligen. Los caminos con más

# Inspiración: El Sistema Inmunológico

## Concepto

Los Sistemas Inmunes Artificiales (AIS) son una clase de sistemas computacionales inspirados en los principios y mecanismos del sistema inmunológico de los vertebrados.

## Capacidades del Sistema Inmunológico a Emular

El sistema inmunológico natural es un sistema complejo, distribuido y altamente adaptativo con notables capacidades de:

- Aprendizaje y memoria.
- Reconocimiento de patrones (distinguiendo entre 'lo propio' y 'lo no propio').
- Autoorganización y robustez.

# Conceptos que Inspiran Algoritmos

- **Selección Clonal:** Clonar y mutar las mejores soluciones (para optimización).
- **Selección Negativa:** Generar detectores que reconozcan patrones anómalos o 'no propios' (para detección de anomalías).
- **Redes Inmunes:** Modelar el sistema como una red auto-regulada (para clustering).
- **Teoría del Peligro:** Reaccionar a señales de 'peligro' en lugar de solo a lo 'no propio' (para detección de intrusiones).






# Conceptos Inmunológicos Clave y Aplicaciones

## Aplicaciones

- Detección de intrusiones en redes informáticas.
- Detección de virus y spam.
- Clasificación de datos.
- Optimización de funciones.
- Robótica.

# Referencias I

-  Bishop, C. M. (2006).  
*Pattern Recognition and Machine Learning*.  
Springer.
-  Engelbrecht, A. P. (2007).  
*Computational Intelligence: An Introduction*.  
John Wiley & Sons Ltd., 2nd edition.
-  Haykin, S. O. (2009).  
*Neural Networks and Learning Machines*.  
Pearson Education, 3rd edition.

## Referencias II



Lantz, B. (2019).

*Machine Learning with R: Expert techniques for predictive modeling.*  
Packt Publishing, 3rd edition.



Michalewicz, Z. (1996).

*Genetic Algorithms + Data Structures = Evolutionary Programs.*  
Springer, Berlin, third edition.



Zadeh, L. A. (1965).

Fuzzy sets.

*Information and Control*, 8(3):338–353.