



Facultad de Ciencias

Licenciatura en
Ciencias de la Computación

Cómputo Evolutivo

.....

Grupo 7108

Profesor: Oscar Hernández Constantino

`constantino92@ciencias.unam.mx`

Ayudante: Malinali González Lara

`malinali_glezl@ciencias.unam.mx`

Contenido de la Presentación

1. Contexto y Motivación
2. Aplicaciones
3. Dinámica del curso
 - 3.1 Temario
 - 3.1.1 Bibliografía
 - 3.2 Criterios de Evaluación
 - 3.3 Consideraciones Generales

Contexto y Motivación

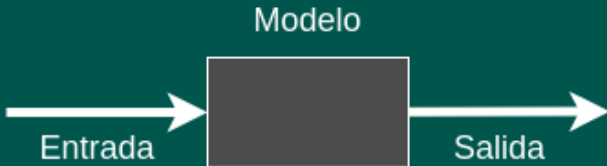
Contexto

- Ciencias de la Computación
 - > Inteligencia Artificial
 - Heurísticas
 - > Metaheurísticas

» **Cómputo Evolutivo**

Modelo de la Caja Negra

De manera informal, podemos ver a una computadora como el siguiente sistema:

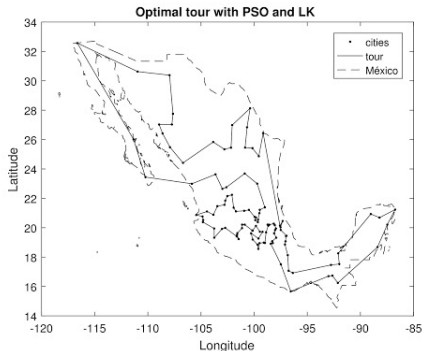


Dependiendo de cuál de estos tres componentes es desconocido, tenemos diferentes tipos problemas.

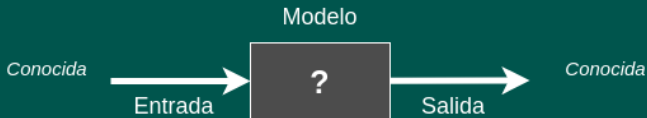
Problemas de Optimización



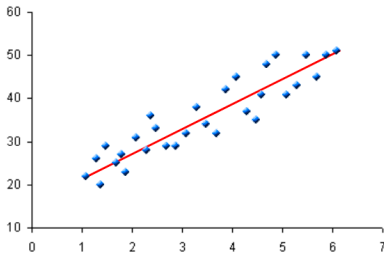
Ejemplo: TSP (Traveling Salesman problem)



Problemas de Modelación



Ejemplo: Regresión Lineal



Problemas de Simulación



Ejemplo: Simulador de Vuelo





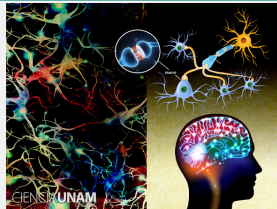
¿Por qué cómputo evolutivo?

Los científicos se han inspirado en las soluciones de la naturaleza.

Cerebro Humano

Ha creado la “rueda”, grandes ciudades, guerras, entre cosas cosa.

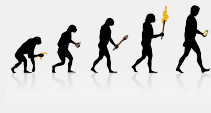
Esto da lugar a las redes neuronales artificiales.



Proceso Evolutivo

Ha creado el cerebro humano.

Es la base para el cómputo evolutivo.



Métafora

Consideraremos una simplificación de la evolución natural.

Evolución Biológica		Cómputo Evolutivo
Ambiente	↔	Problema de optimización
Evolución	↔	Solución de problemas
Individuo	↔	Solución candidata
Aptitud	↔	Función objetivo

Ventajas Cómputo Evolutivo

Los algoritmos son:

- simples y flexibles,
- aplicables a diferentes dominios (representaciones)
- altamente paralelizables
- robustos y adaptables a situaciones dinámicas

No se requiere conocimiento previo, pero sí existe se puede utilizar.

Desventajas Cómputo Evolutivo

- No se garantiza obtener una solución óptima
- Se necesita un proceso adicional para ajustar parámetros
- Podrían requerir muchos recursos computacionales

Aplicaciones

Aplicaciones

Las aplicaciones del cómputo evolutivo caen dentro de un amplio rango de áreas.

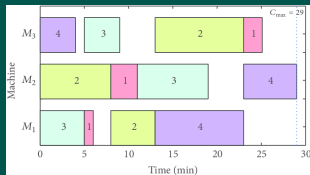
- Planificación
- Diseño
- Simulación
- Control
- Aprendizaje Máquina

Estas categorías no son absolutas o definitivas.

Aplicaciones - Planificación II

- **Calendarización**

Ejemplo: Planificación de tareas (JSSP, Job Shop Scheduling Problem)



- **Empacado**

Ejemplo: Problema de la Mochila (KP, Knapsack Problem)



Aplicaciones - Planificación III

- Planificación de horarios

Ciencias de la Computación (plan 2013) 2021-1

Navegación: Ciencias » Horarios 2021-1 » Ciencias de la Computación

Buscar en grupos

CARRERAS

Primer Semestre

Álgebra Superior I, 25 grupos
Estructuras Discretas, 5 grupos
Inglés I, 4 grupos
Introducción a Ciencias de la Computación, 6 grupos
Matemáticas para las Ciencias Aplicadas I, 4 grupos

MATERIAS

GRUPOS

Segundo Semestre

Álgebra Superior II, 9 grupos
Estructuras de Datos, 2 grupos
Gráficas y Juegos, 5 grupos
Inglés II, un grupo
Matemáticas para las Ciencias Aplicadas II, 6 grupos

SALONES / LABORATORIOS (Recursos)

PROFESORES

ALUMNOS

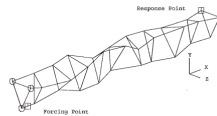
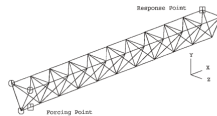
Tercer Semestre

Álgebra Lineal I, 25 grupos

Aplicaciones - Diseño

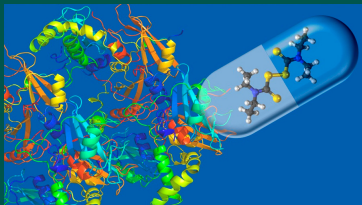
Optimización en diseños industriales

Ejemplo: Estructura para un satélite

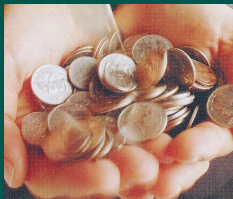


Aplicaciones - Simulación

- Aplicaciones en **Química y Biología**. Ejemplo: Diseño de Fármacos



- Aplicaciones a la **Economía**. Modelar interacciones en la bolsa.



Aplicaciones - Control

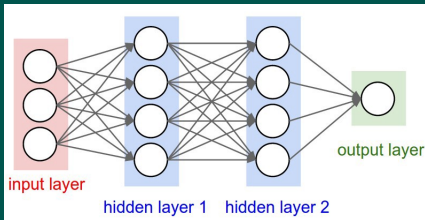
Diseño de controladores en robots

Ejemplo: Optimización de movimientos

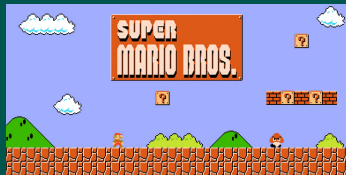


Aplicaciones - Máquina

- Aprendizaje Máquina



- En general, en problemas de Optimización



Dinámica del curso

Requisitos

- Programación
- Estructuras de Datos
- Fundamentos de Probabilidad y Estadísticas
- Análisis de Algoritmos
- Complejidad Computacional

Dinámica del curso

- **Curso en modalidad virtual.** Sesiones por videoconferencia (Zoom).
- Utilizaremos Classroom para el seguimiento del curso:
 - PDFs de las presentaciones
 - Lecturas / Artículos
 - Códigos de ejemplo. Lenguaje de programación: C (C++), Java, Python
 - Tareas, Proyectos
- Ayudantía: sesiones de dudas, ayuda para las tareas, material de apoyo, etc.
- La asistencia es importante, será tomada en cuenta para redondeo de calificación o como participación extra.

Temario I

1. Introducción al Cómputo Evolutivo

- 1.1 Breve historia del Cómputo Evolutivo
- 1.2 Problemas de Optimización
- 1.3 Metaheurísticas y su clasificación

2. Metaheurísticas de Trayectoria

- 2.1 Búsquedas Locales
- 2.2 Recocido Simulado
- 2.3 Búsqueda Tabú
- 2.4 Otras búsquedas

3. Metaheurísticas Poblacionales

- 3.1 Algoritmo Genético
- 3.2 Operadores
 - 3.2.1 Estrategias de Selección
 - 3.2.2 Cruza

Temario II

- 3.2.3 Mutación
 - 3.2.4 Reemplazo
- 3.3 Teorema de los Esquemas (Holland)
- 4. Evaluación de Rendimiento
 - 4.1 Medidas de Desempeño
 - 4.2 Ajuste de Parámetros
- 5. Otros Paradigmas de Cómputo Evolutivo
 - 5.1 Programación Evolutiva
 - 5.2 Estrategias Evolutivas
 - 5.2.1 Evolución Diferencial
 - 5.2.2 CMA
 - 5.3 Algoritmos de Estimación de Distribuciones
 - 5.4 Algoritmos Bio-inspirados
 - 5.4.1 PSO (Enjambre de Partículas)

Temario III

5.4.2 ACO (Colonia de Hormigas)

6. Métodos Híbridos

6.1 Algoritmo Memético (Genético + Búsqueda Local)

6.2 Encadenamiento de Rutas (Path Relinking)

7. Temas avanzados

7.1 Algoritmos evolutivos multi-objetivo

7.2 Optimización con restricciones

7.3 Algoritmos evolutivos paralelos

Bibliografía I

- [1] Carlos Coello.
Notas del curso introducción a la computación evolutiva, 2020.
- [2] Agoston E Eiben and James E Smith.
Introduction to evolutionary computing.
Springer, 2015.
- [3] Zbigniew Michalewicz and David B Fogel.
How to solve it: modern heuristics.
Springer Science & Business Media, 2013.
- [4] Dan Simon.
Evolutionary optimization algorithms.
John Wiley & Sons, 2013.

Bibliografía II

[5] El-Ghazali Talbi.

Metaheuristics: from design to implementation, volume 74.

John Wiley & Sons, 2009.

Criterios de Evaluación I

- **Tareas [65 %]**

Se deberán enviar siguiendo los lineamientos de entrega.
Deberán entregar un reporte y el código desarrollado.

- **Proyecto Final [35 %]**

Se puede elegir alguno de los sugeridos, o bien pueden proponer alguno. Es válido combinarlo con el proyecto de otra materia. Deberán hacer una exposición breve de su proyecto.

Consideraciones

- Habrá tareas individuales, y algunas podrán ser en equipos. El máximo número de integrantes por equipo depende del número de inscritos.
- En todos los rubros (tareas, proyectos) se podrá tener una evaluación con réplica oral. Si no asisten, pierden derecho a réplica y se quedan con la calificación que sea asignada.
- Tareas/proyectos copiados tendrán una calificación de cero la primera vez. La segunda incidencia implica calificación final de 5.
- Dada la naturaleza del curso y la forma de evaluación, NO habrá examen final.

Consideraciones

- Las entregas tardías se penalizarán con 1 punto por cada día de retraso. Después de 3 días naturales, no se aceptan entregas.
- En todas las tareas / prácticas se debe entregar un **reporte en pdf**.
- En todas las entregas (tareas / prácticas, proyectos) se debe indicar siempre, al menos, una referencia (diferente a wikipedia).