



Algoritmos Genéticos

Héctor Alejandro Montes

hamontesv@uaemex.mx

<http://scfi.uaemex.mx/hamontes>

Evaluación del curso

- Ordinaria
 - Tareas 30%
 - Exámenes parciales 30%
 - Proyecto final 40%
- Extraordinaria y a Título de Suficiencia
 - Examen teórico 50%
 - Examen práctico 50%
- Prohibido el **plagio** de todo tipo

Conocimientos previos

- Programador competente en al menos un lenguaje
 - C, C++, C#, Java, Visual Basic, Objective C, Python, o cualquier otro que sea su favorito
- Matemáticas
 - Nivel de Ingeniería
 - Probabilidad y Estadística
 - Análisis básico de datos
- Análisis de lecturas

Sentido común

- Asistencia según legislación
- Comportamiento de sentido común
 - No fumar, no comer, *no usar celulares*, etc.
- Evitar excusas sin sentido
 - “tengo trabajo”, “te lo mandé”, “no recibí tú correo”, “te busqué...”, “no tuve tiempo...”
- Entregar a tiempo
 - **NO HAY EXCEPCIONES NI EXCUSAS**

Contenido del curso

- Breve historia del Cómputo Evolutivo
 - Algoritmos evolutivos
 - Métodos heurísticos
- Terminología biológica
- Algunas aplicaciones exitosas
- Búsqueda y optimización
 - ¿Buscar u optimizar?
 - Búsqueda aleatoria
 - *Hill climbing*

Contenido del curso (2)

- Algoritmo general de un AG
 - El AG simple
- Elementos de un AG
 - Representación, Población inicial, Función de aptitud
 - Operadores genéticos, Parámetros de control
- Optimización de una función simple
- ¿Porqué y cómo funcionan los AGs?
- Espacios de búsqueda y *fitness landscapes*

Contenido del curso (3)

- Problemas tratables e intratables
 - Complejidad y clasificación de problemas
- El problema de satisfactibilidad
- El problema del agente viajero
- El problema de la transportación
- El problema de programación (scheduling)
- El problema de horarios
- Particionando objetos y grafos

Contenido del curso (4)

- Representaciones binarias
- Representaciones reales
- Representaciones de permutaciones
- Otros operadores de cruza y mutación
- Métodos de selección
- Ajuste de parámetros
- Generación de números aleatorios

Contenido del curso (5)

- Fundamentos teóricos de los AGs
 - Teorema de esquemas o *schemata*
 - Teorema *No free lunch*
- ¿Cuándo deberíamos utilizar los AGs?
- Incorporación de conocimiento específico al problema
- Manejo de restricciones
- Múltiples objetivos

Proyecto

- Durante el curso deberás completar un proyecto sobre un problema específico.
- Deberás concentrarte en desarrollar pruebas claras que deriven en evidencia suficiente para la solución del problema tratado.
- Deberás hacer énfasis en la **metodología** mas que en la **complejidad** del problema.
- Si ya estás involucrado en algún proyecto (tesis o proyecto de otro curso, por ejemplo), es **recomendable** que selecciones un problema que se relacione con ese proyecto.
- Entregar en la fecha establecida.

Un '*buen*' proyecto

- Para fines de este curso, un *buen* problema de estudio es aquel que satisface la mayoría de las siguientes condiciones:
 - Es fácil de establecer
 - Es un reto encontrar soluciones óptimas o cercanas al óptimo.
 - Tiene relevancia.
 - Ha sido tema de varios estudios previos.
 - Hay instancias del problema con soluciones óptimas conocidas.

Bibliografía

- David E. Goldberg. *Genetic Algorithms in Search, Optimisation and Machine Learning*. Addison-Wesley, 1989.
- Z. Michalewicz. *Genetic Algorithms + Data Structures = Evolution Programs (3rd edition)*. Springer-Verlag, 1996.
- Randy L. Haupt, S. E. Haupt. *Practical Genetic Algorithms*. Wiley-Interscience, 2004.
- Melanie Mitchell. *An Introduction to Genetic Algorithms*. MIT Press, 1996.
- K. A. De Jong. *Evolutionary Computation: A Unified Approach*. MIT Press, 2006.
- **Artículos** (*papers*), reportes técnicos, entre otros disponibles en Internet.