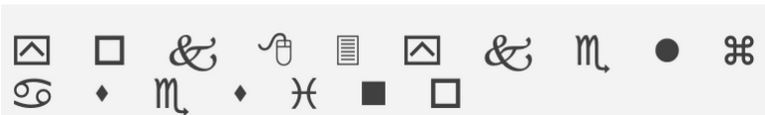




GEN - Interdisciplina: Genética

Experiencia

1969, California. El Asesino del Zodiaco a vuelto a actuar. Pero esta vez ha enviado un mensaje cifrado a la prensa, similar a este pero más largo:



Probar todas las posibles combinaciones de símbolos a letras tardaría millones de años. Por suerte acabas de atender a una clase sobre Algoritmos Genéticos, y vas a usarlos para descifrar el mensaje.

Competencias

Específicas:

- Conceptos básicos de los algoritmos genéticos tales como Feasiblity, Fitness, Crossover, o Mutation

Transversales:

- Conocer un paradigma totalmente diferente a los habituales (Algoritmos Genéticos)

BKNs:

- ¡Algoritmos basados en la naturaleza!



Material

El problema está en las slides y se ha explicado en clase (si hay dudas se puede buscar a un compañero de aventuras en [DISCORD](#)). A continuación se agregan unas capturas:

Zodiac

Mensaje con m caracteres (ej. 7000), que usa n símbolos distintos (ej. 300)

El mismo símbolo se refiere siempre a la misma letra (o al espacio). Pero obviamente una misma letra (o el espacio) está representada por varios símbolos*

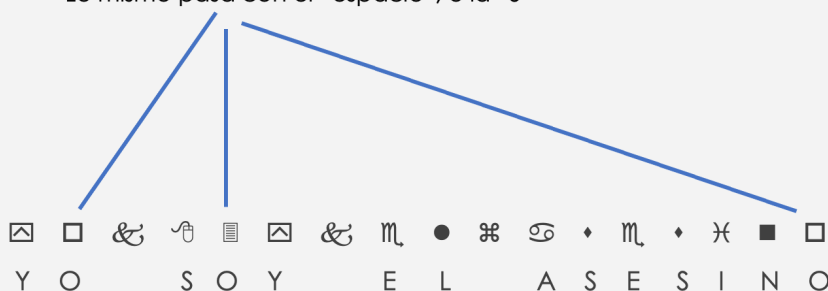
*Obvio, hay 300 símbolos pero solo hay 27 (26 letras + 1 espacio) en el alfabeto inglés



Tenemos una lista con todas las palabras del diccionario

Queremos encontrar una traducción de los símbolos a letras (+ espacio) tal que al decodificar el mensaje la mayoría de las palabras sean palabras reales del diccionario

el cuadrado vacío significa "O" pero no todas las "O" son un cuadrado. Lo mismo pasa con el "espacio", o la "S"



Las cuatro palabras que formo colocando los espacios son palabras del diccionario. Así que esta sería una solución perfecta.

Pueden haber nombres propios o errores del asesino. Así aunque no sea perfecta, preferimos soluciones con más número de palabras reales sobre otras con menos.



Entregable



Para entregar: Link en **CANVAS**

Se te pide un **documento** (preferiblemente electrónico como Word, pero también pueden ser fotos de papeles) donde detalles las 5 cosas siguientes sobre tu propuesta para solucionar el desafío con algoritmos genéticos:

Zodiac (Entregable)

1. Tu **Solution Representation**

- Asegúrate que tu propuesta puede representar todas las soluciones posibles

2. La función de **Feasibility**

- Con tu representación, que soluciones son válidas y cuales no.

3. La función de **Fitness**

- Calcular un valor de como de buena o mala es una solución.

4. El método de **Crossover**

- Cómo dos individuos se pueden cruzar para crear un descendiente
- Asegúrate que tu crossover intente dar siempre descendientes válidos

5. El método de **Mutation**

- Como se puede agregar un pequeño cambio al azar a tu individuo
- Intenta que tu mutación intente dar siempre descendientes válidos



Ejemplo 1

Ejemplo con el problema de la "Mochila de Supervivencia" (para no spoilear el de Zodiac)

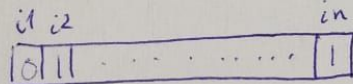
MOCHILA DE SUPERVIVENCIA

Problema: n items

cada item tiene un peso y unos "puntos de supervivencia"
la mochila peso máximo

Objetivo: mejor solución máximo puntos de supervivencia
sin romper la mochila

Representación
de la solución

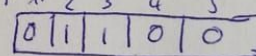


lista de largo n (n de items)
con 0 (no llevo el objeto)
1 (sí llevo el objeto)

item	peso	puntos
1	10	5
2	30	10
3	40	25
4	20	25
5	20	20

max
 ≤ 40

ejemplo:



me llevo el item 2 y
el item 3

Feasability

Solo tengo que comprobar que
no se rompa la bolsa.

→ sumo los pesos de los items
con 1 y no puede superar al max.

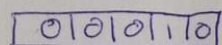
1	2	3	4	5
0	1	1	0	0

no valida porque
peso
 $30 + 40 \geq 40$

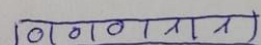
Fitness

Cuanto mas puntos de supervivencia, mejor
la solución. Sumo los puntos

0 0 1 0 1 1 0 valida pg
 $20 \leq 40$

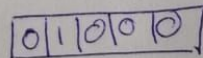


→ 25 puntos

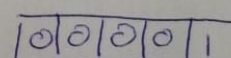


→ 45 puntos

mejor la sigue



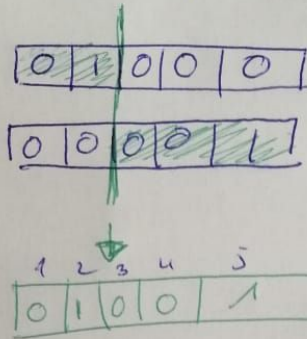
→ 10 puntos



→ 20 puntos

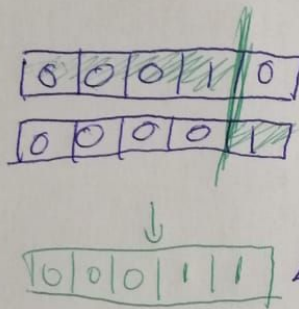
la segunda es el doble
de buena que la primera.
pero las dos son peores que las 2
anteriores

Crossover: Corto por un punto ^(al azar) y fusiono las partes.



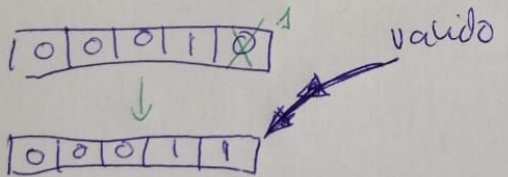
este crossover garantiza que tiene la forma correcta siempre pero no que sea valida por que se puede pasar el peso

$$30 + 20 > 40 \quad \text{max} \\ (\text{no valida})$$

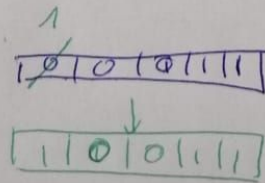


valida por que peso $20 + 20 \leq 40$ max

Mutacion: un numero al azar pasa de 0 \rightarrow 1 (o de 1 \rightarrow 0)



esta mutacion no garantiza que el resultado sea valido
de 0 a 1
(al mutar quizas sobrepasa el peso de 1 a 0 no pasa nada)



no valido
peso $10 + 20 + 20 > 40$ max



Rúbrica

Default view

Aa ID	# %	Descripción
<u>R1</u>	20%	Tu Solution Represent Asegúrate que tu propo todas las soluciones po
<u>R2</u>	20%	La función de Feasibil Con tu representación, válidas y cuales no.
<u>R3</u>	20%	La función de Fitness Calcular un valor de có una solución.
<u>R4</u>	20%	El método de Crossover - Cómo dos individuos crear un descendiente. - Asegúrate que tu cro (que se pueda) descen
<u>R5</u>	20%	El método de Mutation - Como se puede agre azar a tu individuo - Intenta que tu mutaci (que se pueda) descen
SUM		100%






Do It Yourself (DIY)



Para evaluar: Recuerda para que te evalúen los **DIY** y te den la nota bonus correspondiente, debes compartirlas en **DISCORD** siguiendo las instrucciones en **CANVAS**. ¡Así motivarás a los demás y podrás buscar sinergias!

 Default view

 Num	 Nivel	 DYI
<u>1</u>	Casual / Pro	Ver algún video donde
<u>2</u>	Casual / Pro	Jugar con alguna librería (PyGAD)
<u>3</u>	Casual / Pro	Generar un código tipo <u>desencriptar usando A</u>
<u>4</u>	Casual / Pro	Aprender de otra técnica (Colony Optimization)
<u>5</u>	Casual / Pro	Ver algunos videos donde <u>/ Evolutionary para resolver de auto en circuito).</u>
<u>6</u>	Casual / Pro	Investigar sobre que son los <u>problemas</u>
<u>7</u>	Casual / Pro	Estudiar que son los <u>problemas</u> <u>verificar</u> y NP (difíciles <u>verificar</u>). Aviso, no es
...		... y cualquier otra idea

Exploración



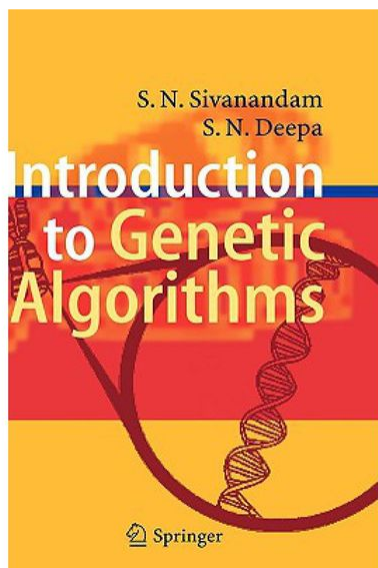
IIC3242 Complejidad Computacional

Programa en la **CARPETA**

- Entender la complejidad de un problema
- Conocer las características de los problemas NP
- Determinar que complejidad tiene un problema



Bibliografía



Introduction to Genetic Algorithms
S.N.Sivanandam y S.N. Deepa
Springer (2008) [Link Bibliotecas UC](#)