

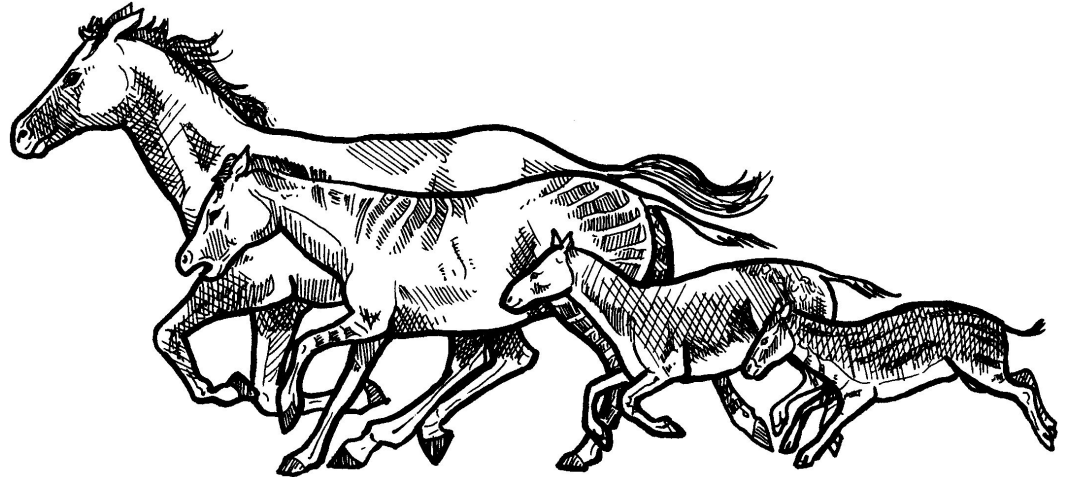
Algoritmos genéticos


Algoritmos y Estructuras de Datos III
DC - FCEyN - Universidad de Buenos Aires



Motivación

Utilizar los principios de la evolución biológica para construir algoritmos heurísticos que encuentren buenas soluciones a problemas.





¿Cómo sobreviven y evolucionan las especies?

- Reproducción.
- Selección natural:
 - Dado una población (conjunto de individuos) en un ambiente, sobreviven aquellos que estén más adaptados al ambiente dejando parte de su genoma como herencia mediante la reproducción.
 - Se basa en:
 - Los nuevos individuos deben heredar las particularidades de los seleccionados en la generación anterior.
 - La muestra poblacional debe contener suficiente varianza en sus genes.
 - La varianza tiene que permitir diferenciar los individuos de la población respecto a su capacidad de sobrevivir.
- Genoma: conjunto total de genes, un genoma describe todas las propiedades particulares de un individuo.



Algoritmos genéticos

Método adaptativo que sirve para resolver de forma heurística problemas de optimización.

Al igual que hill climbing, va caminando por el espacio de soluciones, pero lo hace de una forma distinta, en vez de moverse de una solución a otra, se mueve de un conjunto de soluciones (población) a otro conjunto de soluciones sin garantizar que el nuevo conjunto sea mejor igual o peor que el anterior.

Cada población pertenece a una generación y cada población hereda propiedades de la generación anterior. En otras palabras, transforma una población en otra mediante un conjunto de operaciones basadas en los principios de la evolución de darwin.

Ventajas respecto a hill climbing: Recorre el espacio de soluciones de una forma más amplia, disminuyendo la probabilidad de quedarse atascado en máximos locales.



But then a **mutation**
occures





Esquema general de los algoritmos genéticos

BEGIN /* Algoritmo Genético Simple */

Generar una población inicial.

Computar la función de evaluación (fitness) de cada individuo.

WHILE NOT cumple el criterio de terminación **DO**

BEGIN /* Producir nueva generación */

FOR |población actual| / 2 **DO**

BEGIN /*Ciclo reproductivo */

Seleccionar dos individuos de la anterior generación.

Cruzar (crossover) con cierta probabilidad los dos individuos obteniendo dos descendientes.

Mutar los dos descendientes con cierta probabilidad.

Evaluar los dos descendientes mediante la función de fitness.

Insertar los dos descendientes mutados en la nueva generación.

END

END

END

END

END



Criterio de terminación

- Al igual que en grasp, puede ser variado, pero siempre debe asegurar que sin importar cuán bien o mal evolucionan las generaciones, es necesario que el algoritmo termine en algún momento.
- Ejemplos:
 - Terminación por convergencia: Se dice que un algoritmo genético converge cuando no hay diferencias entre una generación y la siguiente.
Un posible problema es la convergencia prematura que nos lleva de forma rápida a máximos locales ya que un pequeño conjunto de la población domina al resto. Una forma de atacar este problema es cambiar la función de fitness y/o selección.
 - Cantidad fija de iteraciones: Siempre se realiza una cantidad fija de iteraciones del algoritmo sin importar la evolución de las generaciones.
 - Condición sobre la calidad de la población: Terminar cuando la calidad de la población respecto de la función objetivo (que puede no ser la misma que la función de fitness) cumple alguna condición.
 - Combinación de criterios: Una combinación de distintos criterios aplicados de forma que uno sirva para mejorar las debilidades de otro.



Métodos de selección

Método utilizado para determinar que individuos de una generación van a reproducirse y cuáles no. Es conveniente que el criterio de selección considere la evaluación del individuo en su decisión.

Ejemplos:

Selección simple: Seleccionar individuos de forma random (no considera la evaluación del individuo)

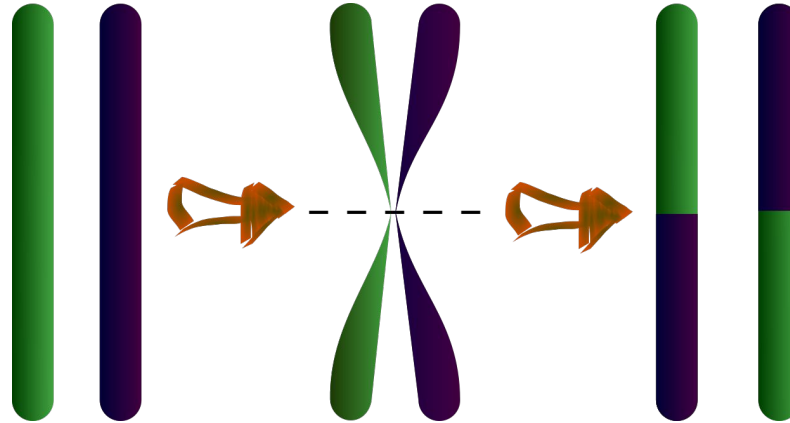
Selección ponderada: La probabilidad de que cada individuo quede seleccionado está ponderado por su evaluación.

Competencia: Luego de hacer competir a los individuos se selecciona a los $|población| / 2$ mejores individuos de la población.



Crossover

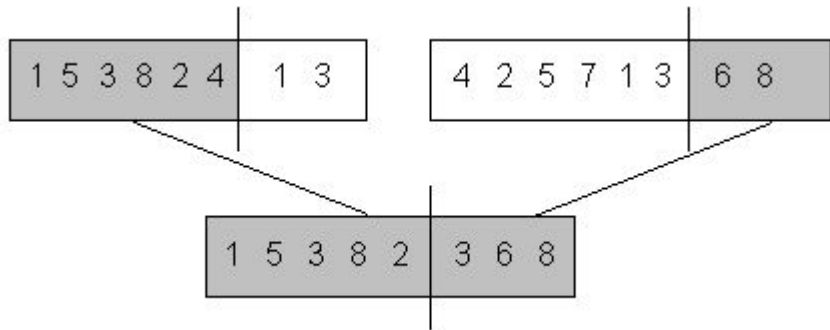
Crossover simple define un punto de corte e intercambia los genomas respecto a ese punto de corte, los genomas resultantes son una combinación de los anterior.



Puede no ser simple, por ejemplo, el punto de corte puede ser random y/o puede contener más de un punto de corte. También puede contener una cantidad random de puntos de corte, o puede ser respecto a otros criterios.

Crossover (simple)

La función de crossover se implementa sobre el genoma, el cual no es más que un vector de genes que contiene todos los genes de los individuos, estos genes son una parametrización de atributos que describen a los individuos. Generalmente los genes son atributos numéricos, pero podrían no serlos, un ejemplos de genes no numéricos se dan comúnmente en programación genética, donde los genes suelen ser las operaciones de la función.

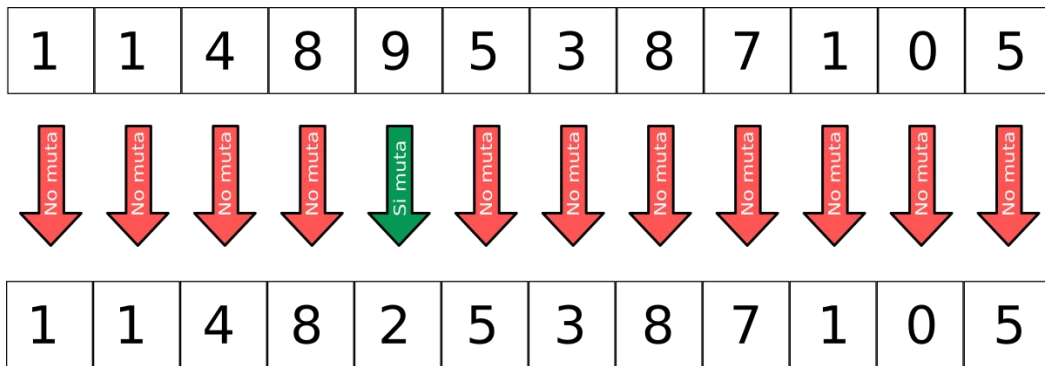


También es habitual tomar como genoma a la codificación binaria de los parámetros numéricos ya que esto modela mejor el fenómeno de crossover biológico donde múltiples genes son los que describen las distintas propiedades particulares del individuo.



Mutación

Es un proceso que altera con cierta probabilidad (generalmente muy baja) cada gen del genoma de un individuo, dicha alteración también se produce de forma aleatoria. Este proceso, modela el fenómeno de la mutación biológica, pero en términos de optimización, permite dar saltos dentro del espacio de soluciones, lo que es muy útil cuando las poblaciones quedan atrapadas en un máximo local.



Ejemplo:

$P(\text{mutación}) \in [0.001, 0.05]$

Mutación(gene) = gene + $N(\mu, \sigma^2)$



Evaluación (fitness)

Para poder determinar el proceso de selección evolutivo, es necesario poder evaluar los individuos de una población e indicar cuán aptos son para sobrevivir al ambiente. Llevado a los algoritmos genéticos, el ambiente es la función la función objetivo a optimizar. Esto no quiere decir que siempre se deba usar dicha función ya que la misma podría ser difícil de conseguir, por ejemplo, en el caso donde los individuos son jugadores de algún juego, lo que se quiere son individuos que sean jugadores expertos, pero resulta difícil medir el nivel de expertise de un jugador, por lo que en estos casos es necesario plantear otra medida de fitness.

Hay que notar que la función de fitness tiene un impacto directo sobre la decisión de quienes van a ser seleccionados, es por esto que de ella depende en gran parte la calidad de los resultados obtenidos.



¿ Dudas ?

MIT (OpenCourseWare) Genetic algorithms: <https://www.youtube.com/watch?v=kHyNqSnzP8Y>

Mario boss entrenado con un algoritmo genético: <https://www.youtube.com/watch?v=ofVKsxeYa6U>