

El Algoritmo Genético Simple

Héctor Alejandro Montes

hamontesv@uaemex.mx

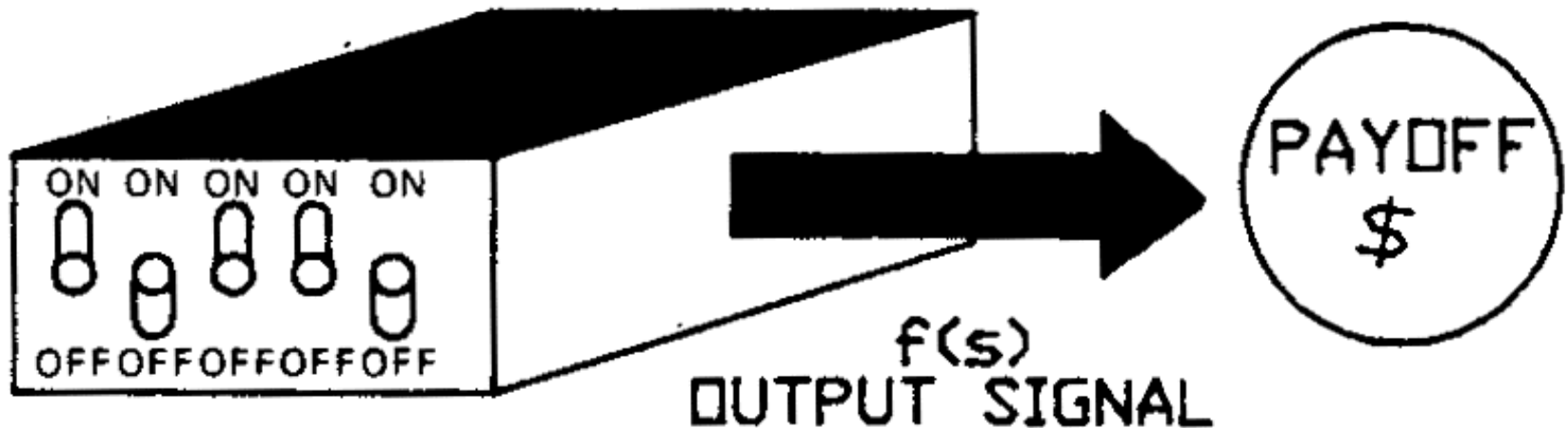
<http://scfi.uaemex.mx/hamontes>

AGs vs. Métodos tradicionales

- AGs utilizan una *codificación* del conjunto de parámetros, no con los parámetros.
- AGs buscan en una *población* de puntos y no en un sólo punto.
- AGs emplean información de *utilidad* (función objetivo), no diferenciales u otro conocimiento auxiliar.
- AGs utilizan reglas de transición *probabilísticas* y **no** reglas *determinísticas*.

Problema ejemplo

- El problema de optimización *caja negra* con cinco interruptores *on-off*.

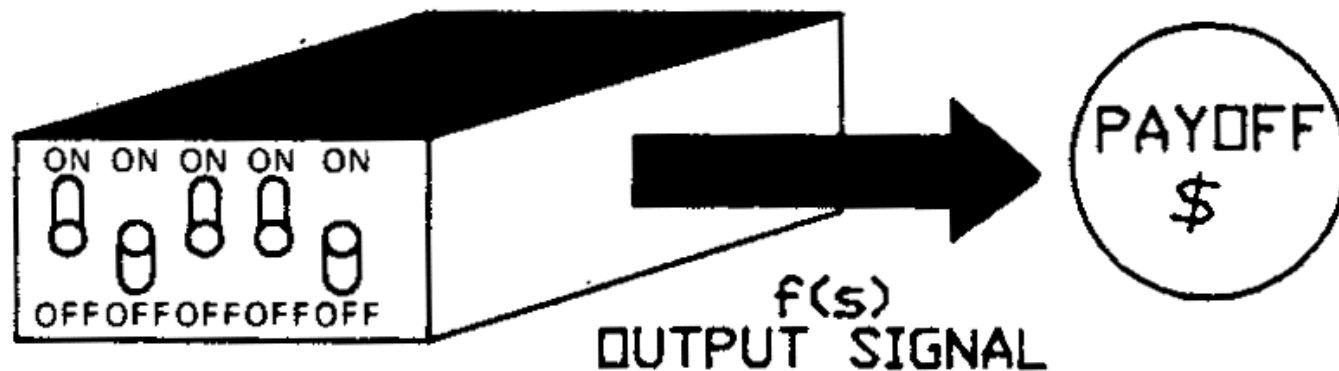


Codificación

- Los AGs sólo requieren una codificación y una medida de utilidad: **NO** necesitan saber el funcionamiento de la *caja negra*.
- El *objetivo* es poner los interruptores de tal forma que se obtenga el valor máximo en la señal de salida medido por $f(s)$.
- Primero, *codifiquemos* los interruptores como una cadena de longitud finita.

Codificación simple

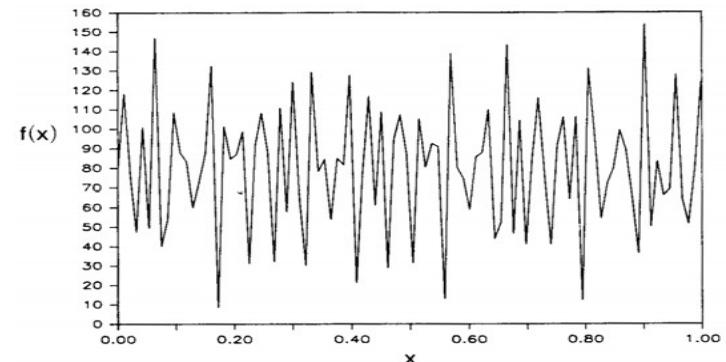
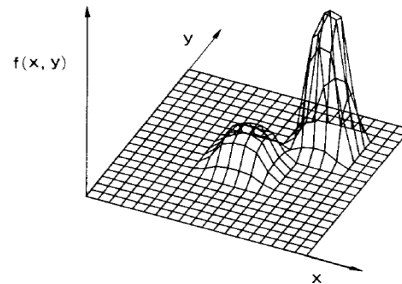
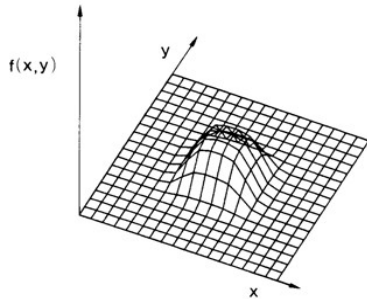
- **1** si el interruptor esta encendido (on)
- **0** si el interruptor esta apagado (off)
 - Ejemplo: **01001**



- ***No todas las codificaciones son tan obvias***

Métodos tradicionales de optimización

- Se mueven tímidamente de un punto a otro en el *espacio de búsqueda* utilizando alguna regla de transición.
- Éstos métodos usualmente carecen de una estrategia para salir de *falsos* picos en espacios multi-modales.



AGs

- Trabajan *simultáneamente* sobre un grupo de puntos (una población de cadenas), explorando varios puntos en paralelo
- Se reduce la probabilidad de encontrar un *falso* pico, a diferencia de los métodos que se mueven punto-a-punto.

Optimizando la “Caja Negra”

- Los AGs comienzan con una población de cadenas usualmente generada al azar.
- Generan luego poblaciones sucesivas de cadenas a partir de la población actual.
- Ejemplo: población inicial de tamaño $n = 4$ (pequeña para un AG)
 - **01101**
 - **11000**
 - **01000**
 - **10011**

Operadores básicos

- Debemos definir operaciones que tomen esta población inicial para generar las siguientes y que mejoren con el tiempo
 - Reproducción
 - Cruza
 - Mutación

Reproducción

- Proceso en el que los individuos (cadenas) se copian de acuerdo a sus valores de *aptitud* f (medida de ganancia, utilidad o calidad)
- Los individuos con un valor f más alto tienen una probabilidad mayor de contribuir con descendencia a la siguiente generación

Reproducción

- Versión artificial de la selección natural
 - Supervivencia del más apto
- En la naturaleza esa aptitud está determinada por la habilidad de la criatura para sobrevivir
- En nuestro escenario artificial, la función objetivo (o de *aptitud*, o *fitness*) es el árbitro que dicta la supervivencia de la cadena

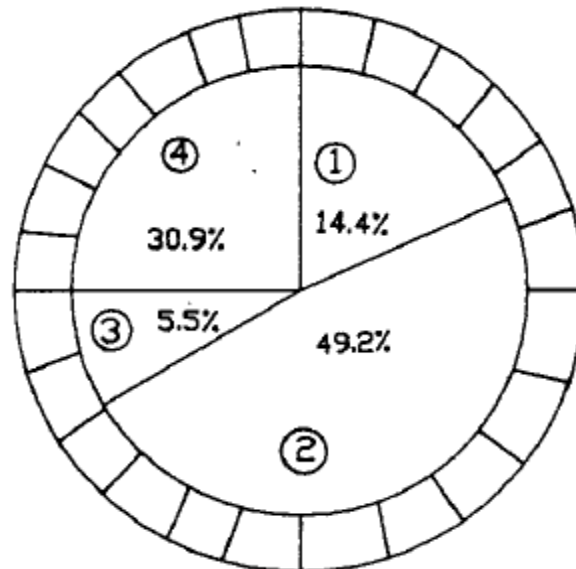
Reproducción

- Suponga que los valores de la función de aptitud f son:

No.	String	Fitness	% of Total
1	01101	169	14.4
2	11000	576	49.2
3	01000	64	5.5
4	10011	361	30.9
Total		1170	100.0

Reproducción

- Imaginen una ruleta *sesgada* o *ponderada* donde cada cadena de la población actual tiene una sección con tamaño *proporcional* a su *fitness*.



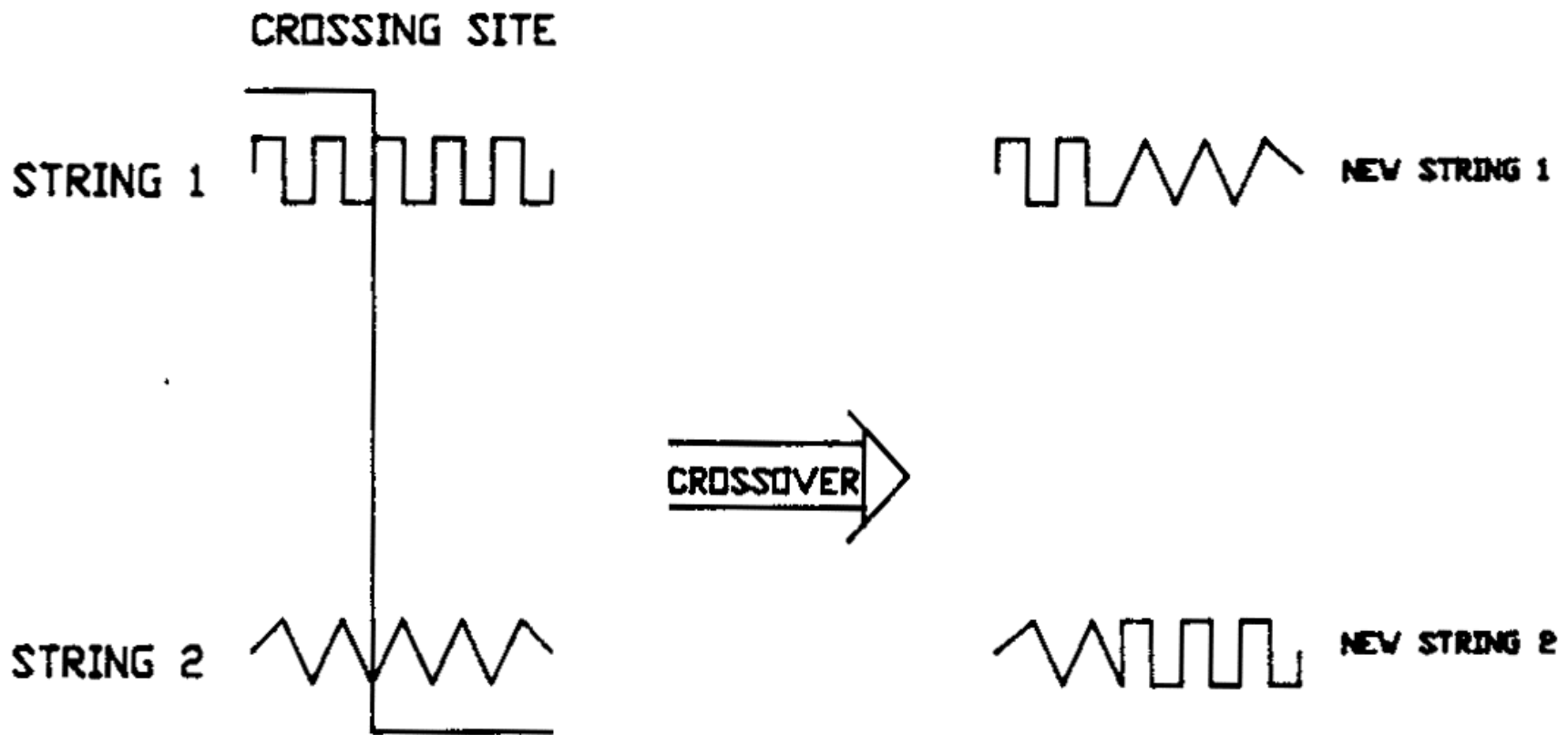
Reproducción

- Para seleccionar los individuos a reproducir, simplemente gire la ruleta
- Se requieren dos padres para generar descendencia
- Una vez que los individuos que se reproducirán han sido seleccionados, se utiliza *cruza* y *mutación* para producir una nueva población

Cruza simple

BEFORE CROSSOVER

AFTER CROSSOVER



Cruza simple

- Una posición k se selecciona al azar de manera uniforme entre 1 y la longitud total L de la cadena
- Dos nuevas cadenas se crean intercambiando todos los caracteres entre $k+1$ y L

Cruza simple, ejemplo

- Suponga que A_1 y A_2 son individuos que han sido elegidos usando la ruleta y que la posición de cruce elegida al azar es la 4:
 - $A_1 = 0 \ 1 \ 1 \ 0 \ | \ 1$
 - $A_2 = 1 \ 1 \ 0 \ 0 \ | \ 0$
- La cruce produce estos dos nuevos individuos:
 - $A'_1 = 0 \ 1 \ 1 \ 0 \ 0$
 - $A'_2 = 1 \ 1 \ 0 \ 0 \ 1$que formarán parte de la nueva generación

Reproducción y Cruza

- Sorprendentemente simple, involucra:
 - Generación de números aleatorios
 - Copias de cadenas
 - Intercambios parciales de cadenas
- La combinación de reproducción y cruza le dan a los AGs mucho de su poder

AGs

- Explotan la información de la población
 - Reproducen individuos de alta calidad de acuerdo a su desempeño (*fitness*)
 - Cruzan porciones de éstos individuos con porciones de otros individuos de alto desempeño
- Especulan sobre nuevos individuos contruídos por porciones “*buenas*” de individuos anteriores

Mutación

- Es la alteración ocasional (con baja probabilidad) del valor de una posición en el individuo
- Por sí sola, es una *búsqueda aleatoria*
- Cuando se utiliza con Reproducción y Cruza, es una estrategia contra la pérdida prematura de información útil

Mutación

- Necesario porque, aunque *Reproducción* y *Cruza* buscan de manera efectiva, ocasionalmente pueden volverse muy especializadas y perder material genético útil
- El operador de *Mutación* protege contra una *pérdida irrecuperable* manteniendo la *diversidad*

Mutación

- Su papel es secundario en un AG
- La frecuencia de mutación es del orden de *una* mutación por *cada mil* intercambios
- La proporción es similarmente pequeña (o más pequeña aún) en la naturaleza

Estructura genral de un AG

Procedimiento Algoritmo Genético

Inicio (1)

$t = 0;$

inicializar $P(t);$

evaluar $P(t);$

Mientras (no se cumpla la condición de parada) hacer

Inicio(2)

$t = t + 1$

seleccionar $P(t)$ desde $P(t-1)$

recombinar $P(t)$

mutación $P(t)$

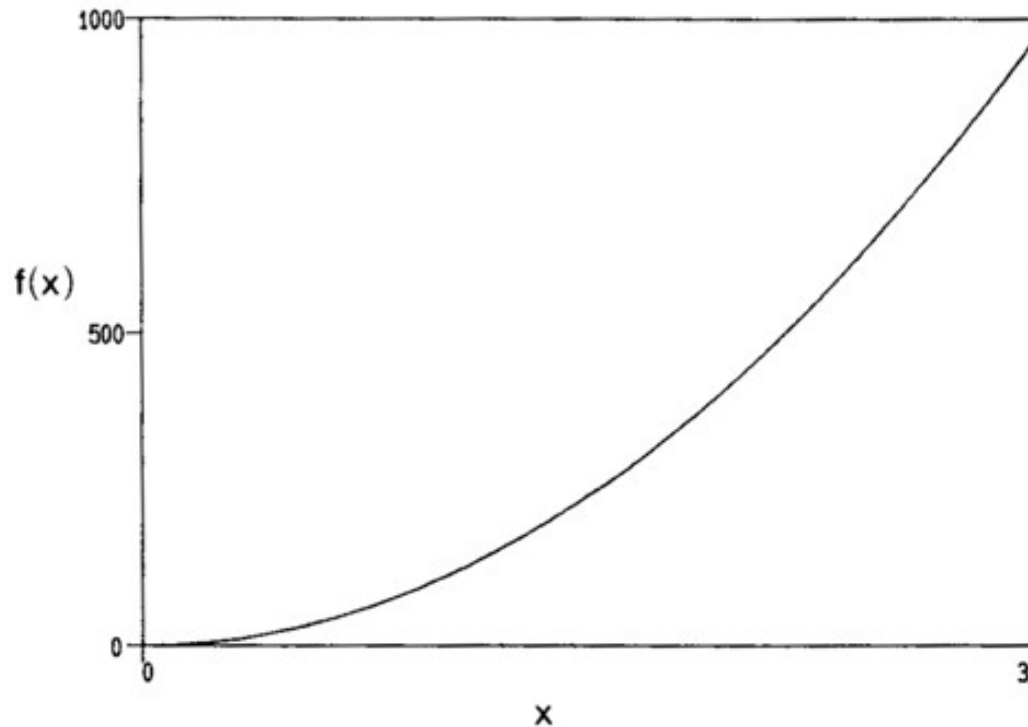
evaluar $P(t)$

Final(2)

Final(1)

Ejemplo

- Maximizar la función $f(x) = x^2$, donde x varía entre 0 y 31



Preparación previa

- **Codificación:** binaria entera sin signo de longitud 5
- **Tamaño de población** = 4
- **Población inicial** generada al *azar*

String No.	Initial Population (Randomly Generated)	x Value (Unsigned Integer)	$f(x)$ x^2	pselect, $\frac{f_i}{\sum f}$	Expected count $\frac{f_i}{\bar{f}}$	Actual Count from (Roulette Wheel)	Mating Pool after Reproduction (Cross Site Shown)	Mate (Randomly Selected)	Crossover Site (Randomly Selected)	New Population	x Value	$f(x)$ x^2
1	0 1 1 0 1	13	169	0.14	0.58	1	0 1 1 0 1	2	4	0 1 1 0 0	12	144
2	1 1 0 0 0	24	576	0.49	1.97	2	1 1 0 0 0	1	4	1 1 0 0 1	25	625
3	0 1 0 0 0	8	64	0.06	0.22	0	1 1 0 0 0	4	2	1 1 0 1 1	27	729
4	1 0 0 1 1	19	361	0.31	1.23	1	1 0 0 1 1	3	2	1 0 0 0 0	16	256
Sum			1170	1.00	4.00	4.0						1754
Average			<u>293</u>	0.25	1.00	1.0						<u>439</u>
Max			<u>576</u>	0.49	1.97	2.0						<u>729</u>

Proceso

- **Selección de ruleta**
 - Individuo 1 y 4: una copia
 - Individuo 2: dos copias
 - Individuo 3: sin copias
- **Cruza simple**
 - Los sitios de cruza se seleccionan al azar
 - Se establece una probabilidad de cruza (0.65)
- **Mutación**
 - Probabilidad de cruza = 0.001
 - 20 bits tranferidos, esperamos 20×0.001 bits a mutar

Evaluar nueva población

- *Decodificar* los nuevos individuos
- Calcular los valores de aptitud: *fitness*
 - El promedio de *fitness* ha mejorado de **293** a **439** en una generación
- Seguir con la **nueva generación**

Ejecución de un AG

- En **cada generación**:
 - Un **nuevo conjunto de soluciones** se crea utilizando partes de sus predecesores más aptos
 - Ocasionalmente, partes nuevas también se añaden y se prueban
 - Los AGs **no** son simples **búsquedas al azar**
 - Explotan información histórica para especular en nuevas direcciones de búsqueda en busca de mejorar