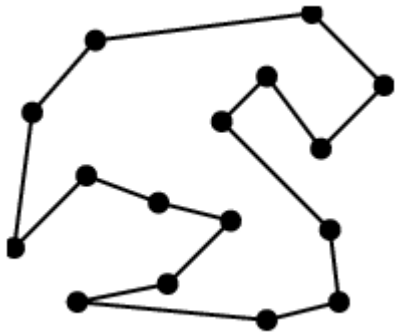
Algoritmos genéticos como métodos de aproximación analítica y búsqueda de óptimos locales

Jorge Salas Chacón	A03804
Rubén Jiménez Goñi	A93212
Juan Camilo Carrillo Casas	A91369
Marco Vinicio Artavia Quesada	A90661

Introducción

- Surgimiento de los Algoritmos Genéticos:
  - Problemas de computabilidad, existen problemas deterministas complejos en cuanto a espacio y tiempo, dentro de estos están los conocidos como NP.
  - La optimización encontrada en los algoritmos genéticos permite dar buenos resultados, siendo esta una herramienta importante aunque no dé la solución óptima al problema.



Términos

- **Teoría de la evolución de Darwin**: Los individuos más fuertes y mejor constituidos sobreviven, mientras que los débiles mueren. Las especies evolucionan con el tiempo y adquieren nuevas características para ajustarse al medio (Solano, Y.).

- **Individuo**: Sujetos que representan las posibles soluciones a un problema dado, el cual se desea resolver mediante un algoritmo genético. Como se verá más adelante, es importante poder representar las características de un individuo mediante una cadena de caracteres.

- **Código genético**: Información del individuo.

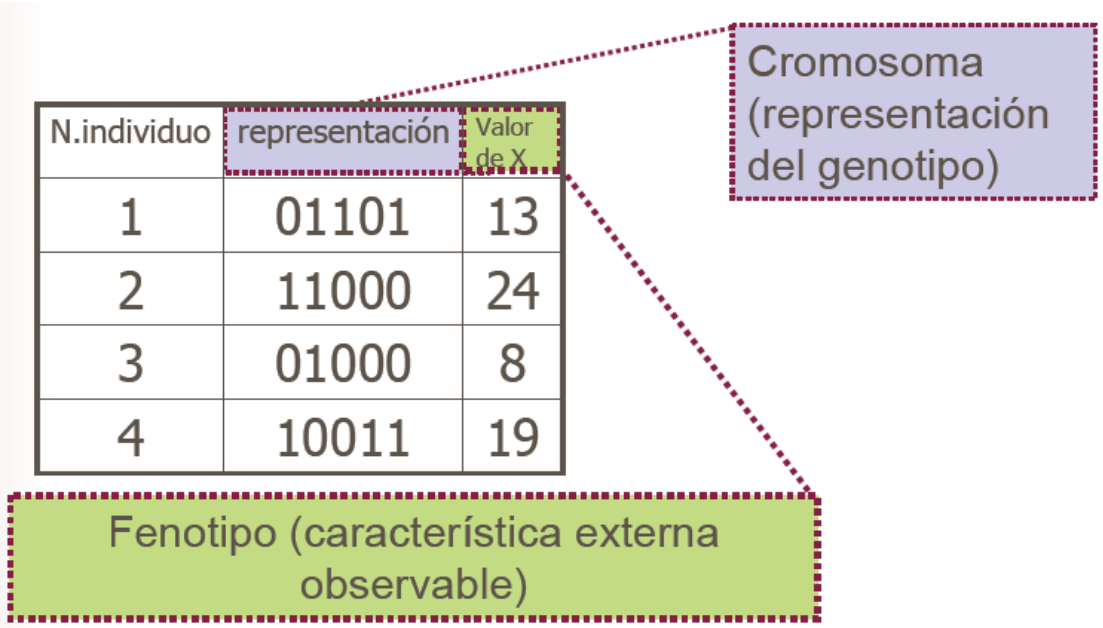
Términos

- **Genotipo**: Constitución genética de un organismo. Conjunto de características que se transmiten a los descendientes (Solano, Y.).

- **Cromosomoma**: Representación del genotipo a través de un string, como por ejemplo una cadena de bits. Están compuestos por genes, los cuales pueden tomar valores llamados alelos (Solano, Y.).

Términos

- **Fenotipo**: Características externas observables de un organismo o individuo. Solución perteneciente al espacio de soluciones del problema en estudio.
- **Ejemplo**: Si se desea maximizar la función $f(x) = x^2$ sobre el intervalo [0, 31]



Fuente: Solano, Yadira. *Algoritmos Genéticos*. Filminas del curso "Paradigmas Computacionales" CI-1441. Año 2011, segundo semestre.

Términos

- **Población**: Conjunto de individuos que serán mutados, reproducidos y evaluados.

- **Función de fitness**: Evalúa la aptitud de un individuo.

- **Operadores genéticos**: Principales mecanismos utilizados para la generación de nuevos individuos.
    1. Reproducción
    2. Cruce
    3. Mutación

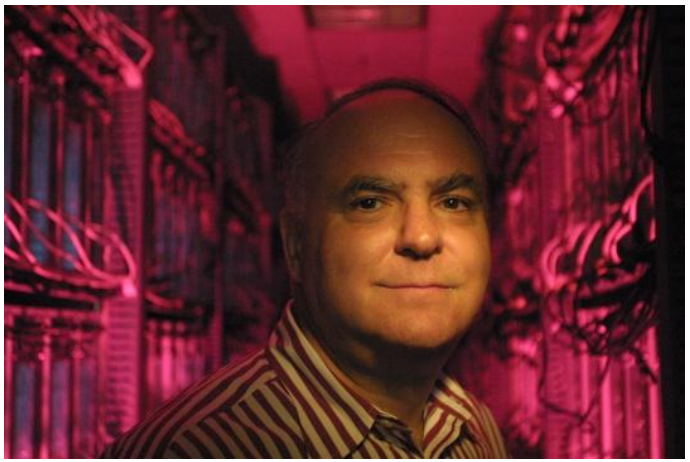
Orígenes

- Friedberg (1958)

- John Holland (1975)

Fuente: http://www.genetic-programming.com/johnkoza.html

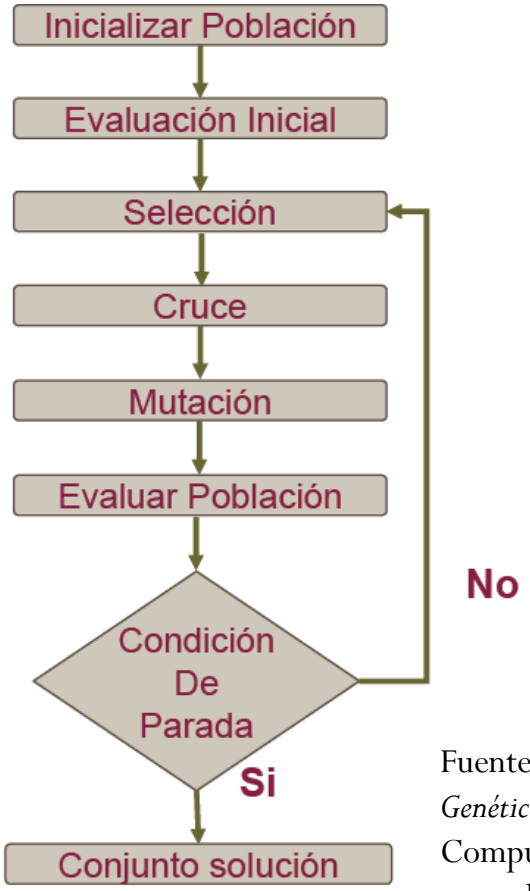
- John Koza (1992)



Fuente: http://www.lsa.umich.edu/psych/people/directory/profiles/faculty/?uniquename=jholland

Algoritmo-Flujo del algoritmo (1)

- Representación
  - Binaria usualmente
- Población inicial
- Función de evaluación
- Selección
- Reproducción



Fuente: Solano, Yadira. *Algoritmos Genéticos*. Filminas del curso "Paradigmas Computacionales" CI-1441. Año 2011, segundo semestre.

Algoritmo-Flujo del algoritmo (2)

```
function GENETIC-ALGORITHM(population, FITNESS-FN) returns an individual
  inputs: population, a set of individuals
          FITNESS-FN, a function that measures the fitness of an individual

  repeat
    parents ← SELECTION(population, FITNESS-FN)
    population ← REPRODUCTION(parents)
  until some individual is fit enough
  return the best individual in population, according to FITNESS-FN
```

Fuente: Russell, S., & Norvig, P. (1995). *Artificial Intelligence A Modern Approach*. New Jersey: Prentice Hall.

Algoritmo-Población inicial

- Primero, se debe buscar la forma de representar la población y cómo se codificará la respuesta
- ¿Qué representa cada miembro de la población?
- Sugerencias para la población inicial
  - Saber soluciones parciales
  - Se genera al azar, y que sea variada
- Tamaño de la población
  - ¿Qué ocurre si es pequeña?
  - ¿Y si es grande?

Algoritmo-Función de fitness

- Es el operador que indica que tan bueno es un individuo.
- Puede ser una función matemática, probabilística o heurística.
- Debe medir la aptitud correctamente para evitar la convergencia prematura del algoritmo.
- Es lo más difícil de diseñar en un algoritmo genético.

$$F: \text{Cromosomas} \rightarrow R^{+}$$

$$x \rightarrow F(x)$$

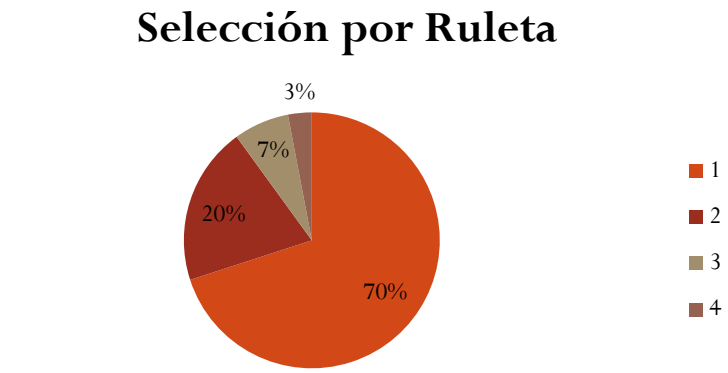
Algoritmo-Operadores genéticos

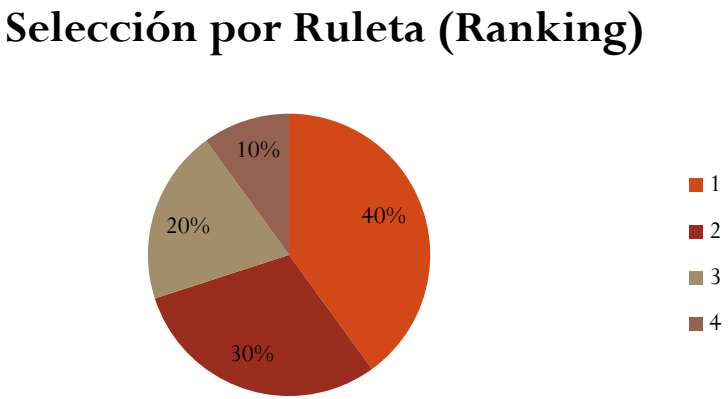
- Selección
  - Determinado por la función de adaptación
- Reproducción
  - Cruce
  - Clonación
  - Élite
- Mutación

Algoritmo-Selección

- Seleccionar individuos de la población para reproducirlos

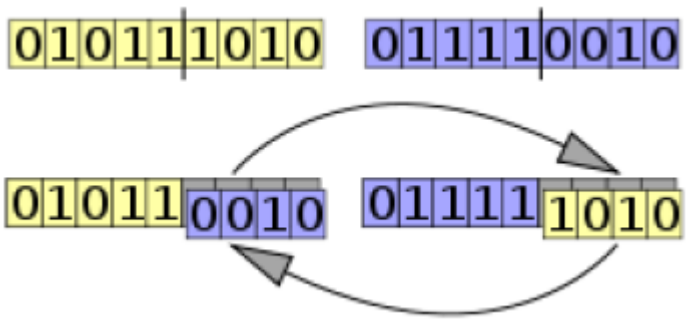
- Algunas opciones
  - Basado en rango
  - Ruleta
    - Ranking
  - Torneo





Algoritmo-Reproducción (1)

- Es la forma como se combinan dos cromosomas para generar los descendientes con características de ambos progenitores.
- Cantidad de genes por padre:
  - Al azar.
  - Determinados.
- Cruce de un punto.
- Cruce de n puntos.



Algoritmo-Reproducción (2)

- Élite
  - Se toma un grupo de individuos sobresalientes y se conservan cruzándose entre ellos.
  - Pueden generar buenas soluciones pero pueden provocar mínimos locales.
  - La élite debe de aumentar entre más cercanas estén las generaciones de llegar a su fin.
- Clonación
  - Se genera una copia del padre en la siguiente generación.

Algoritmo-Mutación (1)

- Consiste en modificar de manera aleatoria parte de un individuo.
- Se modifican los genes del individuo, los cuales usualmente se representan como una cadena de bits, aunque esto no es necesario.
- Se debe asignar una probabilidad de ocurrencia a la mutación.
- Introduce nuevo material genético a la población.

Algoritmo-Mutación (2)

- Asegura que todos los puntos del espacio de búsqueda tengan probabilidad mayor que cero de ser explorados.

- Garantiza la convergencia.

- La mutación contribuye a:
  - La diversidad genética de la especie.
  - Incrementar los saltos evolutivos.
  - Desbloquear el algoritmo.

Fuente: Solano, Yadira. *Algoritmos Genéticos*. Filminas del curso "Paradigmas Computacionales" CI-1441. Año 2011, segundo semestre.

Algoritmo-Mutación (3)

- Cuando los individuos se representan como cadenas de bits, existen diversas técnicas de mutación disponibles, entre ellas:

  - **Mutación multibit**: Se cambia aleatoriamente más de un bit de la cadena del individuo.

  - **Mutación de un bit**: A diferencia de la anterior, solo se modifica un bit.

  - **Mutación de intercambio**: Se seleccionan dos bits de la cadena aleatoriamente y se intercambian.

  - **Mutación de barajado**: Se selecciona un rango de bits de la cadena y se reacomodan aleatoriamente.

Aplicaciones de algoritmos genéticos en robótica

- Robótica Evolutiva
  - Adaptación al medio ambiente
- ¿Porqué se usa?
  - Es difícil ver todos los casos
  - Ambientes no controlados son complicados
- ¿Hay inconvenientes en usarlos?
  - Forma de representación
  - Función fitness
  - Muchas diferencias entre ambiente simulado y real

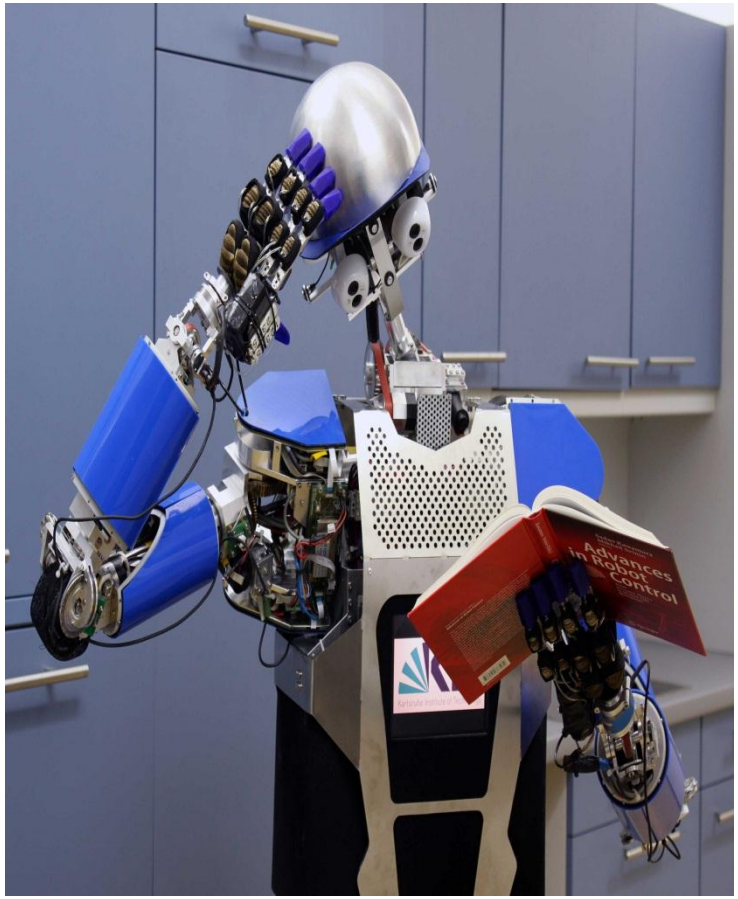


Imagen tomada de : http://www.itas.fzk.de/v/evolutionary_robotics/info_eng.htm

Aplicaciones de algoritmos genéticos en robótica- Ejemplos de uso (1)

- Control de navegación
  - Moverse sin colisionar con obstáculos
  - Robot Khepera
    - Modificación de pesos en Red Neuronal
    - Función fitness: rotación de la rueda
    - Comportamiento emergente: moduló velocidad
- Planificación de trayectorias
- Optimización de trayectorias

Aplicaciones de algoritmos genéticos en robótica- Ejemplos de uso (2)

Robot Khepera

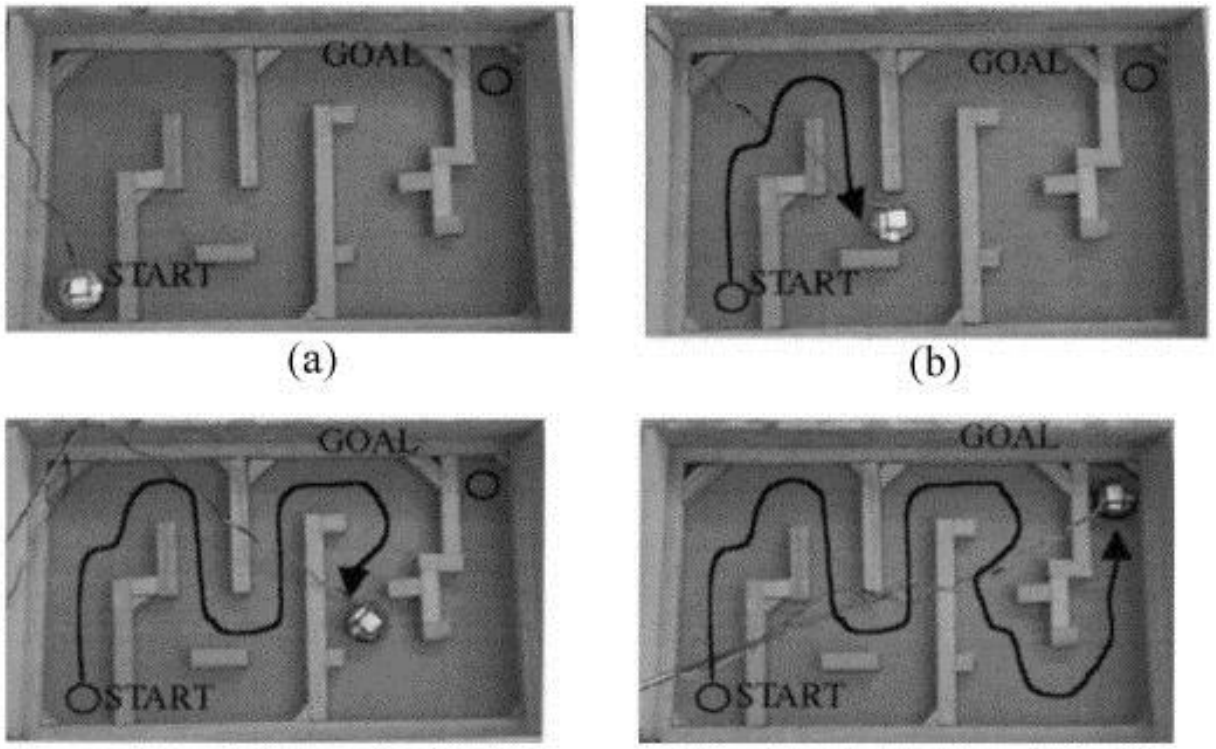


Imagen tomada de: http://www.sciencedirect.com/science/article/pii/S0263224100000440

Aplicaciones de algoritmos genéticos en robótica- Ejemplos de uso (3)

- Optimización en técnicas de transporte
- Calificación de objetos
  - Ejemplo de Nolfi y Marocco
    - Algoritmos genéticos mejoran red neuronal
    - Debía identificar diferencias entre una línea ancha y otra angosta
  - Complicado porque depende de muchas variables
- Brazos robóticos
  - Evolucionar movimientos de efectores
- Construcción de robots
  - Optimización de la suspensión

Aplicaciones de algoritmos genéticos en robótica- Ejemplos de uso (4)

- Optimización de parámetros
  - Controlador PID
- Balanceo de robot
- Aprender a caminar
- Evolución cooperativa
  - Robocopa

Imagen tomada de:
http://www.humanoidsoccer.org/

Aplicaciones de algoritmos genéticos en robótica- Ejemplos de uso (5)

- Robot que aprende a evitar colisiones, manteniéndose en el centro

- Robot que aprende a caminar hacia adelante:

Ventajas (1)

- Su componente aleatorio les permite evitar caer en trampas en que los métodos deterministas pueden caer.

- No obstante, recorren el espacio de soluciones de una manera más inteligente que una búsqueda aleatoria pura.

- Menos sensibles a quedar atrapados en óptimos locales, pues trabajan con conjuntos de soluciones.

Ventajas (2)

- Aceptan cambios en la función de fitness, lo cual permite mejorar el algoritmo sin mucho esfuerzo y sin modificaciones extensas al código.

- Se pueden aplicar en una amplia gama de áreas además de la robótica.

- Permiten explotar arquitecturas en paralelo. La evaluación de la función de fitness es altamente paralelizable. Ej: John Koza y el cluster Beowulf.

Ventajas (3)

- No requieren conocimientos específicos del área de conocimiento del problema a resolver, pues se pueden usar como caja negra. No obstante, conocimientos en dicha área pueden ayudar a complementar el trabajo de los operadores genéticos.

Desventajas (1)

- Por su carácter aleatorio, no se puede garantizar que produzcan una buena solución.

- En general se centran en obtener una muy buena solución, pero no la mejor de todas (es decir, la óptima), pues se les considera como un tipo de técnica heurística.

- Pueden llegar a requerir alto poder de computo, pues se utilizan poblaciones grandes y se requieren muchas iteraciones en las cuales se evalúa el fitness de cada individuo.

Desventajas (2)

- Lo anterior influye negativamente en su eficiencia y por esto se les considera lentos.

- La población inicial y los criterios de selección y reemplazo pueden influir en el tiempo que se requiere para obtener la solución deseada.

- Algunos autores consideran que su alta aplicabilidad los hace poco eficientes para algunos tipos muy específicos de problemas.

Conclusiones (1)

- Los algoritmos genéticos pueden ser utilizados en una gran cantidad de áreas
- Dan soluciones a diferentes tipos de problemas, en una forma relativamente eficiente
- Hay que tener un sumo cuidado a la hora de representar la población, su función fitness y de cómo seleccionamos a los individuos

Conclusiones (2)

- Los algoritmos genéticos son ideales para realizar optimizaciones gracias a sus principios darwinianos.

- En el área de la robótica, son vitales para realizar pruebas en ambientes no controlados. Además, son más consistentes.

Bibliografía

Russell, S., & Norvig, P. (1995). *Artificial Intelligence A Modern Approach.* New Jersey: Prentice Hall.

Salas, Jorge. *Programación Genética*. Proyecto de Investigación teórica del curso "Paradigmas Computacionales" CI-1441. Año 2011, segundo semestre.

Solano, Yadira. *Algoritmos Genéticos*. Filminas del curso "Paradigmas Computacionales" CI-1441. Año 2011, segundo semestre.

Wahde, M. *Evolutionary Robotics: The Use of Artificial Evolution in Robotics*. Chalmers University of Technology, Gotemborg, Suecia, 2004

Zhang, J. (2006). Genetic Algorithms for Optimal Design of Vehicle Suspensions. *Engineering of Intelligent Systems, 2006 IEEE International Conference on*, (págs. 1 - 6).