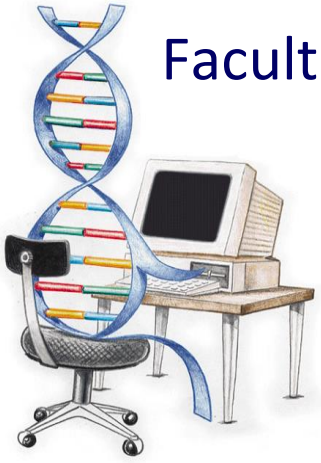
ALGORITMOS EVOLUTIVOS
Curso 2024

Tema 2: Computación Evolutiva

Centro de Cálculo, Instituto de Computación
Facultad de Ingeniería, Universidad de la República, Uruguay



cecal

UNIVERSIDAD DE LA REPÚBLICA URUGUAY

UNIVERSIDAD DE LA REPÚBLICA URUGUAY

"I have called this principle, by which, each slight variation, if useful, is preserved, by the term of Natural Selection.

… … …

The expression often used by Mr. Herbert Spencer of the Survival of the Fittest is more accurate, and is sometimes equally convenient."

CHARLES DARWIN

On the Origin of Species by means of Natural Selection, 1859.

Computación evolutiva

Contenido

UNIVERSIDAD DE LA REPÚBLICA URUGUAY

1. Evolución natural y neo-darwinismo
2. Computación evolutiva
3. Técnicas de computación evolutiva
   - Programación genética
   - Estrategias de evolución
   - Algoritmos genéticos
4. Análisis comparativo
   - Comparación entre técnicas de computación evolutiva
   - Comparación con respecto a otras técnicas
   - Análisis de ventajas y desventajas de su aplicación
5. Teorema No Free Lunch

UNIVERSIDAD DE LA REPÚBLICA URUGUAY

1

EVOLUCIÓN NATURAL

1 - Teorías evolutivas

Teoría de la evolución

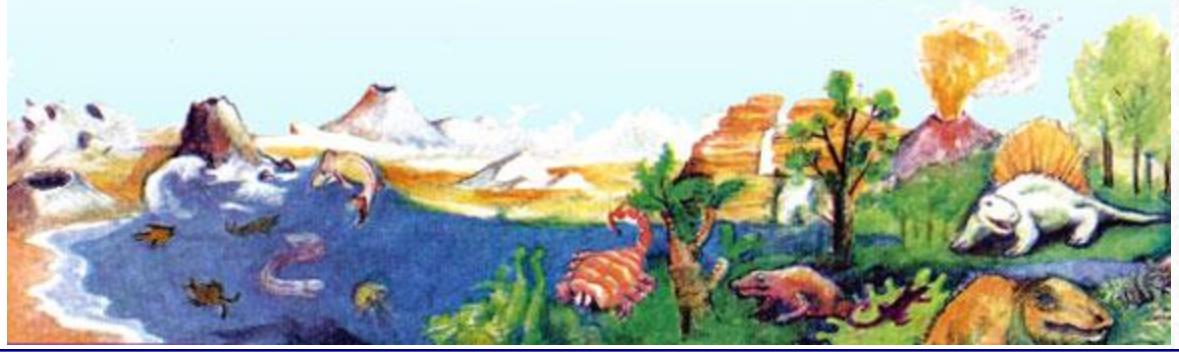
UNIVERSIDAD DE LA REPÚBLICA URUGUAY

- Teoría científica que explica la diversidad de formas de vida en la Tierra
  - Soportada por los hechos y las evidencias
  - Nunca refutada por ningún argumento o evidencia

- Es una de las teorías científicas FORMIDABLES (Penrose, 2000).

"Revolución científica"	Evolución	Revolución de la física
Astronomía, Física	Ciencias Naturales	Física
Copérnico, Kepler, Galileo, Newton	Lamarck, Darwin, Wallace	Einstein, Bohr, Schrodinger

1500 1600 1700 1800 1900 2000



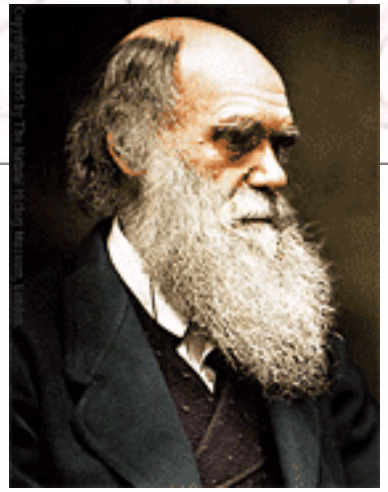
1 - Teorías evolutivas

Teoría de la evolución

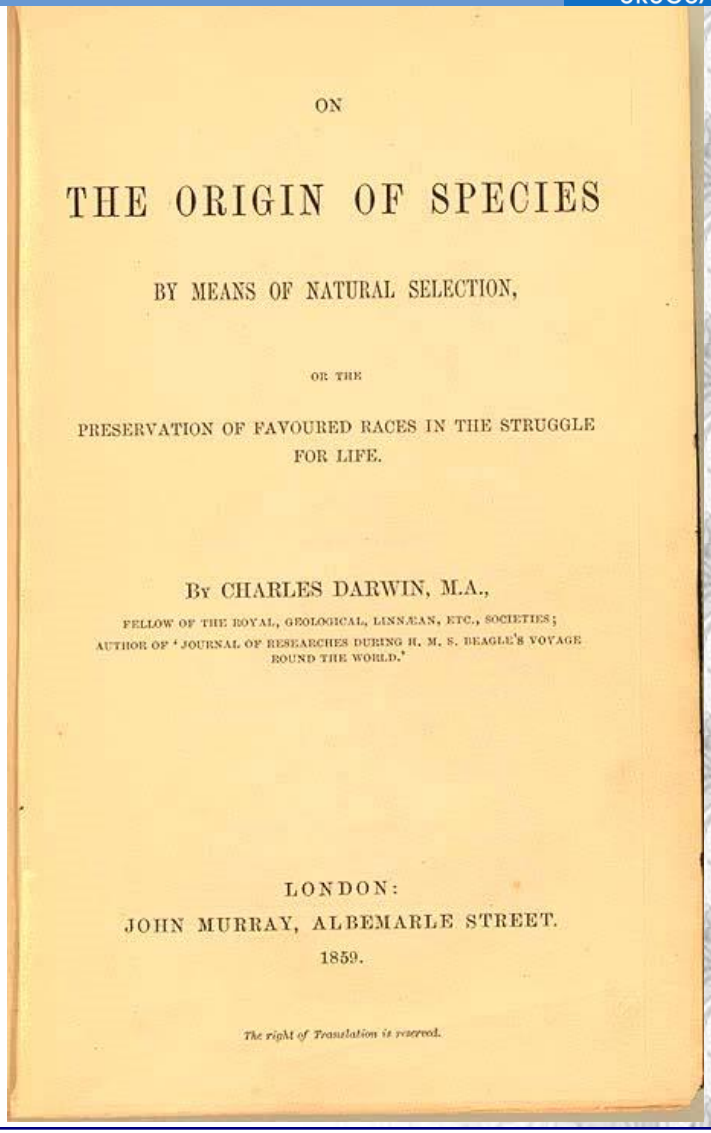
UNIVERSIDAD DE LA REPÚBLICA URUGUAY

"On the Origin of Species by Means of Natural Selection, or the Preservation of Favoured Races in the Struggle for Life"

Charles Darwin, 1859



Charles Darwin



ON

THE ORIGIN OF SPECIES

BY MEANS OF NATURAL SELECTION,

OR THE

PRESERVATION OF FAVOURED RACES IN THE STRUGGLE FOR LIFE.

By CHARLES DARWIN, M.A.,

FELLOW OF THE ROYAL, GEOLOGICAL, LINNÆAN, ETC., SOCIETIES;
AUTHOR OF 'JOURNAL OF RESEARCHES DURING H. M. S. BEAGLE'S VOYAGE ROUND THE WORLD.'

LONDON:
JOHN MURRAY, ALBEMARLE STREET.
1859.

The right of Translation is reserved.

1 - Teorías evolutivas

Teoría de la evolución: los pioneros

UNIVERSIDAD DE LA REPÚBLICA URUGUAY

- George Louis Leclerc (Conde de Buffon)
  - Primera referencia anti-creacionista: *Histoire Naturelle* (1749–1788, 50 volúmenes)
  - A partir de las similitudes entre el hombre y los simios, especuló sobre la existencia de un ancestro común
  - No describió un mecanismo responsable de los cambios orgánicos; creía que el ambiente influía y los provocaba

G.L. Leclerc (Buffon)

- Pierre Antoine de Monet (Chevalier de Lamarck)
  - Elaboró la primera teoría evolutiva (el *Lamarckismo*), a principios del siglo XIX
  - Propuso un mecanismo evolutivo basado en la *influencia del medio ambiente* y postuló la *transmisión de las características adquiridas* durante la vida a los individuos descendientes

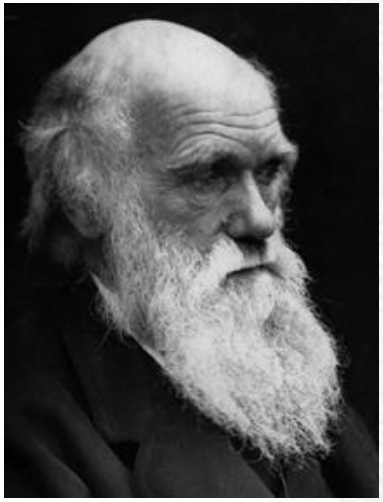
P.A. de Monet (Lamarck)

1 - Teorías evolutivas

Teoría de la evolución

UNIVERSIDAD DE LA REPÚBLICA URUGUAY

- Los sistematizadores
  - Charles Darwin y Alfred Russel Wallace
  - Plantearon el concepto de evolución de las especies a través de un lento proceso de selección natural



Charles Darwin

Alfred Wallace

Teoría de la evolución

UNIVERSIDAD DE LA REPÚBLICA URUGUAY

- Los continuadores
  - Gregor Mendel describió las leyes que rigen la herencia genética en 1865. Sus ideas fueron relegadas hasta 1900

Gregor Mendel

  - August Weismann desarrolló en 1892 una teoría sobre la transmisión de información hereditaria a través del germoplasma

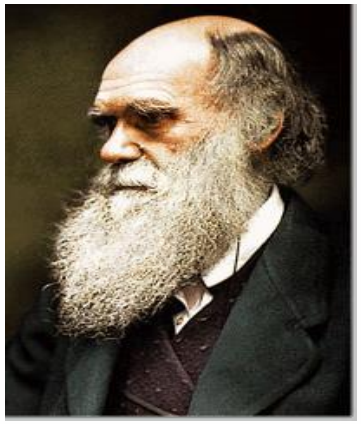
August Weismann

1 - Teorías evolutivas

Neo-darwinismo

UNIVERSIDAD DE LA REPÚBLICA URUGUAY

- Teoría evolutiva que combina las ideas de Darwin-Wallace, Weismann y Mendel




- Explica la vida a través de los conceptos de:
  - Competencia entre individuos
  - Selección natural
  - Transmisión hereditaria
  - Reproducción y mutación

2

UNIVERSIDAD DE LA REPÚBLICA URUGUAY

"Evolution is cleverer than you are"

FRANCIS CRICK

Elbow Room: the varieties of free will worth wanting

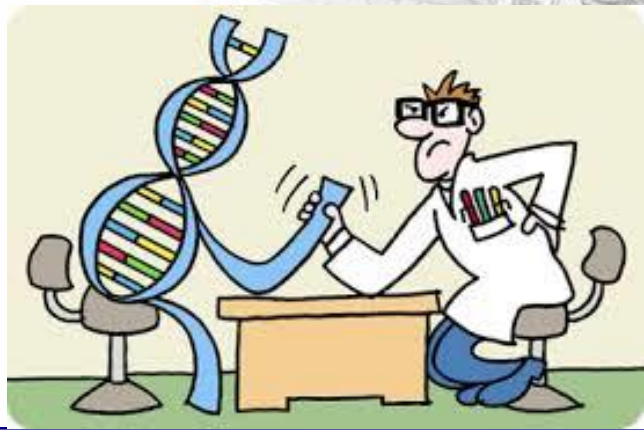
D. Dennett, 1984

COMPUTACIÓN EVOLUTIVA

Conceptos

UNIVERSIDAD DE LA REPÚBLICA URUGUAY

- Metaheurísticas que emulan la evolución biológica
  - Interpretan la naturaleza como una formidable maquinaria de resolución de problemas
  - Aplican un mecanismo análogo a los procesos evolutivos naturales, para resolver problemas de búsqueda y optimización
  - Trabajan con una población (de representaciones) de soluciones
  - Principios: selección natural (aptitud), reproducción (recombinación y mutación) y diversidad genética
  - Siguen la idea de la *supervivencia de los individuos más aptos*, evaluando la aptitud de acuerdo al problema a resolver, mediante una *función de fitness*

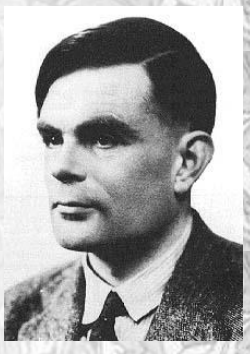


2 – Computación evolutiva

Reseña histórica

UNIVERSIDAD DE LA REPÚBLICA URUGUAY

- Los programas “autoreplicables” y “evolutivos” fueron sugeridos desde los inicios de la era de la computación, entre 1948 y 1966.
  - Alan Turing investigó las relaciones sobre evolución natural y aprendizaje. Propuso desarrollar programas automodificables, capaces de jugar ajedrez y simular otras actividades inteligentes sencillas, utilizando técnicas evolutivas
  - Neumann János propuso mecanismos evolutivos para implementar autómatas con poder computacional equivalente a máquinas de Turing. Conjeturó sobre poblaciones de autómatas trabajando cooperativamente y comunicándose entre sí. (“Teoría de autómatas autorreplicables”, inconcluso, 1966)



Alan Turing

Neumann János

Autoestudio:

Reseña historica sobre técnicas de computación evolutiva

Slides 77-82 y páginas 7-12 en http://www.fing.edu.uy/~sergion/Tesis.pdf

2 – Computación evolutiva

Reseña histórica

UNIVERSIDAD DE LA REPÚBLICA URUGUAY

- Nils Barricelli (Princeton, entre 1953–1956) desarrolló las primeras simulaciones de un sistema evolutivo en una computadora digital. Diseñó un modelo computacional para la evolución darwiniana y estudió el rol de la simbiogénesis en el origen y desarrollo de la vida
- Implementó sobre código de máquina un universo de 512 celdas a ser ocupadas por números de 8 bits, cuya propagación era gobernada por un conjunto de reglas evolutivas
- Barricelli sentó varios precedentes: selección por aptitud y operador probabilístico de mutación, necesidad y utilidad de la recombinación

Nils Barricelli

2 – Computación evolutiva

Reseña histórica

UNIVERSIDAD DE LA REPÚBLICA URUGUAY

- George Friedman (1956) formuló una propuesta de aplicación de las técnicas evolutivas a la robótica, para evolucionar circuitos de control

  ("His MS Thesis on Selective Feedback Computers was the first identified paper published in the field of Evolutionary Computation").

George Friedman

Bio at MIT's Engineering Systems Division

- George Box (1957) propuso un enfoque evolutivo para la optimización en producción industrial (EVOP–Evolutionary Operation), basado en mecanismos dinámicos para ajustar variables de control
- Alexander Fraser (1957) publicó trabajos sobre la evolución de sistemas biológicos en una computadora digital
- Friedberg (1958) aplica aprendizaje por recompensa para evolucionar instrucciones. Enfoque criticado desde la IA

Reseña histórica

- Woodrow Bledsoe y Hans Bremermann (1960–1961) fueron precursores del concepto de algoritmos evolutivos. Interpretaron la evolución como un proceso de optimización y la aplicaron para mejorar algoritmos de reconocimiento de patrones con redes neuronales
  - Sugirieron la idea de codificación binaria, el uso de un valor de aptitud y el trabajo con poblaciones. Notaron la importancia de un operador de mutación para evitar estancamiento en óptimos locales
  - En 1965 Bremermann et al. presentaron el primer resultado teórico sobre la operativa de los algoritmos evolutivos, al determinar la probabilidad de mutación óptima para resolver problemas linealmente separables
- Newell y Simon (1963) propusieron el General Problem Solver, capaz de resolver sencillos problemas utilizando evolución guiada por heurísticas especificadas por el usuario. Primer abordaje evolutivo genérico, independiente del problema y dominio de aplicación

2 – Computación evolutiva

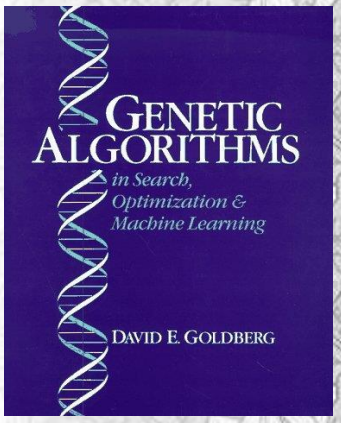
Reseña histórica

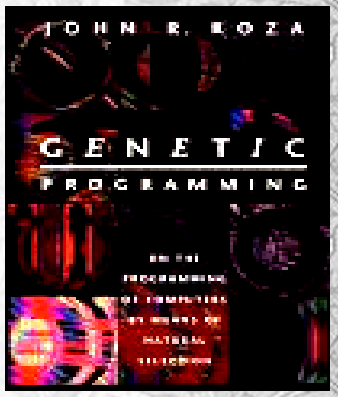
UNIVERSIDAD DE LA REPÚBLICA URUGUAY

- Ingo Rechenberg propuso en 1965 las estrategias de evolución (*Evolutionsstrategie*), que consisten en un método de ajustes discretos aleatorios inspirado en el mecanismo de mutación que ocurre en la naturaleza
- En 1966, Fogel, Owens y Walsh concibieron el uso de la evolución simulada en la solución de problemas mediante la técnica que dió origen a la programación genética
- Durante la década de 1960 John Holland planteó los “planes reproductivos” y “adaptativos” que imitan el proceso de la evolución. En 1975 su propuesta se formalizó en los algoritmos genéticos

Reseña histórica

- La década de 1980 fue testigo de la maduración de las técnicas de computación evolutiva
  - En 1985 se desarrolla la primer conferencia del área (ICGA)
  - En 1986, Hicklin y Fujiki evolucionaron expresiones condicionales LISP con el objetivo de resolver problemas en teoría de juegos
  - David Goldberg publicó en 1989 el primer texto sistematizador, que presenta los algoritmos genéticos, sus fundamentos teóricos y una variada gama de aplicaciones.
  - En el mismo año, John Koza propuso una codificación de árbol para representar programas, con un operador de cruce que intercambia sub-árboles, técnica que posteriormente, se denominó programación genética





Reseña histórica

- A inicios de la década de 1990, Thomas Ray desarrolló un simulador en el que evolucionaban programas en lenguaje ensamblador.
  - Los programas competían por ciclos de CPU, a la vez que intentaban reproducirse (copiarse en la memoria de la máquina)
  - Se partía de un programa único con la capacidad de auto-replicarse, a partir del cual se generaban nuevos segmentos de código, que podían subdivirse para dar origen a nuevos segmentos de código
  - Es uno de los pocos intentos por simular un “ecosistema” con el único propósito de observar los comportamientos que emergen de la dinámica evolutiva
  - La propuesta sugirió un nuevo rumbo (aún poco explorado) para la computación evolutiva

Definición y conceptos

- Engloba a un amplio conjunto de técnicas de resolución que siguen un mecanismo análogo a los procesos de evolución natural
- Esquema genérico de un algoritmo evolutivo (AE) trabajando sobre una población P

```
Inicializar(P(0));
generacion = 0;
mientras (no CriterioParada) hacer
    Evaluar(P(generacion));
    Padres = Seleccionar(P(generacion));
    Hijos = Aplicar Operadores Evolutivos(Padres);
    NuevaPoblacion = Remplazar(Hijos,P(generacion));
    generacion++;
    P(generacion) = NuevaPoblacion;
fin
retornar Mejor Solución Encontrada
```

Definición y conceptos

- Un AE trabaja sobre una población de *individuos* que representan soluciones potenciales al problema a resolver
  - La representación (individuo) es el *genotipo*, la solución el *fenotipo*
- Una *función de fitness* evalúa los individuos de acuerdo a su adecuacion para la resolución del problema
- La evolución consiste en un ciclo que consta de cuatro etapas:
  1. Evaluación: se asigna un valor de fitness a cada individuo
  2. Selección: se determinan candidatos adecuados para la generación de la nueva generación
  3. Aplicación de los operadores evolutivos: se genera un conjunto de descendientes a partir de los individuos seleccionados, mediante operadores que emulan la evolución natural
  4. Reemplazo: mecanismo que realiza el recambio generacional

Definición y conceptos

- La población es inicializada mediante un mecanismo aleatorio o guiado por heurísticas específicas
- Diversas políticas de seleccción y reemplazo permiten definir las características del algoritmo evolutivo
  - Privilegiar los individuos más adaptados (elitismo), aumentar la presión selectiva, incrementar la diversidad genética, etc.
- Los operadores evolutivos determinan el mecanismo de exploración del espacio de búsqueda del problema
  - Amplia gama: recombinación y mutación los más difundidos
  - Determinan las diferentes variantes de AE
- La condición de parada determina la finalización del AE
  - Número de generaciones, variación de valores de fitness, estimaciones del error cometido, etc.

Paradigmas

- Tres paradigmas fundamentales:
  - Algoritmos genéticos (Genetic algorithms – GA)
  - Programación genética (Genetic programming – GP)
    - Evolución de programas (estucturas con contenido semántico)
    - Evolución basada en selección estocástica y mutación
  - Estrategias de evolución (Evolution strategies – ES)
    - Evoluciona variables (en general números reales), del problema y de la propia técnica
    - Operadores específicos para el trabajo con números reales

3

UNIVERSIDAD DE LA REPÚBLICA URUGUAY

TÉCNICAS de COMPUTACIÓN EVOLUTIVA

Programación Genética

- Antecedente: programación automática
  - *"How can computers learn to solve problems without being explicitly programmed? In other words, how can computers be made to do what is needed to be done, without being told exactly how to do it?"* (A. Samuel, 1959)
- Programación evolutiva (Fogel, Owens y Walsh, 1966): método para evolucionar comportamientos de autómatas de estado finito para la predicción de series temporales
  - Enfoque "moderno": usa población y destaca importancia de la selección
- Holland (1975) sugirió utilizar AE con codificaciones que representaran programas. La idea se reformuló a mediados de 1980, permitiendo representaciones que extendieron su aplicabilidad
  - Programas evolutivos aplicados a heurísticas de optimización (Smith, 1985), al dilema del prisionero (Fujicki, 1986), y a la evolución de expresiones LISP para programas capaces de resolver juegos sencillos (Hicklin, 1986)

Programación Genética

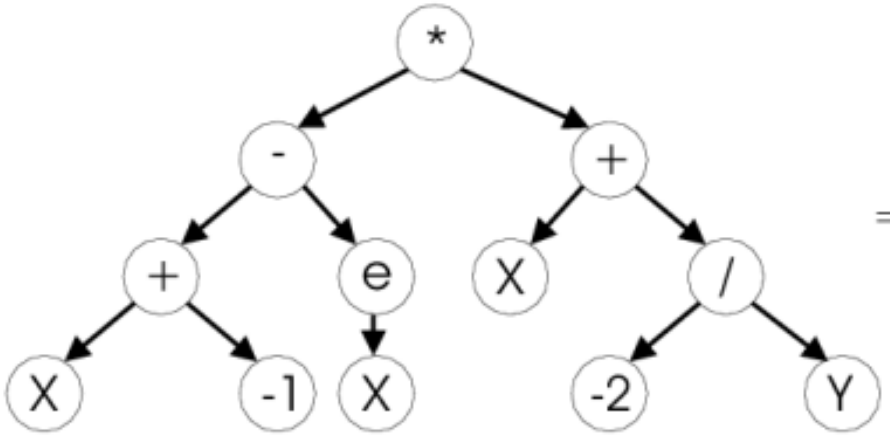
- John Koza (1996) propuso el mecanismo evolutivo de programación genética para generar programas, extendiendo ideas previas de Cramer sobre representaciones con árboles de parsing
- A partir de la propuesta de Koza, se investigó sobre la aplicabilidad de la técnica a un amplio conjunto de problemas, sobre sus fundamentos teóricos, y el diseño de operadores evolutivos adecuados
- En los últimos años, las líneas de trabajo en el área de la programación genetica se han concentrado principalmente en aplicaciones como el entrenamiento de redes neuronales y la evolución de sistemas difusos
- Algunos nuevos fundamentos matemáticos han sido presentados por parte del creador de las técnicas (Fogel, 1995)

Programación Genética

- Propone un mecanismo evolutivo para crear (síntesis o inducción) de programas de alto nivel
- Utiliza un conjunto de terminales para representar variables independientes del problema, funciones sin parámetros y constantes
- Maneja un conjunto de funciones primitivas que pueden ser aplicadas sobre los terminales
- Emplea cuatro operadores:
  - *Reproducción*: copia un programa a la nueva población
  - *Cruzamiento*: crea nuevo(s) programa(s) combinando partes de dos programas existentes
  - *Mutación*: crea un nuevo programa, modificando una parte de un programa ya existente
  - *Operación de cambio de arquitectura*: selecciona una operación de la biblioteca de este tipo de operaciones y la aplica a un programa

Programación Genética

- Ejemplo de codificación



$$\Rightarrow (x - 1 - e^x) \cdot (x + \frac{-2}{y})$$

- Representa el programa LISP (- (- x 1) (* x (* x x)))
- Orden de evaluacion izquierda-derecha.
- Operadores frecuentemente utilizados
  - Aritméticos: +, -, *, /, sin, cos, exp, etc.
  - Lógicos o Booleanos: AND, OR, XOR, NOT, NAND, etc.
  - Específicos del problema: Max, Min, varianza, etc.

Algoritmo de Programación Genética

1. Generar una población inicial a partir de la composición de funciones y terminales en forma aleatoria o heurística
2. Aplicar Iterativamente los pasos a) y b) a la población de programas hasta que se cumpla un criterio de parada
   a) Ejecutar cada programa de la población y asignarle su valor de fitness
   b) Crear una nueva población de programas a partir de aplicar las cuatro operaciones, sobre programas seleccionados con una probabilidad relacionada su fitness
3. Retornar el mejor programa encontrado

Autoestudio:

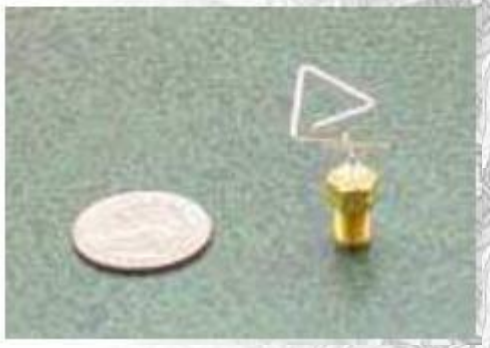
Evolución de programas LISP y autómatas celulares

Tema 1 en "Aplicación de AE a resolución de problemas"

http://www.fing.edu.uy/inco/cursos/geneticos/ae/2012/clases/pdf/clase13_2012.pdf

Programación Genética

- Aplicaciones
  - Problemas de control
  - Planificación
  - Regresión simbólica
  - Compresión de imágenes
  - Robótica
- En la lectura adicional recomendada:
  - Diseño de antenas
  - Robot para el juego de “crear palabras”



Estrategias de Evolución

- Propuestas en 1965 para la optimización de parámetros (números reales) en problemas hidrodinámicos de alto grado de complejidad
- Originalmente modelaba la evolución de los propios valores a optimizar, sin utilizar codificaciones, generando un único descendiente a partir de un único individuo padre (modelo (1+1)–ES)
- La base del mecanismo evolutivo es el operador de mutación
- Selección determinística (extintiva): Los peores individuos obtienen probabilidad cero de ser seleccionados. Un individuo solamente se mantiene si es mejor que su padre, de lo contrario se elimina
- Para la generación del nuevo individuo se utiliza una fórmula aritmética $\bar{x}^{t+1} = \bar{x}^{t} + N(0, \bar{\sigma})$, siendo t la generación y N(0,σ) una distribución gaussiana con media 0 y desviación estándar σ

Estrategias de Evolución

- En 1973, Rechenberg extendió su propuesta, introduciendo el modelo (μ+1)–ES, que incorpora el concepto de población
  - En este modelo, μ padres generan un único hijo, que reemplaza en la población a su peor padre
- En 1975, Schwefel propuso la generación múltiples de hijos
  - μ padres generan λ hijos
  - Modelo (μ+λ) – ES: selección entre padres e hijos (los μ mejores individuos de la unión de padres e hijos sobreviven)
  - Modelo (μ,λ) – ES: selección entre hijos únicamente (los μ mejores hijos sobreviven)

Estrategias de Evolución

- Esquema algorítmico

```
t = 0;
initialize(P(t=0));
evaluate(P(t=0));
while isNotTerminated() do
    P_p(t) = selectBest(μ, P(t));
    P_c(t) = reproduce(λ, P_p);
    mutate(P_c(t));
    evaluate(P_c(t));
    if (usePlusStrategy) then P(t+1) = P_c(t) ∪ P(t);
    else P(t+1) = P_c(t);
    t = t +1;
end
```

selección elitista

mutación es operador principal

Estrategias de Evolución: resultados teóricos

- Rechenberg formuló una regla para el ajuste de la desviación estándar de forma determinista durante el proceso evolutivo, asegurando la convergencia al óptimo.
  - Regla del éxito 1/5: establece que la razón entre mutaciones exitosas y el total de mutaciones debe ser 1/5. En caso de ser mayor, se debe incrementar la desviación estándar; en caso contrario, se debe decrementar
- En 1996, Thomas Bäck derivó una regla similar para el modelo (μ,λ) – ES (en este caso, la razón es 1/7).

Estrategias de Evolución

- Auto-Adaptación
  - Mecanismo mediante el cual además de evolucionar las variables del problema, también se evolucionan los parámetros de la técnica
  - Las ES suele incluir la autoadaptación de su parámetro de ajuste, la desviación estándar
- Operadores de recombinación
  - No incluidos en las primeras versiones de ES
  - Las ES simulan el proceso evolutivo a nivel de los individuos, por lo cual es posible la recombinación
    - Sexuales: actúan sobre dos individuos elegidos aleatoriamente de la población de padres
    - Panmícticos: se elige un padre al azar y se mantiene fijo mientras para cada componente se elige un segundo padre al azar de entre toda la población

Algoritmos Genéticos

- Enfatizan la importancia de del operador de recombinación sobre el de la mutación. La recombinación opera como operador principal, mientras que la mutación es un operador secundario
- En general utilizan selección probabilística
- La representación en el algoritmo genético tradicional es la cadena binaria

1	0	1	1	0	0	1	0	1

- Aplican un esquema orientado a maximizar una función de aptitud o *fitness*

Algoritmos Genéticos

- En su versión estándar, el esquema algorítmico de un AG sigue el patrón estándar de un AE, utilizando como operadores evolutivos a la recombinación y la mutación
- Originalmente aplicados para trabajar con codificaciones binarias para problemas de búsqueda en espacios con cardinalidad numerable. Se han extendido para contemplar codificación real y codificaciones no tradicionales, dependientes del problema a resolver (grafos, árboles, estructuras sintácticas)
- Ciertos autores han considerado al mecanismo de selección proporcional al fitness como característica distintiva de los AG. Sin embargo, existe una gran diversidad de propuestas que trabajan con otros mecanismos de selección

4

UNIVERSIDAD DE LA REPÚBLICA URUGUAY

COMPARACIÓN de TÉCNICAS de COMPUTACIÓN EVOLUTIVA

Comparación de enfoques

- Codificación
  - AG codifica las soluciones del problema (trabaja a nivel del genotipo)
  - GP y ES trabajan a nivel del fenotipo
- Selección
  - En general, AG y GP utilizan selección probabilística
  - ES utiliza selección determinística
- Operadores
  - En AG la recombinación es el operador principal, la mutación es un operador secundario. En general son no-adaptativos
  - GP tiene recombinación limitada por la semántica
  - ES se basa en la mutación, la recombinación es un operador secundario. En general implementan autoadaptación de sus parámetros

Comparación de enfoques

	Estrategias de evolución	Programación Genética	Algoritmos Genéticos
Representación	real	real	binaria/entera
Función de aptitud	valor de la función objetivo	*fitness,* vinculado con la función objetivo	*fitness,* vinculado con la función objetivo
Autoadaptación	desviaciones estándar y ángulos de rotación	ninguna varianzas (PE estándar) Coeficientes de correlación (meta-PE)	No se aplica
Mutación	gausiana, operador principal	específica	inversión de bits, operador secundario

Comparación de enfoques

	Estrategias de evolución	Programación Genética	Algoritmos Genéticos
Selección	determinista extintiva	probabilista extintiva	probabilística, basada en preservación
Recombinación	aritmética/gaussiana operador secundario	específica, semántica	n-puntos, uniforme, operador principal
Restricciones	arbitrarias de desigualdad	ninguna dadas por la semántica	límites en el mecanismo de codificación
Teoría	convergencia global para ($\mu+\lambda$)	convergencia global para meta-PE	teoría de esquemas, convergencia global para AG elitistas

Comparación con otras técnicas de búsqueda

- No existe opinión unánime sobre las ventajas de las técnicas de computación evolutiva
- Sin embargo, se han popularizado por su versatilidad para la resolución de problemas en muy variadas áreas de aplicación
- Características generales:
  - Altamente útiles para resolver problemas con espacio de soluciones de dimensión elevada o no muy bien comprendido de antemano, donde no es sencillo diseñar operadores específicos de hill-climbing y búsqueda local
  - Útiles en problemas multimodales donde las heurísticas tradicionales pueden quedar atrapadas en óptimos locales
  - Útiles en casos donde no es necesario obtener una solución óptima al problema, sino que una buena solución aproximada es suficiente

Comparación con otras técnicas de búsqueda

- Utilizan una población de soluciones, siendo menos sensibles a quedar atrapados en óptimos locales que las técnicas de solución única
- La búsqueda del mecanismo evolutivo es independiente de particularidades del dominio, y permite definir esquemas genéricos capaces de abordar diferentes clases de problemas. Además, se aplica sobre representaciones, aumentando la aplicabilidad de los algoritmos evolutivos a cualquier problema codificable
- No son sensibles a efectos de no linearidad de la función a optimizar u otras características del problema que usualmente afectan a los algoritmos estándar de optimización (que utilizan información adicional, como las técnicas de hill-climbing o las que asumen aspectos de linearidad, convexidad, diferenciabilidad, etc. sobre la función a optimizar). Esta característica brinda a los algoritmos evolutivos una significativa robustez de funcionamiento y de aplicabilidad

Comparación con otras técnicas de búsqueda

- No necesitan conocimientos específicos sobre el problema a resolver. El algoritmo evolutivo funciona como una caja negra que solamente utiliza como dato de entrada la función de fitness definida para el problema, para evaluar a los individuos en el ciclo evolutivo.
- Aunque no utiliza información adicional sobre las características del problema, el conocimiento específico puede incorporarse en la codificación o en los operadores evolutivos, para mejorar la búsqueda.
- Utilizan operadores probabilísticos (mientras que varias técnicas tradicionales utilizan operadores determinísticos), sin embargo recorren el espacio de soluciones en forma más inteligente que la búsqueda aleatoria (*random walk*).

Principales ventajas

- Simplicidad conceptual del mecanismo de exploración
- Amplia aplicabilidad
- Superiores a métodos tradicionales en muchos problemas reales
- Pueden incorporar conocimiento sobre el dominio del problema
- Pueden hibridizarse con otras técnicas de búsqueda
- Son robustas a los cambios dinámicos
- En general, pueden auto-adaptar sus parámetros
- Explotan de modo natural las ventajas del procesamiento paralelo

Desventajas

- Han sido duramente criticadas por los investigadores de la IA simbólica por considerarlas, en forma equivocada, inferiores a las búsquedas aleatorias
  - La programación automática fue considerada una “moda pasajera” y el enfoque evolutivo fue visto como otro intento por lograr lo imposible
  - Los especialistas de IA clásica las consideran “mal fundamentadas” e “inestables”
- Su alta aplicabilidad las hace poco eficientes para algunos tipos específicos de problemas (*No Free Lunch Theorem*)
- Sin incorporar conocimiento del problema, pueden tener problema para hallar los óptimos globales
- Su eficiencia computacional puede ser pobre, demandando grandes recursos de cómputo para problemas complejos

No Free Lunch

- Es un teorema en el área de la optimización combinatoria demostrado por David Wolpert y William Macready en 1995
- Establece que todos los algoritmos que buscan optimizar (hallar un máximo o un mínimo) una función se comportan de la misma forma al ser promediados sobre todas las posibles funciones a optimizar
- El teorema justifica por qué ante el conjunto de todos los problemas de optimización matemáticamente posibles, en promedio, los algoritmos se comportarán exactamente de la misma manera
- Habitualmente se utiliza como fundamento en contra de la utilización de algoritmos genéricos de búsqueda (como los algortimos evolutivos) cuando son utilizados con muy poco conocimiento (e inclusive sin conocimiento alguno) del dominio del problema

No Free Lunch

- “A general-purpose universal optimization strategy is theoretically impossible, and the only way one strategy can outperform another is if it is specialized to the specific problem under consideration" (Ho and Pepyne, “Simple Explanation of the No-Free-Lunch Theorem and Its Implications”, 2002)
- Puede ser visto como una justificación para la utilización de la mayor cantidad posible de conocimiento del dominio y por tanto realizar ajustes específicos para las diferentes rutinas dependiendo del dominio

Pero ¿ y entonces ? …

No Free Lunch

- Más allá de la correctitud matemática de la demostración, su enunciado alude a la totalidad de los problemas de optimización.
- No es posible aplicarlo sobre un subconjunto de las funciones a optimizar, por lo cual en principio su aplicabilidad es reducida
- ¿Es posible que exista un algoritmo general que sea mejor para los casos de funciones de aquellos problemas de optimización del mundo real que son de interés para resolver?

El NFTL no contesta esta pregunta !!

- Una alternativa es hibridizar entre sí una metaheuristica genérica (por ejemplo, algoritmos evolutivos) con otras que incorporen información específica del dominio de un determinado problema (por ejemplo, una búsqueda local)