

CI5438. Inteligencia Artificial II

Clase 8: Algoritmos Evolutivos

Cap 2 Eiben and Smith

Ivette C. Martínez

Universidad Simón Bolívar

2 de Noviembre de 2009

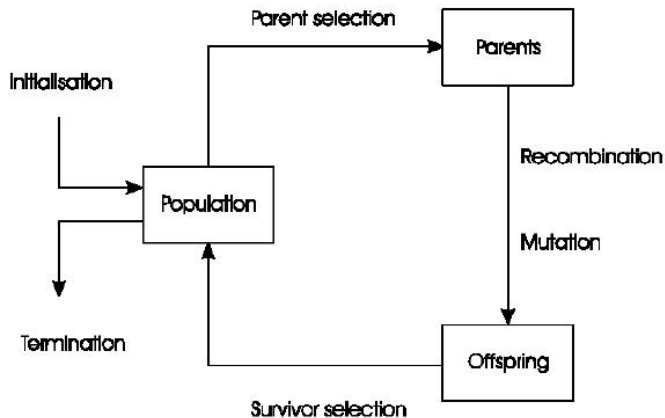
La metáfora de la Computación Evolutiva

- Una población de individuos existe en un ambiente con recursos limitados.
- La **Competencia** por estos recursos causa que se seleccionen los “mejores” (fitter) individuos, que son aquellos que mejor se adaptan al ambiente.
- Estos individuos actúan como semilla para la generación de nuevos individuos a través de la recombinación y la mutación.
- Se evalúa el *fitness* de los nuevos individuos, y estos compiten (posiblemente con sus padres) por sobrevivir.
- Con el tiempo la **Selección Natural** causa el incremento en el fitness de la población.

La metáfora de la Computación Evolutiva

- Los Algoritmos Evolutivos pertenecen a la categoría de algoritmos de “generación y prueba” (*generate and test*).
- Son algoritmos estocásticos, basados en poblaciones.
- Los operadores de variación (recombinación y mutación) crean la diversidad necesaria, y por la tanto facilitan la novedad.
- La selección reduce la diversidad y actúa como una fuerza que empuja la calidad.

Esquema General de los AEs



Pseudo Código de un AE típico

```
BEGIN
  INITIALISE population with random candidate solutions;
  EVALUATE each candidate;
  REPEAT UNTIL ( TERMINATION CONDITION is satisfied ) DO
    1 SELECT parents;
    2 RECOMBINE pairs of parents;
    3 MUTATE the resulting offspring;
    4 EVALUATE new candidates;
    5 SELECT individuals for the next generation;
  OD
END
```

Cuáles son los diferentes tipos de EAs

- Históricamente los diferentes tipos de EAs se han asociado con diferentes representaciones:
 - Strings Binarios: Algoritmos Genéticos
 - Vectores de valores reales: Estrategias Evolutivas
 - Máquinas de Estado Finito: Programación Evolutiva
 - árboles LISP: Programación Genética
- Estas diferencias son “prácticamente” irrelevantes, la mejor estrategia:
 - Escoger la representación adecuada para el problema.
 - Escoger los operadores de variación más adecuados para esta representación.
- Los operadores de selección solo usan el *fitness*, por lo que son independientes de la representación.

- Las soluciones candidatas (**individuos**) existen en el *espacio fenotípico*.
- Estas se codifican en **cromosomas**, que existen en el espacio genotípico.
 - Codificación: fenotipo \Rightarrow genotipo (no necesariamente uno a uno)
 - Decodificación: genotipo \Rightarrow fenotipo (debe ser uno a uno)
- Los cromosomas contienen **genes**, los cuáles se alojan en posiciones (usualmente fijas) llamadas **loci**, y tienen un valor (**alelo**).

Para encontrar el óptimo global, todas las soluciones posibles deben ser representadas en el espacio genotípico.

Función de Evaluación (Fitness)

- Representa los requerimientos a los cuáles se debe adaptar la población.
- es decir, es una función de *calidad* u *objetivo*.
- Asigna un único valor real a cada fenotipo el cuál es la base para la selección
 - Luego, A mayor discriminación (valores diferentes) mejor
- Típicamente hablaremos de maximización del fitness
 - Algunos problemas pueden plantearse mejor como una minimización, pero la conversión es trivial.

- Contiene (representaciones de) posibles soluciones.
- Usualmente tiene tamaño fijo y es un multiconjunto de genotipos
- Algunos EAs sofisticados también incorporan estructura espacial sobre las poblaciones, ejm: un grid.
- Los operadores de selección usualmente toman en cuenta toda la población; i.e., las probabilidades de reproducción son relativas a la generación actual.
- La **Diversidad** de una población se refiere al número de diferentes fitnesses/fenotipos/genotipos presentes (nótese que no son la misma cosa).

Mecanismos de selección de padres

- Asigna probabilidades a los individuos a actuarán como padres dependiendo de sus “fitnesses” .
- Usualmente es probabilístico:
 - Soluciones de alta calidad son mas propensas a convertirse en padres que las de baja calidad
 - pero no está garantizado
 - incluso el peor de la población actual tiene una probabilidad diferente de cero de convertirse en padre.
- Esta naturaleza *estocástica* puede ayudar a escapar de óptimos locales.

Operadores de variación

- Su rol es generar nuevas soluciones candidatas
- Usualmente se dividen en dos tipos de acuerdo con su aridad (número de entradas):
 - Aridad 1: operadores de mutación
 - Aridad > 1 : operadores de recombinación
 - Aridad = 2: **crossover**
- Ha habido muchos debates sobre la importancia relativa de la recombinación y la mutación
 - Hoy en día la mayoría de los AEs usan ambas.
 - La escogencia de un operador de variación en particular depende de la representación.

- Actúa sobre un genotipo y devuelve otro
- Su aleatoriedad es esencial y lo diferencia de otros operadores heurísticos unarios.
- Su importancia depende de la representación y el dialecto:
 - AGs Binarios: operador base responsable de preservar e introducir diversidad.
 - EP para Máquinas de Estado Finitos s/ variables continuas - solo es un operador de búsqueda
 - GP - difícilmente usado
- Puede garantizar la conectividad del espacio de búsqueda y por lo tanto las pruebas de convergencia.

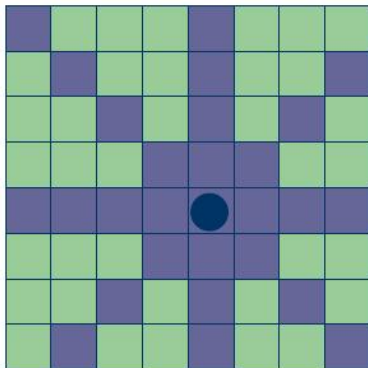
- Mezcla información de padres en su descendencia.
- La escogencia de que información mezclar es estocástica.
- la mayoría de los descendientes puede ser peor, o igual que sus padres.
- Se espera que algunos sean mejores mediante la combinación de elementos de genotipos que lleven a buenos razgos.
- Este principio se ha usado durante milenios por los criadores de plantas y animales.

Selección de sobrevivientes

- También conocido como **reemplazo**
- La mayoría de los EAs usan un tamaño de población fijo, luego necesitan una manera de ir de (padres + hijos) a una nueva generación.
- A menudo determinístico
 - Basado en fitness: jerarquizar padres + hijos y tomar los mejores.
 - Basado en edad: Hacer tantos descendientes como padres y eliminar todos los padres.
- Algunas veces combinar (elitismo).

- La inicialización usualmente es aleatoria,
 - Necesidad de garantizar una difusión y mezcla parejas de los posibles valores de los alelos.
 - Puede incluir soluciones existentes, o usar heurísticas específicas al problema para “sembrar” la población.
- La condición de terminación se verifica que cada generación
 - Alcanzar algún fitness (conocido/deseado)
 - Alcanzar un máximo número de generaciones.
 - Alcanzar un nivel mínimo de diversidad.
 - Alcanzar cierto número de generaciones sin mejoras en el fitness.

Ejemplo: 8-reinas

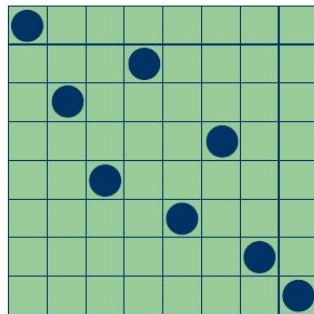


Colocar 8 reinas en un tablero 8x8 de tal forma que no se ataquen entre si.

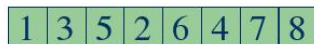
8-reinas: representación

Phenotype:
a board configuration

Genotype:
a permutation of
the numbers 1 - 8



Obvious mapping



8-reinas: Evaluación del fitness

- Penalidad para una reina:
El número de reinas que puede atacar
- Penalidad para una configuración:
La suma de las penalidades de todas las reinas
- Nota: Queremos minimizar la penalidad
- Fitness de una configuración:
El inverso de penalidad

8-reinas: mutación

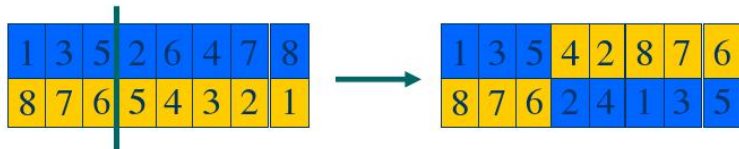
Pequeñas variaciones en una permutación:
ejm: intercambiar los valores de dos posiciones escogidas aleatoriamente.



8-reinas: recombinación

Combinar dos permutaciones en dos nuevas permutaciones:

- escoger un punto de crossover aleatorio
- copiar las primeras partes en los hijos
- crear las segundas partes insertando valores del otro padre:
 - en el orden en que aparecen aquí
 - comenzando después del punto de crossover
 - saltando los valores que ya están en el hijo



- Selección de padres:
 - Escoger 5 padres y tomar los mejores 2 para hacer crossover
- Selección de sobrevivientes (reemplazo)
 - Cuando se inserta un nuevo hijo a la población, escoger un miembro existente para ser reemplazado:
 - Ordenando la población por fitness en orden decreciente
 - Reemplazando el primero de esta lista que tenga un fitness menos que el hijo en cuestión

Representation	Permutations
Recombination	"Cut-and-crossfill" crossover
Recombination probability	100%
Mutation	Swap
Mutation probability	80%
Parent selection	Best 2 out of random 5
Survival selection	Replace worst
Population size	100
Number of Offspring	2
Initialisation	Random
Termination condition	Solution or 10,000 fitness evaluation

Notar que este es sólo uno de los posibles conjuntos de opciones de operadores y parámetros

Comportamiento típico de un EA

Fases en la optimización sobre un paisaje de fitness (fitness landscape) de 1 dimensión.



Early phase:
quasi-random population distribution

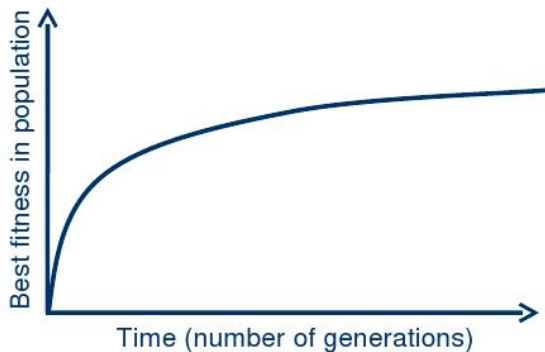


Mid-phase:
population arranged around/on hills



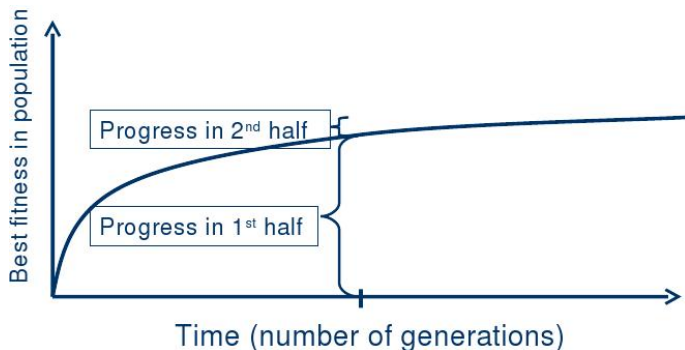
Late phase:
population concentrated on high hills

Corrida típica: la progresión del fitness



Corrida típica de un EA, muestra el denominado “anytime behavior”

Son beneficiosas las corridas largas?



Respuesta:

- Depende de cuánto se quiera el último “pedacito” de progreso
- Puede ser mejor hacer mas corridas cortas

