

Programación Evolutiva

Tema 8: Extensiones de los algoritmos genéticos. Otros temas

Carlos Cervigón, Lourdes Araujo. 2009-2010

Extensiones a los algoritmos genéticos

- ❑ Extensiones del modelo básico de Algoritmos Genéticos.
- ❑ Conservan las ideas fundamentales de los AGs.
- ❑ Introducen cambios en la estructura básica.
 - Algoritmos Genéticos con edades
 - AGs con mecanismos de escisión
 - Subpoblaciones por agregación
 - Búsqueda Multiobjetivo
 - Algoritmos Genéticos Paralelos
 - Algoritmos Genéticos Diploides

Algoritmos Genéticos con edades

- ❑ Al crear un individuo
 - Se le asigna una duración, que depende de su aptitud.
 - Se inicializa a cero su contador de edad.
- ❑ Tras cada generación
 - Se incrementa la edad de cada individuo.
 - Se eliminan los que han completado su duración.

Algoritmos Genéticos con edades

Aparte del parecido con la naturaleza encontramos las siguientes motivaciones:

- ❑ Mantener el control de la edad es suficiente para controlar la presión selectiva, haciendo innecesaria la selección de criadores.
- ❑ La introducción de la edad hace variable el tamaño de la población, proporcionando un mecanismo de autorregulación de este tamaño.

- En los AG con edades no hay selección.
- El reemplazo elimina los individuos que han sobrepasado su duración.
- El tamaño de la descendencia depende del tamaño actual de la población:

$$tam_descendencia[t] := \lfloor \rho \cdot tam_pob[t] \rfloor$$

- Donde ρ es el coeficiente de reproducción (valores próximos a 0,4).
- La calidad del AG mejora al aumentar ρ a costa de incrementar el coste computacional.

```
funcion AGConEdad(){
P[0] = Poblacion_inicial(); //asignando edad
AptP[0] = Evaluacion(P[0]);
EdadesP[0] = Edad_poblacion(P[0]);
t = 0;
mientras (t < Num_max_gen) y no CondTermina(P[t], AptP[t]){
    EdadesP[t]++;
    Q[t] = Reproduccion(P[t], ρ); //al azar
    Q[t] = Mutacion(Q[t]);
    P[t] = Mezclar(P[t], Q[t]);
    AptP[t] = Evaluacion(P[t]);
    EdadesP[t] = Nuevas_edades(P[t]);
    P[t] = EliminarAntiguos(P[t], AptP[t], EdadesP[t]);
    t++;
}
```

- Como no hay selección todos los individuos de la población pueden ser elegidos como progenitores de forma equiprobable.
- A cada individuo se le asigna su duración al ser evaluado por primera vez (valor que permanece constante hasta que el individuo desaparece por superar su duración).
- Tamaño de la población en la t-ésima generación:

$$tam_pob[t + 1] =$$

$$tam_pob[t] + tam_descendencia[t] - Eliminados[t]$$

- Donde $Eliminados[t]$ es el número de individuos que han rebasado su duración en la iteración

- Lo más complejo es la asignación de duraciones:
- Requisitos del criterio de asignación de duraciones:
 - Reforzar la presencia de los más aptos
 - Controlar el tamaño de la población.
- La descendencia de un individuo es directamente proporcional a su duración: se asignan duraciones más altas a los individuos más aptos.
- La duración no debe ser estrictamente proporcional a la aptitud para no violar el segundo requisito.
- Al calcular la duración de un individuo conviene considerar, además de la aptitud, el estado de la búsqueda, dado por las aptitudes media y extremas de la población.

3 estrategias:

Asignación proporcional

$$TiempoVida(\mathbf{v}) = \min\{MaxTV, MinTV + DifTV \frac{Aptitud(\mathbf{v})}{AptMed}\}$$

Asignación lineal

$$TiempoVida(\mathbf{v}) = MinTV + 2 DifTV \frac{Aptitud(\mathbf{v}) - AptMin}{AptMax - AptMin}$$

- Las cotas a las duraciones $MaxTV$ y $MinTV$ son parámetros con valores por defecto 7 y 1. $DifTV := \frac{1}{2}(MaxTV - MinTV)$

Asignación bilineal

$$TiempoVida(\mathbf{v}) =$$

$$\begin{cases} MinTV + DifTV \frac{Aptitud(\mathbf{v}) - AptMin}{AptMed - AptMin} & \text{sii } Aptitud(\mathbf{v}) \leq AptMed \\ \frac{1}{2}(MinTV + MaxTV) + DifTV \frac{Aptitud(\mathbf{v}) - AptMed}{AptMax - AptMed} & \text{sii } Aptitud(\mathbf{v}) > AptMed \end{cases}$$

- Con la asignación proporcional las probabilidades de supervivencia son proporcionales a su aptitud relativa.
- La cota $MaxTV$ se añade para evitar el crecimiento descontrolado de la población.
- La asignación lineal se introduce porque la experiencia indica que conviene considerar también la aptitud de los individuos respecto a la mejor aptitud obtenida hasta entonces (aptitud objetiva):

$$AptObj(\mathbf{v}) := \frac{Aptitud(\mathbf{v})}{AptMax(P)}$$

- La asignación lineal tiene el inconveniente de que cuando hay muchos individuos con aptitudes similares a la mejor, se asignan largas duraciones a todos ellos (crece significativamente el tamaño de la población).
- Con la estrategia de asignación bilineal se trata de rebajar esta tendencia.
- La estrategia de asignación lineal es la que proporciona mayor precisión, pero es la más costosa.
- La menos costosa es la bilineal, pero no tiene buena precisión.
- La asignación proporcional representa un punto intermedio.

Tamaño de población en AG con edades

- En un AG con edades el tamaño de la población evoluciona de forma oscilante:
 - Al principio, cuando la variedad de aptitudes es relativamente alta, el tamaño de la población crece (búsqueda en anchura del óptimo).
 - Una vez que ha sido localizada la vecindad del óptimo, el algoritmo comienza a converger y el tamaño de la población se reduce.
 - Cuando se consigue una mejoría tiene lugar otra explosión demográfica, seguida de otra etapa de convergencia.

Algoritmos con mecanismos de escisión

- Permiten trabajar con grupos de individuos con capacidades similares (aptitud en cierta región del espacio de búsqueda).
- Se aplican en problemas de búsqueda múltiple:
 - **Búsqueda exhaustiva:** Se trata de localizar todos los óptimos de una función objetivo, (o todos los puntos que verifican cierto criterio de validez).
 - **Búsqueda multiobjetivo :** Tratan de optimizar simultáneamente varios objetivos (función objetivo vectorial).

Algoritmos con mecanismos de escisión

- Los problemas de **búsqueda exhaustiva** se tratan estimulando la formación de subpoblaciones estables en torno a cada uno de los óptimos, de modo que según progrese el AG más cerca se encuentren los individuos de su respectivo óptimo.
- Principales procedimientos:
 - por agregación
 - por segregación
- Ambas hacen uso de la idea de aptitud compartida, que se basa en la de distancia entre individuos (a nivel de genotipo o de fenotipo).
- Se trata de hacer que la aptitud de un individuo se reduzca al compartir los recursos de su entorno con sus vecinos.

Búsqueda multiobjetivo

- En muchos problemas se necesita optimizar simultáneamente varios objetivos $f_1(\mathbf{x}), \dots, f_r(\mathbf{x})$
- Matemáticamente:

$$\begin{cases} \text{maximizar} & f_k(\mathbf{x}) \quad (\forall k = 1, \dots, r) \\ \text{sujeto a} & \mathbf{x} \in \mathcal{X} \subseteq \mathcal{R}^m \end{cases}$$
- A veces los objetivos $f_k(\mathbf{x})$, admiten una base común de medidas (por ejemplo, costes o beneficios equivalentes).
 - El problema puede reducirse a uno monoobjetivo mediante una **función combinada de objetivos**:

$$f(\mathbf{x}) := w_1 f_1(\mathbf{x}) + \dots + w_r f_r(\mathbf{x})$$
 - Son los factores de ponderación, que por convenio la suma de los factores de ponderación vale 1.
 - La elección de los w_k es fundamental

Búsqueda multiobjetivo

- ❑ Usualmente los objetivos no pueden reducirse a una base común.
- ❑ **Técnica de las subpoblaciones:**
 - Se utiliza una población dividida en subpoblaciones de igual tamaño, cada una de ellas responsable de un único objetivo.
 - La selección se realiza dentro de cada subpoblación.
 - El cruce se realizaba entre subpoblaciones.
 - Resultados mediocres.
- ❑ Los mejores resultados se obtienen con la búsqueda de la soluciones no dominadas.

Búsqueda multiobjetivo

- ❑ Soluciones no dominadas u óptimos de Pareto:
- ❑ Una solución $x \in \mathcal{X}$ es un óptimo de Pareto cuando no existe ninguna solución mejor en ningún objetivo:

$$\nexists y \in \mathcal{X}$$

tal que

- $f_i(y) \geq f_i(x) (\forall i = 1, \dots, k)$ y
- $f_j(y) > f_j(x)$ para algún j .

- ❑ Se espera que las mejores soluciones estén entre ellas.

Búsqueda multiobjetivo

- ❑ Uso de las soluciones no dominadas:
 - Se ordenan los individuos en función de su grado de dominio en la población: a los individuos no dominados se les da el primer puesto, el segundo se da a los individuos no dominados por otros individuos más que por los del primer puesto, etc.
 - Esos índices se usan como aptitudes brutas de los respectivos individuos
 - Las técnicas de escisión se introducen para mantener un alto grado de diversidad a lo largo de todo el proceso.

Algoritmos genéticos paralelos

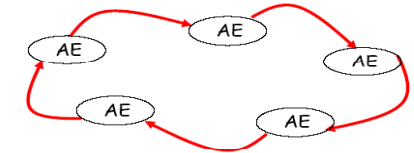
- ❑ El coste computacional de los AG suele ser alto → procesamiento paralelo.
- ❑ Principales modelos de paralelización de un AG:
 - Modelos **centralizados o en granja**
 - Modelos **distribuidos o en isla**
 - Modelos **masivamente paralelos o celulares**

Modelos centralizados o en granja

- ❑ Un **proceso central** coordina el funcionamiento de varios procesos subordinados : Cada procesador hace una parte de cada paso del algoritmo (selección, cruce y mutación) sobre la población común.
- ❑ El proceso central hace la selección de las operaciones que realizan los restantes procesos.
- ❑ **Paralelismo de grano fino**: suele implementarse en memoria compartida.
- ❑ Los resultados son los mismos que los de la ejecución secuencial, sólo cambia en tiempo.
- ❑ Inconvenientes inherentes al centralismo:
 - Necesidad de sincronismo (posibles tiempos de espera).
 - Fiabilidad dependiente del proceso central.

Modelos distribuidos o en isla

- ❑ Un **conjunto de subpoblaciones** evolucionan independientemente en cada procesador o isla.
- ❑ En cada procesador se lleva a cabo un Algoritmo Genético
- ❑ En cada isla la población inicial se genera con diferente semilla.
- ❑ Cada cierto número de generaciones (epoch) intercambian individuos entre las poblaciones: migraciones. Esto genera diversidad genética.
- ❑ Podemos emplear distintos operadores en cada isla



Modelos distribuidos o en isla

- ❑ Parámetros adicionales:
 - Número de generaciones entre intercambios.
 - Número de individuos a intercambiar.
 - Selección de los individuos a enviar.
 - Selección de los individuos a reemplazar por los recibidos.
- ❑ Paralelismo de grano grueso.
 - Se suelen implementar en memoria distribuida.
 - Implementación sencilla.
 - Mejoran la calidad de las soluciones, además del tiempo

Modelos distribuidos o en isla

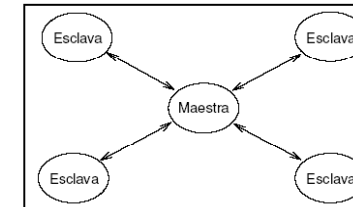
- ❑ ¿Cada cuánto tiempo debemos intercambiar individuos?
 - Demasiado rápido y todas las poblaciones convergen hacia la misma solución. Demasiado lento y tomará mucho tiempo lograr una solución
 - Número habitual de generaciones por *epoch*: entre 25 y 150
- ❑ ¿Cuántos y cuáles individuos intercambiar?
 - Depende del tamaño de la población (normalmente entre 2 y 5)
 - Normalmente, un mayor número de poblaciones proporciona mejores resultados
 - Es mejor migrar individuos seleccionados al azar que siempre los mejores
 - Para el reemplazo podemos seleccionar los individuos aleatoriamente o tomar los peores

Modelos distribuidos o en isla

- ❑ Se pueden distinguir diferentes modelos de islas en función de la comunicación entre las subpoblaciones.
- ❑ Modelos típicos:
 - Estrella
 - En red
 - En anillo

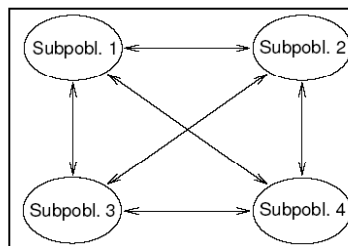
Modelo en estrella

- ❑ Una subpoblación es seleccionada como maestra (la que tiene mejor media en el valor de la función objetivo), siendo las demás consideradas como esclavas.
- ❑ Todas las subpoblaciones esclavas mandan sus **$h1$** mejores individuos (**$h1 > 1$**) a la subpoblación maestra la cual a su vez manda sus **$h2$** mejores individuos (**$h2 > 1$**) a cada una de las subpoblaciones esclavas.



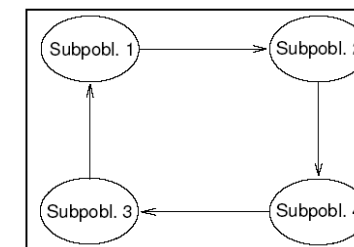
Modelo en red

- ❑ No hay una jerarquía entre las subpoblaciones; todas mandan sus **$h3$** (**$h3 > 1$**) mejores individuos al resto de las subpoblaciones.



Modelo en anillo

- ❑ Cada subpoblación envía sus **$h4$** mejores individuos (**$h4 > 1$**), a una población vecina, efectuándose la migración en un único sentido de flujo.



Modelos masivamente paralelos

- ❑ Usan una disposición espacial de los individuos en una única población.
- ❑ Las operaciones genéticas tienen lugar en paralelo en cada nodo de la población:
 - Cada individuo sólo interacciona con sus vecinos.
- ❑ El reemplazamiento destruye al individuo considerado sobrescribiéndolo con el nuevo.
- ❑ El aislamiento de los individuos produce una gran diversidad y baja presión selectiva.

Algoritmos genéticos diploides

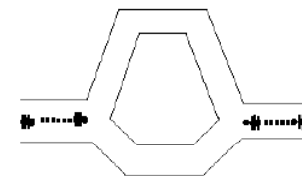
- ❑ Son AGs con representación duplicada (en biología diploides).
- ❑ En cada gen hay lugar para dos alelos.
- ❑ Uno de los alelos es dominante y se expresará al calcular la aptitud del individuo.
- ❑ En la reproducción, cada individuo hereda en cada gen un alelo al azar de cada uno de sus progenitores.
- ❑ La duplicación de los alelos evita la destrucción prematura de información que pudiera resultar valiosa en el futuro.
- ❑ Especialmente útil cuando se trabaja con funciones de aptitud cambiantes en el tiempo.

Otros temas : Colonias de hormigas

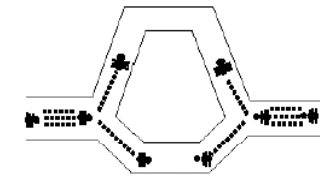
- ❑ Los modelos basados en colonias de hormigas se basan en el comportamiento colectivo de las hormigas en la búsqueda de alimentos para su subsistencia.
- ❑ Insectos casi ciegos, moviéndose aproximadamente al azar, pueden encontrar el camino más corto desde su nido hasta la fuente de alimentos y regresar.
- ❑ Cuando una hormiga se mueve, deja una señal odorífera, depositando una sustancia denominada feromona, para que las demás puedan seguirla.
- ❑ En principio, una hormiga aislada se mueve esencialmente al azar, pero las siguientes deciden con una buena probabilidad seguir el camino con mayor cantidad de feromonas.

Colonias de hormigas

- ❑ En las siguientes figuras se observa como las hormigas establecen el camino más corto.
- ❑ En la figura (a) las hormigas llegan al punto en que tienen que decidir por uno de los caminos que se les presenta.



(a)

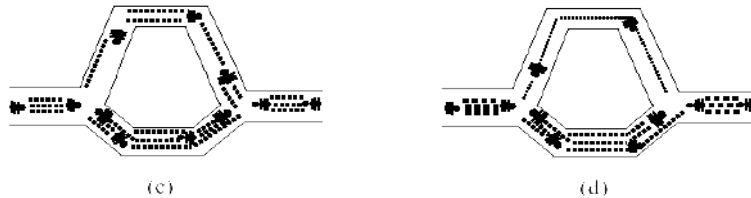


(b)

- ❑ en (b) realizan la elección de manera aleatoria, algunas hormigas eligen el camino hacia arriba y otras hacia abajo

Colonias de hormigas

- En (c) como las hormigas se mueven aproximadamente a una velocidad constante, las que eligieron el camino más corto alcanzarán el otro extremo más rápido que las otras que tomaron el camino más largo, depositando mayor cantidad de feromona por unidad de longitud; en (d) la cantidad de feromona depositada en el trayecto más corto hace que la mayoría de las hormigas elijan este camino, por realimentación positiva, en la que la probabilidad con la que una hormiga escoge un camino aumenta con el número de hormigas que previamente hayan elegido el mismo camino.



Colonias de hormigas

- Se ha utilizado para el problema del viajante (Traveling Salesman Problem - TSP).
- Los algoritmos ACO son procesos iterativos. En cada iteración se "lanza" una colonia de m hormigas y cada una de las hormigas de la colonia construye una solución al problema.
- Las hormigas construyen las soluciones de manera probabilística, guiándose por un rastro de feromona artificial y por una información calculada a priori de manera heurística. La regla probabilística para el caso del TSP es:

$$p_{ij}^k(t) = \frac{[\tau_{ij}(t)]^\alpha \cdot [\eta_{ij}]^\beta}{\sum_{l \in N_i^k} [\tau_{il}(t)]^\alpha \cdot [\eta_{il}]^\beta} \quad \text{con } j \in N_i^k$$

Colonias de hormigas

- donde $p_{ij}^k(t)$ es la probabilidad con la que, en una iteración t del algoritmo, la hormiga k , situada actualmente en la ciudad i , elige a la ciudad j como próxima parada.
- N_i^k es el conjunto de ciudades no visitadas todavía por la hormiga k .
- $\tau_{ij}(t)$ es la cantidad de feromona acumulada sobre el arco (i,j) de la red en la iteración t .
- η_{ij} es la información heurística para la que, en el caso del TSP, se utiliza la inversa de la distancia existente entre las ciudades i y j .
- α y β son dos parámetros del algoritmo, los cuales hay que ajustar.

Colonias de hormigas

- Cuando todas las hormigas han construido una solución debe actualizarse la feromona en cada arco. La fórmula a seguir es:

$$\tau_{ij}(t+1) = (1 - \rho) \cdot \tau_{ij}(t) + \Delta \tau_{ij}^{best} \quad \Delta \tau_{ij}^{best} = \begin{cases} 1/L^{best} & \text{si el arco } (i,j) \text{ pertenece a } T^{best} \\ 0 & \text{en caso contrario} \end{cases}$$

- donde ρ es el coeficiente de evaporación de la feromona.
 T^{best} puede ser la mejor solución encontrada hasta el momento o bien la mejor solución encontrada en la iteración.
 L^{best} es la longitud de la solución T^{best}

- ❑ Se obliga a que el nivel de feromona esté en el rango $[\tau_{\min}, \tau_{\max}]$
- ❑ Estos límites se imponen con el objetivo de evitar el estancamiento en la búsqueda de soluciones. Toda la feromona se inicializa con τ_{\max}
- ❑ Tras la actualización de la feromona puede comenzarse una nueva iteración.
- ❑ El resultado final es la mejor solución encontrada a lo largo de todas las iteraciones realizadas.