

Técnicas, Entornos y Aplicaciones de Inteligencia Artificial

Práctica 3: Algoritmos genéticos

Ramon Ruiz Dolz
4CO21-2017

1. Introducción

En esta memoria procederemos al diseño de un algoritmo genético para el problema de la asignación de recursos. Concretamente un problema donde tenemos 3 lectores y 3 periódicos, dónde queremos que todos los periódicos sean leídos por todos los lectores en el menor tiempo posible, teniendo un solo periódico disponible a la vez y no pudiendo compartirlo mientras lo leen. El ejemplo propuesto se plasma en la siguiente tabla, en la última parte de la memoria realizaremos una iteración del algoritmo con estos datos de ejemplo.

	P1	P2	P3
L1	10	15	25
L2	18	20	15
L3	20	30	10

En un algoritmo genético podemos destacar los siguientes elementos relevantes: el individuo tipo que compondrá la población, la función fitness que definirá el “valor” de este individuo, los criterios de selección y de cruce, los criterios de mutación, el criterio de reemplazo de la población y finalmente el motivo de parada del algoritmo. En las siguientes secciones procederemos al diseño de todos estos elementos.

2. El individuo

Para este problema, el individuo diseñado ha sido el siguiente, para n lectores y m periódicos, siendo cada lista un lector i, y cada elemento de la lista un periódico j.

$$([P_{11}, P_{12}, \dots, P_{1m}]_1, [P_{21}, P_{22}, \dots, P_{2m}]_2, \dots, [P_{n1}, P_{n2}, \dots, P_{nm}]_n)$$

Como podemos observar, esto es un individuo generalizado para este problemas, podemos concretarlo limitando el número de lectores a 3 y el número de periódicos a 3 también, obteniendo el siguiente tipo de individuo.

$$([P_{11}, P_{12}, P_{13}]_1, [P_{21}, P_{22}, P_{23}]_2, [P_{31}, P_{32}, P_{33}]_3)$$

En este caso siendo P_{11} por ejemplo, el periódico que lee el lector 1 en el momento 1 pudiendo este ser el periódico P1, P2 o P3.

3. La función fitness

Este problema es un problema de minimización, lo que buscamos es minimizar el tiempo de lectura, $F = \min(\text{fitness}(\text{individuos}))$. Por lo tanto, la función fitness consistirá en el máximo de los tiempos de lectura de los 3 lectores.

$$\text{fitness} = \max(T_{L1}'', T_{L2}'', T_{L3}'')$$

Podemos definir los tiempos de lectura de cada lector de la siguiente forma, teniendo en cuenta que habrá solapamientos y no se reducirá a la suma de los tiempos de lectura de los periódicos. Para n lectores y 3 periódicos:

$$\begin{aligned} T_{Ln} &= P_{n1} \\ T_{Ln}' &= P_{n2} + \text{si}(T_{L \text{ de } Pn2} \text{ en etapa anterior} > T_{Ln})\{T_{L \text{ de } Pn2} - T_{Ln}\} + T_{Ln} \\ T_{Ln}'' &= P_{n3} + \text{si}(T_{L \text{ de } Pn3} \text{ en etapa anterior} > T_{Ln}')\{T_{L \text{ de } Pn3} - T_{Ln}'\} + T_{Ln}' \end{aligned}$$

4. Selección y cruce

En este apartado definiremos los criterios para seleccionar a los padres y para obtener los hijos mediante el cruce de los padres.

Para la selección, con los valores de fitness obtenidos podemos obtener un ranking de individuos mediante la fórmula:

$$\text{rank}(i) = \frac{\text{fitness}(i)}{\sum_i \text{fitness}(i)}$$

donde i es cada individuo e I es el conjunto de los individuos. Mediante esta fórmula ordenamos los individuos en orden ascendente del valor rank siendo los individuos con valor más bajo los “mejores”. En una población de n individuos por ejemplo, elegiremos a los 2 con valor rank más bajo para ser los padres.

Para el cruce, realizaremos la combinación de las listas de los padres, eligiendo 2 listas del padre con menor valor rank y 1 del otro, por ejemplo L_1 y L_3 del padre_1 y L_2 del padre_2 hasta realizar todas las combinaciones posibles obteniendo 3 hijos en nuestro ejemplo de problema siendo individuos con 3 listas.

5. Mutación

La mutación en mi algoritmo genético consistirá en un paso muy simple, con la probabilidad de mutación 0.25, el intercambio de posición de 2 elementos dentro de una de las listas.

6. Reemplazo

Por otra parte, el reemplazo en este algoritmo consistirá en sustituir los n “peores” individuos por los n hijos generados. En caso de generar 3 hijos, pues los 3 individuos con menor valor de fitness serán reemplazados.

7. Parada

La condición de parada que he elegido para el algoritmo será la de convergencia. Tras 5 iteraciones sin obtener ninguna mejora el algoritmo se detendrá ofreciéndonos el individuo con mejor valor de fitness (menor tiempo) como solución al problema.

8. Traza

Sea la población inicial generada aleatoriamente dónde la codificación consiste en cada lista es un lector y cada elemento de la lista el periódico que lee en cada unidad de tiempo (1º, 2º o 3º) y su decodificación consistirá en la construcción del diagrama de gantt correspondiente con este orden de lectura y la tabla de tiempos de lectura según el lector y periódico:

$$i_1 = ([3, 1, 2], [2, 1, 3], [2, 3, 1])$$

$$i_2 = ([1, 3, 2], [3, 2, 1], [2, 1, 3])$$

$$i_3 = ([1, 2, 3], [1, 3, 2], [2, 1, 3])$$

$$i_4 = ([2, 3, 1], [2, 1, 3], [3, 1, 2])$$

$$i_5 = ([3, 2, 1], [3, 1, 2], [1, 2, 3])$$

Calculamos los valores de fitness para todos los individuos siguiendo el método descrito anteriormente:

$$fitness(i_1) = \max(65, 75, 80) = 80$$

$$fitness(i_2) = \max(65, 68, 60) = 68$$

$$fitness(i_3) = \max(70, 65, 80) = 80$$

$$fitness(i_4) = \max(63, 68, 85) = 85$$

$$fitness(i_5) = \max(68, 90, 80) = 90$$

Con los valores de fitness podemos elaborar el ranking quedando tal que:

$$i_2 = 0.168$$

$$i_1 = 0.198$$

$$i_3 = 0.198$$

$$i_4 = 0.210$$

$$i_5 = 0.223$$

Por lo tanto seleccionaremos a los individuos 1 y 2 para ser los padres y realizar las combinaciones pertinentes resultando los hijos 6, 7 y 8 mediante el cruce.

$$i_6 = ([1, 3, 2], [3, 2, 1], [2, 3, 1])$$

$$i_7 = ([1, 3, 2], [2, 1, 3], [2, 1, 3])$$

$$i_8 = ([3, 1, 2], [3, 2, 1], [2, 1, 3])$$

Finalmente, se aplica la mutación sobre los supervivientes de la última generación siguiendo las probabilidades, en este caso 0.25:

$$i_1 = ([3, 2, 1], [2, 1, 3], [2, 3, 1])$$

$$i_2 = ([1, 3, 2], [3, 2, 1], [3, 1, 2])$$

Resultando por lo tanto en la población final para la siguiente iteración:

$$i_1 = ([3, 2, 1], [2, 1, 3], [2, 3, 1])$$

$$i_2 = ([1, 3, 2], [3, 2, 1], [3, 1, 2])$$

$$i_6 = ([1, 3, 2], [3, 2, 1], [2, 3, 1])$$

$$i_7 = ([1, 3, 2], [2, 1, 3], [2, 1, 3])$$

$$i_8 = ([3, 1, 2], [3, 2, 1], [2, 1, 3])$$