

Metaheurísticas

Práctica 3



APNEA DEL SUEÑO

Damián Martínez Ávila
José Ángel Expósito Fernández
Gabriel Cordero Contreras

29 de Abril de 2022

[Enlace al código](#)

Índice

1. Representación de los Datos	2
2. Evaluación de las Soluciones	3
3. Población Inicial	4
4. Algoritmo Genético	4
5. Resultados	5
5.1. Longitud de secuencia de 7 a 9	5
5.2. Longitud de secuencia de 10 a 14	5
5.3. Longitud de secuencia mayor o igual a 15	6

1. Representación de los Datos

Los datos de cada paciente se almacenan en un diccionario donde las keys son los distintos eventos y los values una lista con los instantes donde ocurre dicho evento.

Preprocesamiento de los datos de cada uno de los pacientes:

```
A 1:B 1:B 11:A 46:C 53  replace(":", " ")
A 1 B 1 B 11 A 46 C 53  split(" ")
[A, 1, B, 1, B, 11, A, 46, C, 53]
```

Vamos recorriendo la lista:

- Si encontramos una letra, nos indica en que Key del diccionario se guarda el próximo instante
- Si encontramos un número, lo almacenamos donde nos haya marcado la letra anterior

```
{ 'A' : [ 1, 46],
  'B' : [ 1, 11],
  'C' : [53]
}
```

En el código hay una pequeña variación para que los instantes vayan cambiando de uno en uno, en vez de con saltos como en el dataset original.

```
{ 'A' : [ 1, 3],
  'B' : [ 1, 2],
  'C' : [ 4]
}
```

2. Evaluación de las Soluciones

Para evaluar la solución usamos dos criterios:

- Frecuencia de la secuencia entre los pacientes:

Para cada uno de los pacientes comprobamos si cada evento de la secuencia les ocurre siguiendo el orden de esta.

Si el evento no está en el diccionario del paciente significa que no cumple la secuencia, por lo que pasamos a comprobar el siguiente paciente.

En caso contrario, debemos comprobar el instante en el que nos encontramos y los que nos muestra el diccionario. Si encontramos en el diccionario uno mayor o igual, la secuencia por el momento es buena y continuamos con la siguiente posición de la solución, actualizando el instante actual con el encontrado en el diccionario. Si no hay ninguno que cumpla la condición, significa que el paciente no cumple la secuencia y comprobamos el siguiente paciente. Cuando se compruebe la última posición y la cumpla, se sumará uno a la frecuencia devuelta.

Un detalle a tener en cuenta es cuando en la secuencia hay dos o más eventos iguales seguidos, hay que tener en cuenta que solo sirve un instante mayor al actual a partir del segundo evento igual porque sino se cogería el instante que se cogió antes siempre. Y el igual es útil cuando en el mismo instante ocurren dos eventos distintos.

- Número distinto de eventos:

Se calcula el número de eventos distintos en una secuencia. En el código se calcula con la función `entropy`.

ABBCAB

Número de eventos distintos: 3

3. Población Inicial

Para que la población inicial tenga al menos individuos donde la frecuencia de la secuencia sea al menos 1, extraemos una secuencia de un paciente aleatorio. Para obtener la secuencia, entre todos los eventos de un paciente seleccionamos aleatoriamente n distintos de ellos, donde n es la longitud de la secuencia deseada.

Eventos Paciente Original: ABCCACBACAB

Secuencia Obtenida: BAAC

En nuestro código obtenemos 10 individuos de esta forma y el resto hasta llegar al tamaño de población de manera aleatoria.

4. Algoritmo Genético

Hemos optado por usar un algoritmo genético elitista en el cual en cada generación se mantienen los 10 mejores individuos y el resto hasta llegar al tamaño de la población son los descendientes de la anterior generación.

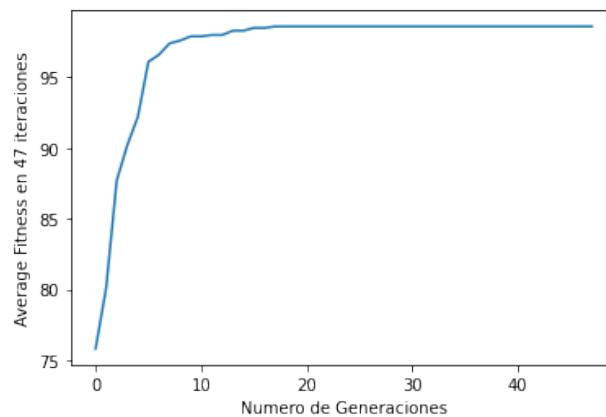
En cuanto al operador de cruce hemos usado Single Point Crossover y con el de mutación Bit Mutation. Para seleccionar los padres se usa un torneo entre 5 individuos aleatorios. El algoritmo se detiene una vez que llegue a las generaciones máximas permitidas o cuando lleve 30 generaciones sin mejorar.

5. Resultados

5.1. Longitud de secuencia de 7 a 9

10 Mejores Secuencias: [('CCCCCFC', 99, 2), ('FCCCCCC', 99, 2), ('CCCCCCL', 99, 2), ('ACCCCC', 99, 2), ('CFCCCC', 99, 2), ('CCCCCCC', 99, 1), ('ACCCCL', 98, 3), ('CCCCCFL', 98, 3), ('CFCCCL', 98, 3), ('ACCCCF', 98, 3)]

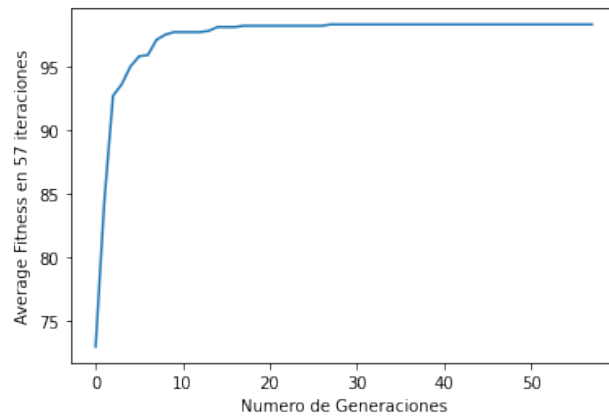
Frecuencia Maxima : 99 Frecuencia Media : 98.6 Frecuencia Minima : 98
Tiempo Empleado: 2.37 Media de eventos: 2.3



5.2. Longitud de secuencia de 10 a 14

10 Mejores Secuencias: [('FCCCCCCCC', 99, 2), ('CFCCCCCCCC', 99, 2), ('CCCCCCCC', 99, 1), ('ACFCCCCCCCC', 98, 3), ('AFCCCCCCCC', 98, 3), ('FCCCCCCCCCL', 98, 3), ('CFCCCCCCCCCL', 98, 3), ('CCCCCCCCCL', 98, 2), ('CCCCCFCCCC', 98, 2), ('ACCCCCCCCC', 98, 2)]

Frecuencia Maxima : 99 Frecuencia Media : 98.3 Frecuencia Minima : 98
Tiempo Empleado: 5.22 Media de eventos: 2.3



5.3. Longitud de secuencia mayor o igual a 15

10 Mejores Secuencias: [('CCCCFCCCCCCCCCCC', 98, 2), ('CCCFCCCCCCCCCCCC', 98, 2), ('CCCCCFCCCCCCCC', 98, 2), ('CCFCCCCCCCCCCCC', 98, 2), ('CCCCCFCCCCCCCC', 98, 2), ('CCCCCCCCCCCCCCCC', 98, 1), ('CCCFCCCCCCCCCCCCCL', 97, 3), ('ACCCCFCCCCCCCC', 97, 3), ('CCCCFCCCCCCCCCCCCCL', 97, 3), ('ACCCCFCCCCCCCC', 97, 3)]

Frecuencia Maxima : 98 Frecuencia Media : 97.6 Frecuencia Minima : 97
 Tiempo Empleado: 13.79124412536621 Media de eventos: 2.3

