Programación Evolutiva:

Práctica 1.

Pablo Mac-Veigh

Jorge Sánchez

<https://github.com/Pabsilon/PracticasPE>

Hemos realizado en Java un programa que resuelve el problema del viajante de comercio. Está representado mediante ciudades de España y busca el recorrido (empezando y acabando en Madrid) cuya distancia recorrida sea mínima.

**Valores por defecto (no significa que sea la mejor combinación):**

* **Población:** 100
* **Generaciones:** 300
* **Método selección:** Ruleta
* **Probabilidad cruce:** 60 %
* **Método cruce:** PMX
* **Probabilidad mutación:** 5 %
* **Método mutación:** Inserción
* **Elitismo:** Desactivado

**Modo de uso:**

Idéntico al de la práctica anterior. Pero, ahora consta de tres tabs adicionales donde se encuentran dos gráficos de la mejor solución encontrada (gráfico ordenado simulando el mapa de España y otro sin ordenar, cada uno en su respectivo tab) y una gráfica con los mejores valores obtenidos al variar los valores de probabilidad de cruce, mutación u número de generaciones (parte opcional).

**Cromosoma:**

Los cromosomas se codifican con un array de genes de enteros, donde cada entero es la id de una ciudad y todos los cromosomas empiezan por la ciudad Madrid, los demás valores se generan aleatoriamente. El fenotipo es la conversión del array de entero (genes) a su respectivo nombre de ciudad.

**Aptitud:**

La aptitud se calcula sumando la distancia de la ciudad 0 a la 1, 1 a la 2… hasta n. Más la distancia de la última ciudad a la ciudad inicial (de n a 0). Siendo la ciudad inicial siempre Madrid.

**Algoritmos de selección, cruce y mutación**

Todos los algoritmos han sido sacados de los apuntes o del libro “*Algoritmos Evolutivos. Un enfoque práctico; Lourdes Araujo. Carlos Cervigón”* a excepción de los dos algoritmos propios.

Método propio de cruce:

Se trata de ir eligiendo una ciudad de cada padre (50 % de probabilidad de coger del padre 1 o del 2) e ir añadiendo esa ciudad a un array. EL hijo 1 se formará cogiendo las ciudades por orden de ese array (si ya no están es el hijo). Las ciudades restantes se introducirán en el hijo 2.

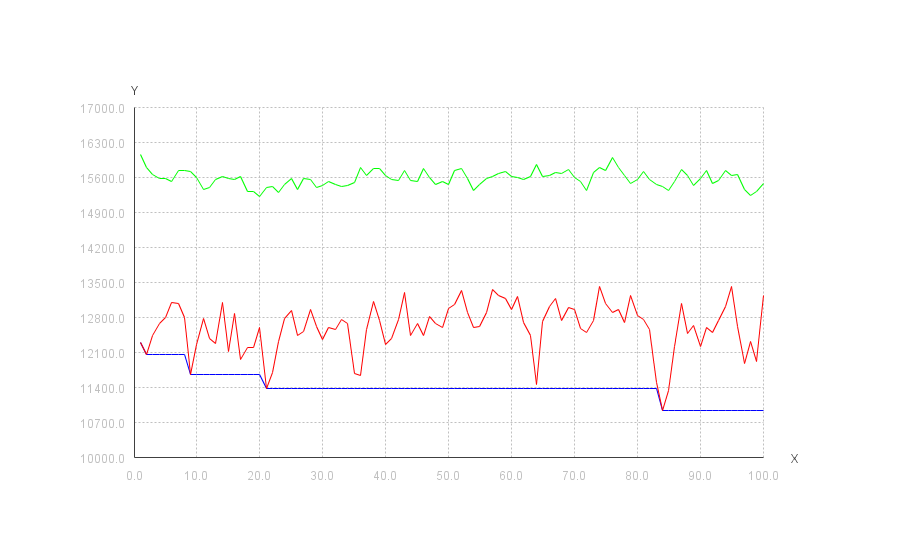
Método propio de mutación:

Se trata de coger un intervalo de genes (acotado por dos puntos aleatorios) del array de genes. Dicho intervalo es invertido (es decir, si se tiene 1-2-3 se cambia a 3-2-1) y es intercambiado por otro intervalo del mismo tamaño y escogido aleatoriamente del array de genes.

**Resultados:**

En todos los ejemplos siguientes se mantendrán los valores por defecto (listados anteriormente). Todo aquel parámetro que haya sido cambiado será notificado previamente.

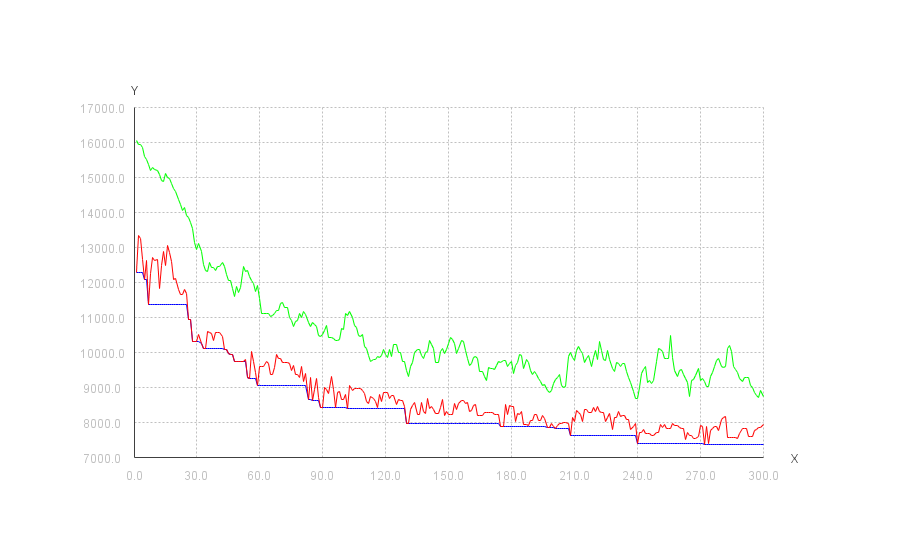
Valores por defecto



Mejor individuo: C:\hlocal\PE\Fotos\Sin título.png

Se puede ver que esta combinación es muy mala. La media de cada generación (línea verde) es más o menos uniforme, sin embargo el mejor valor de cada generación (línea roja) tiene muchos picos, aunque suele mantenerse en valores entre 13000 y 1200. El mejor valor (línea azul) se actualiza pocas veces.

Método de cruce propio y método de mutación propio



Mejor individuo:

C:\hlocal\PE\Fotos\Sin título.png

La combinación de los métodos propios produce una gran mejoría respecto a los valores iniciales. Se aprecia una mejoría progresiva en todos los valores (media generación (verde), mejor generación (roja), mejor valor absoluto (azul)). Así como pocos picos de cambio a peores aptitudes o picos de leve cambio entre generaciones. La media de mejores aptitudes por ejecución de esta combinación es de 8500. Siendo 7300 el mejor valor obtenido es unas 5 ejecuciones.

Método de cruce propio y método de mutación propio, elitismo y torneo con 2 individuos

Los resultados y gráficas son iguales a la combinación anterior solo que la media de mejores aptitudes por generación son de 6500, haciendo esta combinación más estable que la anterior. Se han obtenido valores de 5800 pero puntualmente.