

Práctica 3

# Algoritmos heurísticos no constructivos: Genéticos

Modelos bioinspirados y heurísticas de búsqueda



Victor Manuel Rodriguez Navarro



## Contenido

Objetivo.....	2
Algoritmo Genético Básico.....	2
Estudio de parámetros.....	2
Tamaño del operador de mutación .....	2
Tamaño de la población.....	3
Tamaño del torneo de selección.....	3
Resultados.....	4
Algoritmo Genético CHC .....	4
Tamaño de la población.....	4
Tamaño del operador de mutación diverge .....	5
Número de reinicializaciones.....	6
Resultados.....	7
Algoritmo Genético Multimodal .....	7
Resultados.....	8
Comparativa global .....	8

## Objetivo

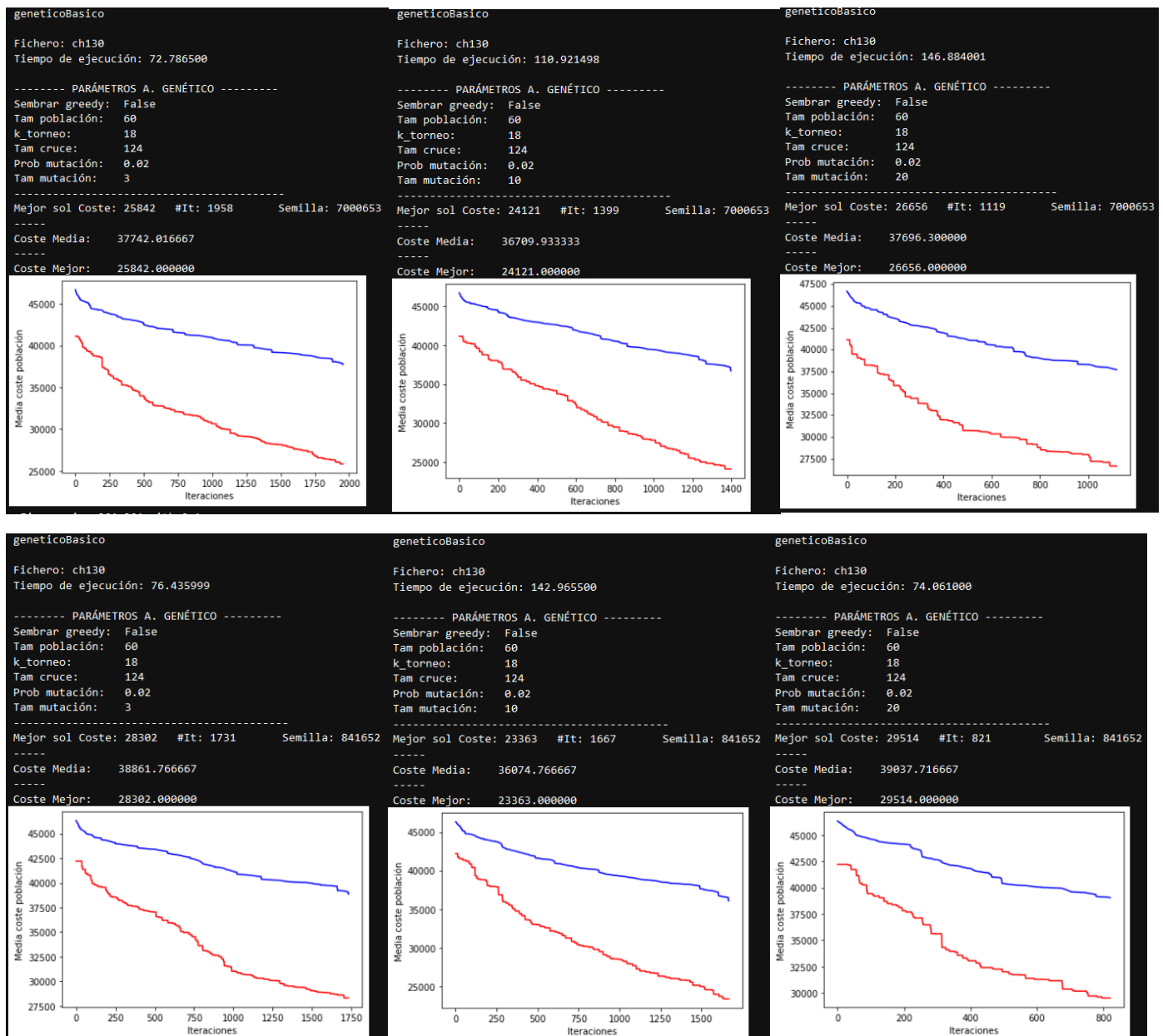
El objetivo de esta práctica es estudiar el funcionamiento de los algoritmos heurísticos no constructivos. Este tipo de algoritmos parten de una solución inicial o un conjunto de ellas para a través de operaciones de transformación ir mejorando las soluciones candidatas. Los algoritmos a estudiar son los evolutivos: genético simple, multimodal y CHC. Para ello, se requerirá que el alumno implemente estos algoritmos, para resolver el *Problema del Viajante de Comercio* (TSP). El comportamiento de los algoritmos implementados deberá compararse con un *Algoritmo Greedy*.

## Algoritmo Genético Básico

### Estudio de parámetros

#### Tamaño del operador de mutación

Vamos a realizar 2 pruebas de distinta semilla con 2%, 8% y 15% del tamaño del individuo, respectivamente.



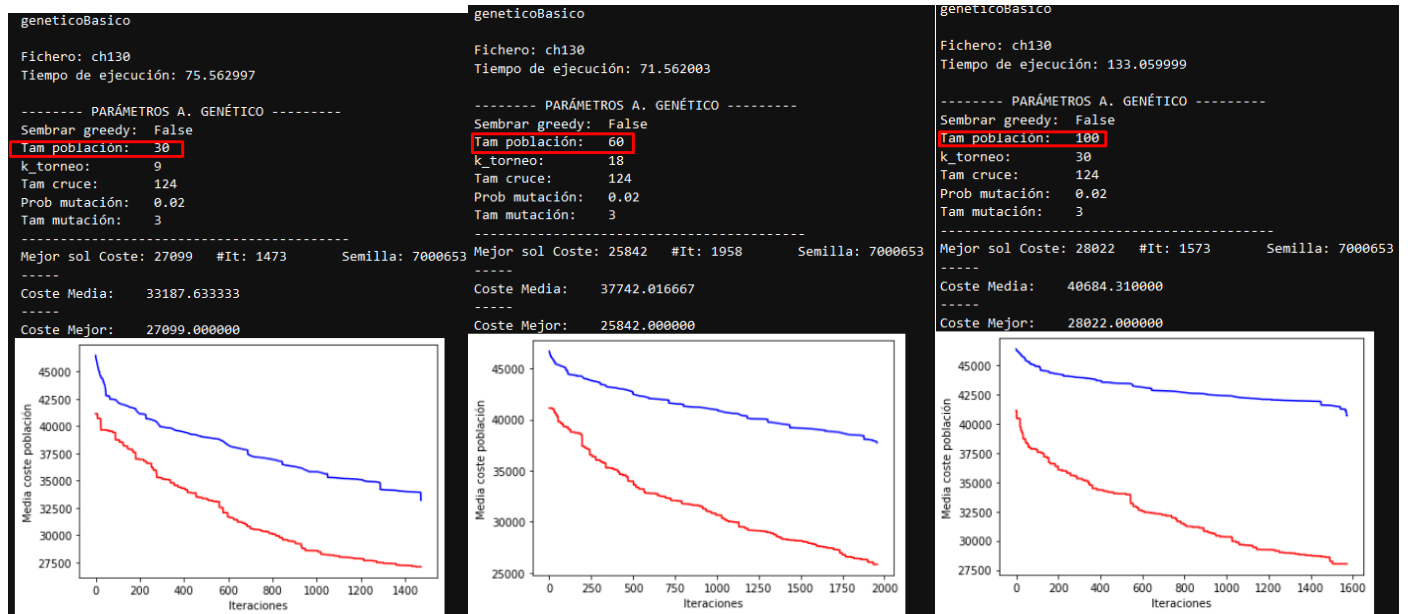
El parámetro del tamaño de la mutación  $k$ , controla la longitud de la sublista de nodos a desordenar (operador de mutación).

El problema del TSP es muy sensible a pequeños cambios en el orden de los nodos del camino, por lo que no es recomendable un valor muy alto, aunque una solución mutada debe sufrir un cambio significativo para aumentar la capacidad de exploración. Es por eso, que nos vamos a quedar con un valor intermedio de 8%. Igualmente, la

probabilidad de que se aplique este operador de mutación sobre un hijo debe ser baja, por lo que vamos a elegir un 2%.

## Tamaño de la población

Vamos a probar con diferentes configuraciones en el tamaño de la población. Recordemos que debe estar en torno a 30-100 individuos.

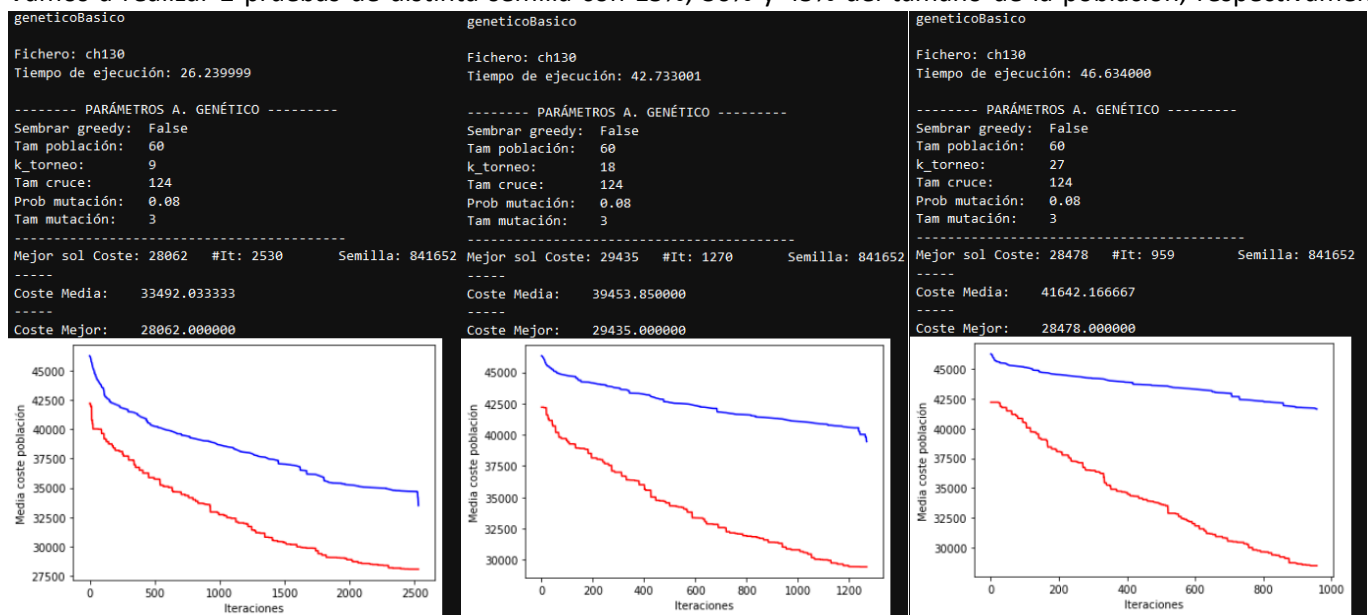


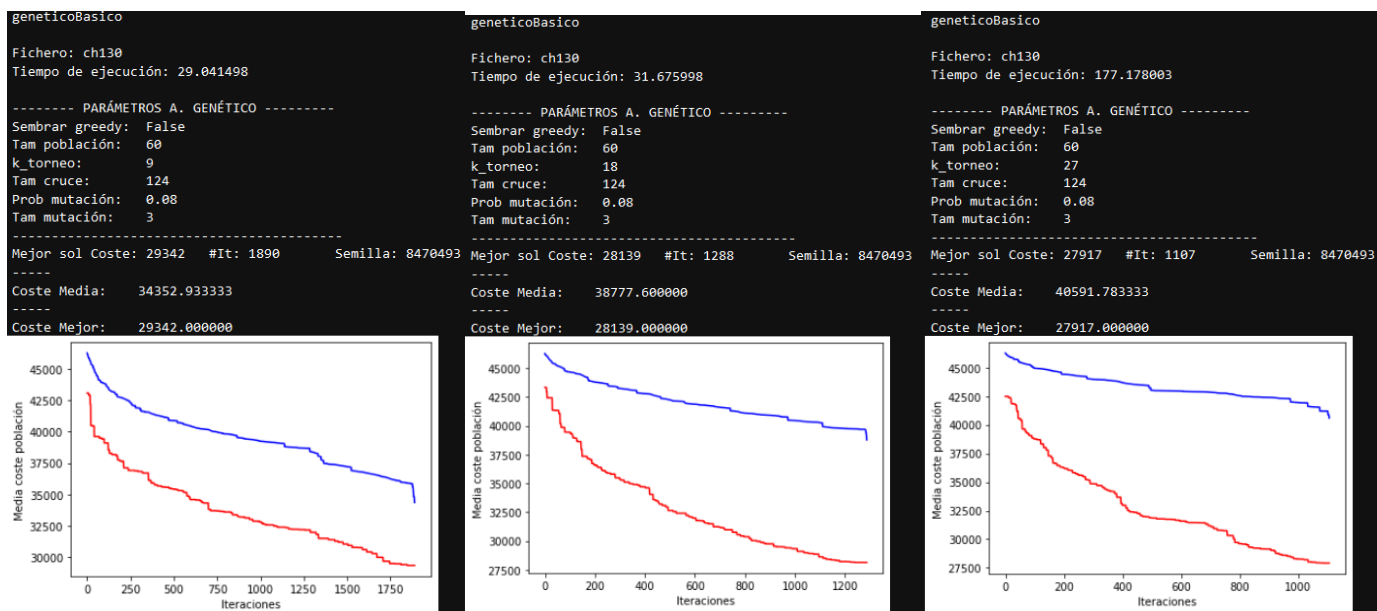
Como vemos, a mayor tamaño, mayor diversidad se presenta en la población. Aunque debemos encontrar un punto medio para poder alcanzar una aproximación a la convergencia con el mejor individuo. Por lo que nos quedaremos con 60 individuos.

## Tamaño del torneo de selección

El parámetro k del tamaño del torneo de selección indica cuántos individuos aleatorios se examinan para ser ordenados por distancia de Hamming al hijo y se devuelve el de menor distancia (más parecido) al hijo, para que este lo reemplace.

Vamos a realizar 2 pruebas de distinta semilla con 15%, 30% y 45% del tamaño de la población, respectivamente.





Como podemos ver, a menor cantidad de individuos examinados, la población tiende a converger más con el mejor individuo. Esto se debe a que al tener menor capacidad de exploración para reemplazar al que más se le parezca, siguen quedando individuos similares que no han sido examinados, y este efecto, a la larga, hace que la población sea muy parecida (poca diversidad).

Por ello, es necesario mantener un equilibrio entre diversidad y convergencia con el mejor individuo, por lo que vamos a elegir el valor intermedio de 30%.

## Resultados

Estos son los resultados tras 5 ejecuciones del algoritmo con distintas semillas aleatorias.

Algoritmo Genético Básico						
	st70		ch130		a280	
	Coste	#EV	Coste	#EV	Coste	#EV
Ejecución1	2259	273	34469	228	26162	308
Ejecución2	2761	169	33770	225	26522	359
Ejecución3	2573	238	34423	258	26320	302
Ejecución4	2569	207	32114	263	26661	242
Ejecución5	2207	252	33239	249	27426	209
Mejor	2 207,00	169,00	32 114,00	225,00	26 162,00	209,00
Media	2 473,80	227,80	33 603,00	244,60	26 618,20	284,00
Des. Tip. (s)	233,83	40,69	974,66	17,30	490,07	58,98

## Algoritmo Genético CHC

### Tamaño de la población

En este algoritmo podemos determinar igualmente el tamaño de la población. Como se basa en reinicializar la población cada vez que converge (al no insertar ningún nuevo descendiente), un número mayor de individuos será más apropiado, para tener más posibilidades de cruzar individuos y que produzcan buenos descendientes. Por eso, vamos a incrementar la población a 75 individuos.

## Tamaño del operador de mutación diverge

Por otro lado, el tamaño de la mutación que se aplica en la función `diverge` (para controlar cuánto se copia y cuánto se muta al coger como plantilla el mejor individuo de la población para crear una nueva población en base a él). Vamos a experimentar con un 5%, 10%, 25% y 35% de mutación a aplicar sobre la plantilla, respectivamente

Tiempo de ejecución: 3.375288

----- PARÁMETROS A. GENÉTICO -----

Sembrar greedy: False

Tam población: 75

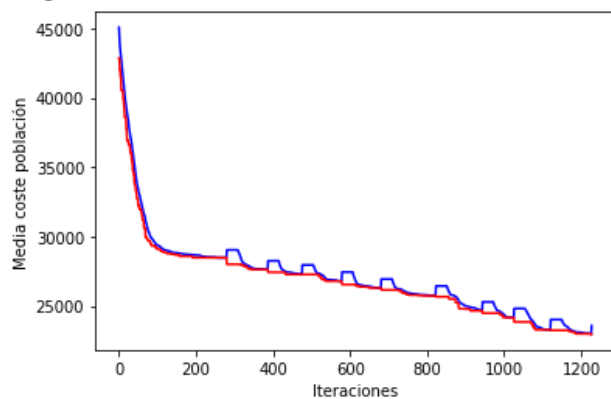
Tam mutación diverge: 6

-----  
Mejor sol Coste: 22951 #It: 1228 Semilla: 1077776

-----  
Coste Media: 23580.440000

-----  
Coste Mejor: 22951.000000

<Figure size 864x864 with 0 Axes>



----- PARÁMETROS A. GENÉTICO -----

Sembrar greedy: False

Tam población: 75

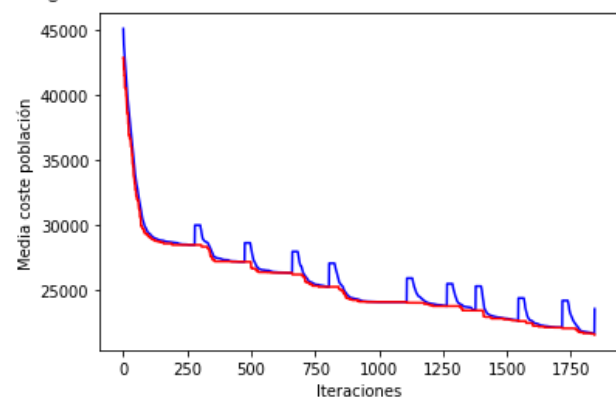
Tam mutación diverge: 13

-----  
Mejor sol Coste: 21594 #It: 1844 Semilla: 1077776

-----  
Coste Media: 23575.520000

-----  
Coste Mejor: 21594.000000

<Figure size 864x864 with 0 Axes>



----- PARÁMETROS A. GENÉTICO -----

Sembrar greedy: False

Tam población: 75

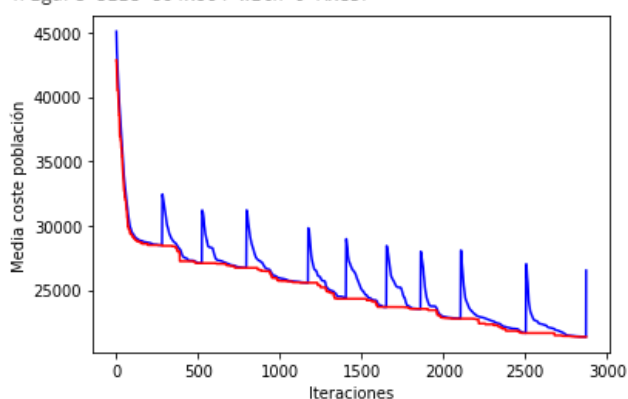
Tam mutación diverge: 32

-----  
Mejor sol Coste: 21353 #It: 2873 Semilla: 1077776

-----  
Coste Media: 26517.186667

-----  
Coste Mejor: 21353.000000

<Figure size 864x864 with 0 Axes>



----- PARÁMETROS A. GENÉTICO -----

Sembrar greedy: False

Tam población: 75

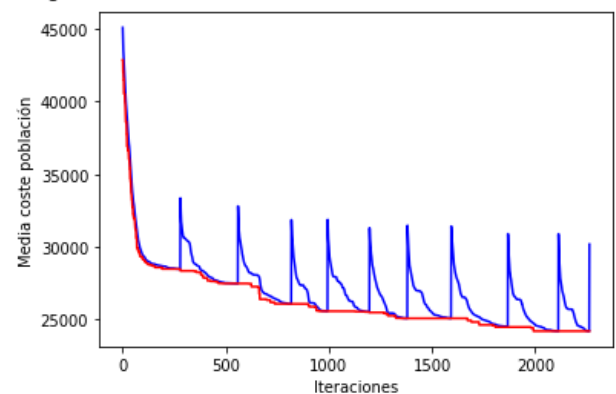
Tam mutación diverge: 46

-----  
Mejor sol Coste: 24205 #It: 2264 Semilla: 1077776

-----  
Coste Media: 30181.186667

-----  
Coste Mejor: 24205.000000

<Figure size 864x864 with 0 Axes>

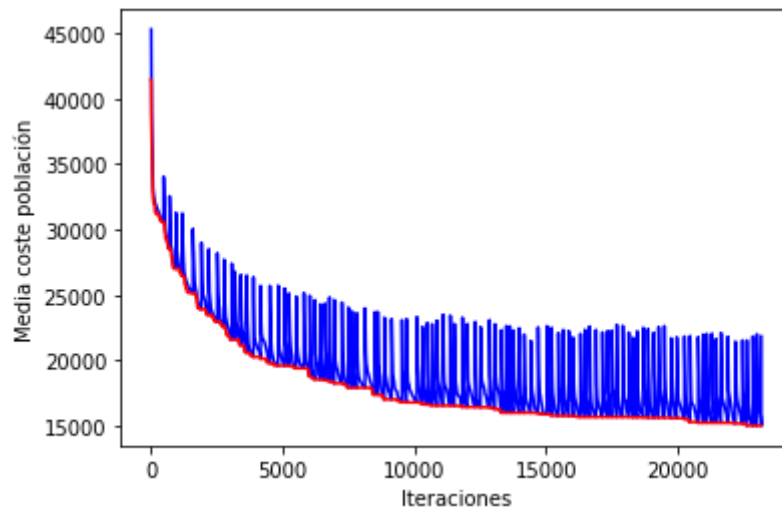


Como vemos, cuanto menos se conserva del individuo plantilla, más acentuados son los picos del coste medio de la población al reinicializar. Aunque dé más diversidad, no se nota mejoría al incrementar el valor de este parámetro de mutación, por lo que nos vamos a quedar con un 25% a mutar, que consigue buenos resultados aportando diversidad, sin consumir muchas iteraciones al alterar la población.

## Número de reinicializaciones

Los resultados anteriores se obtuvieron con 10 reinicializaciones. No obstante, los algoritmos genéticos funcionan mejor si procesan durante más tiempo. Es por eso por lo que vamos a subir las reinicializaciones a 100.

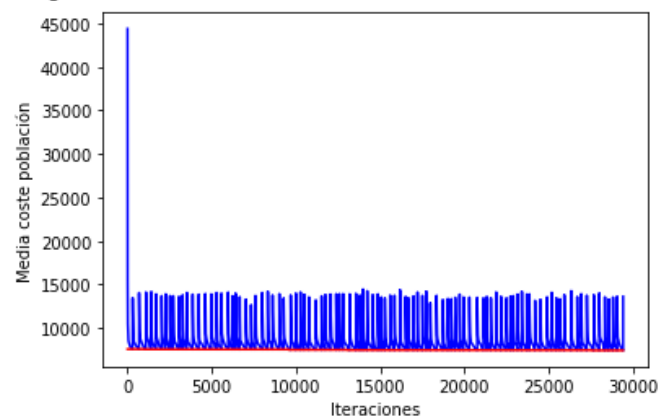
```
----- PARÁMETROS A. GENÉTICO -----  
Sembrar greedy: False  
Tam población: 75  
Tam mutación diverge: 32  
-----  
Mejor sol Coste: 15035 #It: 23201 Semilla: 6481120  
-----  
Coste Media: 21845.560000  
-----  
Coste Mejor: 15035.000000  
<Figure size 864x864 with 0 Axes>
```



De esta forma, conseguimos bajar notablemente el coste medio y de la mejor solución obtenida al final del algoritmo (24668 vs 15035 unidades de coste), igualmente superando así al algoritmo genético básico.

Por último, si probamos a sembrar una solución del algoritmo greedy, veremos que tras la ejecución del algoritmo genético CHC, conseguimos una solución que supera al propio algoritmo greedy.

```
----- PARÁMETROS A. GENÉTICO -----  
Sembrar greedy: True  
Tam población: 75  
Tam mutación diverge: 32  
-----  
Mejor sol Coste: 7434 #It: 29436 Semilla: 6481120  
-----  
Coste Media: 13618.866667  
-----  
Coste Mejor: 7434.000000  
<Figure size 864x864 with 0 Axes>
```



## Resultados

Estos son los resultados tras 5 ejecuciones del algoritmo con distinta semilla aleatoria.

Algoritmo Genético CHC						
	st70		ch130		a280	
	Coste	#EV	Coste	#EV	Coste	#EV
Ejecución1	1147	8347	17357	11592	12597	20267
Ejecución2	1360	7724	17056	11053	12310	21101
Ejecución3	1155	8394	17258	11875	12832	20789
Ejecución4	1170	8332	16030	11809	12353	20371
Ejecución5	1327	8120	16183	11579	12372	20748
Mejor	1 147,00	7 724,00	16 030,00	11 053,00	12 310,00	20 267,00
Media	1 231,80	8 183,40	16 776,80	11 581,60	12 492,80	20 655,20
Des. Tip. (s)	102,96	277,61	623,79	323,00	219,92	337,91

Algoritmo Genético CHC Greedy						
	st70		ch130		a280	
	Coste	#EV	Coste	#EV	Coste	#EV
Ejecución1	781	8119	7431	13766	3115	20395
Ejecución2	792	8230	7504	14454	3131	20459
Ejecución3	761	8950	7433	14111	3137	19615
Ejecución4	792	8277	7451	14520	3132	20177
Ejecución5	790	8519	7502	13971	3132	19683
Mejor	761,00	8 119,00	7 431,00	13 766,00	3 115,00	19 615,00
Media	783,20	8 419,00	7 464,20	14 164,40	3 129,40	20 065,80
Des. Tip. (s)	13,22	330,85	36,27	319,88	8,38	395,32

En esta última tabla, podemos ver que, si se siembra la solución greedy como un individuo de la población inicial, se consiguen resultados muy buenos, que llegan a superar al propio greedy.

## Algoritmo Genético Multimodal

Por último, para el algoritmo genético multimodal, al haber usado el método estacionario para el algoritmo genético básico, hemos optado por implementar el método secuencial de 5 nichos, en lugar de clearing.

```
geneticoMultiModal

Fichero: ch130
0 []
1 [22889]
2 [22889, 21182]
3 [22889, 21182, 21431]
4 [22889, 21182, 21431, 22025]
Tiempo de ejecución: 2243.710419

----- PARÁMETROS A. GENÉTICO -----
Sembrar greedy: False
Tam población: 60
k_torneo: 18
Tam cruce: 124
Prob mutación: 0.02
Tam mutación: 10
Radio nicho: 13.0
Tamaño nicho (Kappa): 5
-----
----- COSTES SOLUCIONES -----
22889
21182
21431
22025
21408
Mejor sol Coste: 21182 Semilla: 1043253
```



Como vemos, en esta ejecución nos muestra para cada una de las cinco llamadas secuenciales al algoritmo básico, el vector de soluciones (mostrando los individuos como su coste) que se le pasa para poder devaluar la función de fitness.

## Resultados

Estos son los resultados tras 5 ejecuciones del algoritmo con distinta semilla aleatoria. Se ha representado la mejor solución de las 5 del nicho.

Algoritmo Genético Multimodal Secuencial			
	st70	ch130	a280
	Coste	Coste	Coste
Ejecución1	1771	21263	20278
Ejecución2	1736	23379	20298
Ejecución3	1601	23596	21432
Ejecución4	1507	21419	19474
Ejecución5	1680	21363	19933
Mejor	1 507,00	21 263,00	19 474,00
Media	1 659,00	22 204,00	20 283,00
Des. Tip. (s)	106,52	1 175,51	724,13

## Comparativa global

En la siguiente gráfica vemos una comparativa global de todos los algoritmos genéticos implementados con el Greedy y Búsqueda Aleatoria del mejor vecino anteriores.

Modelo	st70 (675)			ch130 (6110)			a280 (2579)		
	Med	Mej	S	Med2	Mej2	S2	Med3	Mej3	S3
Greedy	830,00	830,00	0	7 579,00	7 579,00	0	3 157,00	3 157,00	0
BA Mejor	1 181,50	1 056,00	63,85	23 473,40	22 094,00	951,24	26 418,50	24 998,00	793,49
AGB	1 718,80	1 564,00	117,89	24 340,20	22 451,00	1 715,72	21 044,00	20 589,00	374,85
CHC	1 231,80	1 147,00	102,96	16 776,80	16 030,00	623,79	12 492,80	12 310,00	219,92
CHC Greedy	783,20	761,00	13,22	7 464,20	7 431,00	36,27	3 129,40	3 115,00	8,38
AGMS	1 659,00	1 507,00	106,52	22 204,00	21 263,00	1 175,51	20 283,00	19 474,00	724,13

Como podemos ver, el que mejor resultados da es el algoritmo genético CHC, el cual llega a superar incluso al Greedy si se le siembra una solución Greedy a la población inicial.

El multimodal acaba siendo equivalente al genético básico al haber sido implementado el algoritmo secuencial. Obtiene 5 soluciones diferentes entre sí (de distintos nichos), mediante la llamada secuencial al genético básico. Como puede elegir la mejor entre las 5 soluciones, muestra mejores resultados que una sola solución del genético básico.

Por último, en cuanto a robustez, también destaca el CHC, ya que es el que presenta menor desviación típica en el coste tras sus múltiples ejecuciones, reflejando que es capaz de dar buenos resultados de forma consistente.