

PRÁCTICA 2

MODELOS BIOINSPIRADOS Y HEURÍSTICAS DE BÚSQUEDAS

ALBERTO RODERO PEÑA

ÍNDICE

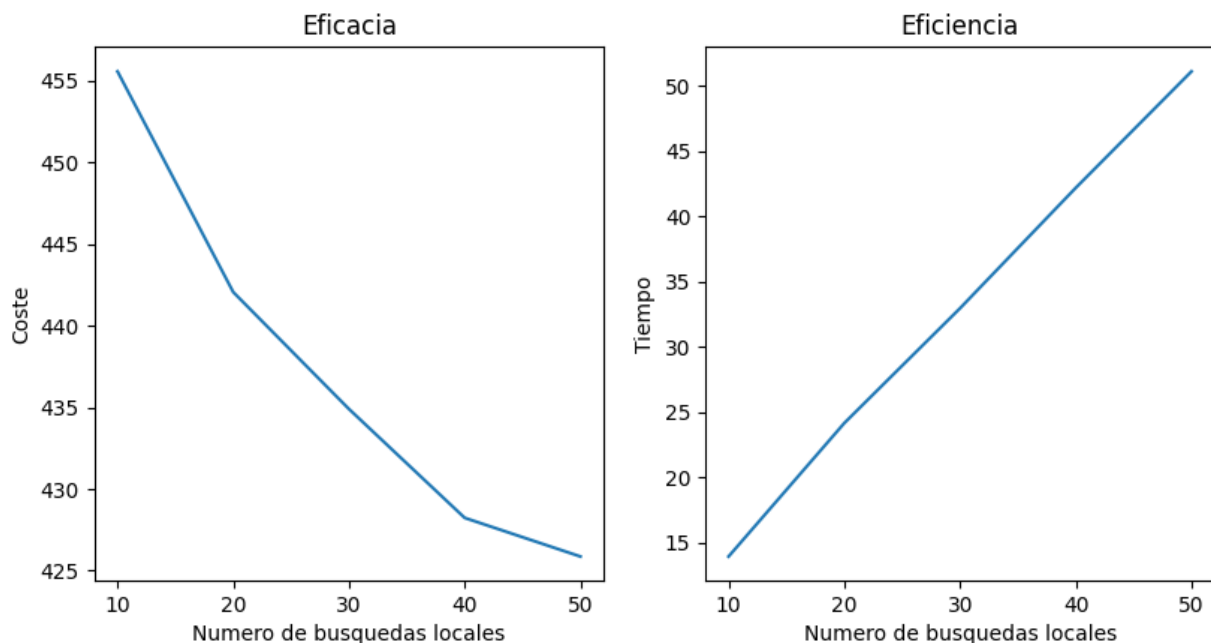
VNS.....	2
Iteraciones máximas	2
Número de intercambio.....	3
Tamaño de cambio.....	3
Ejecución aleatoria.....	4
GENÉTICO BÁSICO.....	6
Número de individuos.....	7
Iteraciones máximas	8
Porcentaje selección	8
Porcentaje reemplazo	9
Porcentaje mutación.....	9
Ejecución aleatoria.....	10
CHC.....	12
Número de reinicios.....	12
Número de individuos.....	12
Número de individuos élite.....	13
Ejecución aleatoria.....	13
MULTIMODAL	15
Número de individuos.....	15
Generaciones	16
Distancia mínima.....	16
Ejecución aleatoria.....	17
COMPARACIÓN	19

VNS

Para el VSN y todos los métodos próximos se ha cambiado la función de evaluación. Ahora existe un peso extra que suma 5 puntos por cada plaza que supere las 205. De forma que los métodos buscarán un equilibrio entre tener menos plazas y tener más puntuación por no poder almacenar tantas bicis, y tener más plazas y tener más puntuación por cada solución que supere las 205 plazas. Mientras antes teníamos un límite inferior de 205 y superior de 220 plazas, ahora solo tendremos un límite inferior. Con esta nueva evaluación el número de plazas final ronda las 230, lo cual no es demasiado con respecto a los límites anteriores, por lo que damos por válido este peso extra de 5. Por lo que la generación de vecinos cambia el número de plazas, pero nunca se elegirá una solución con demasiadas plazas.

La implementación del VNS con el parámetro k es la indicada en la práctica. La condición de parada no vendrá dada por k y su límite, sino por un número de búsquedas locales.

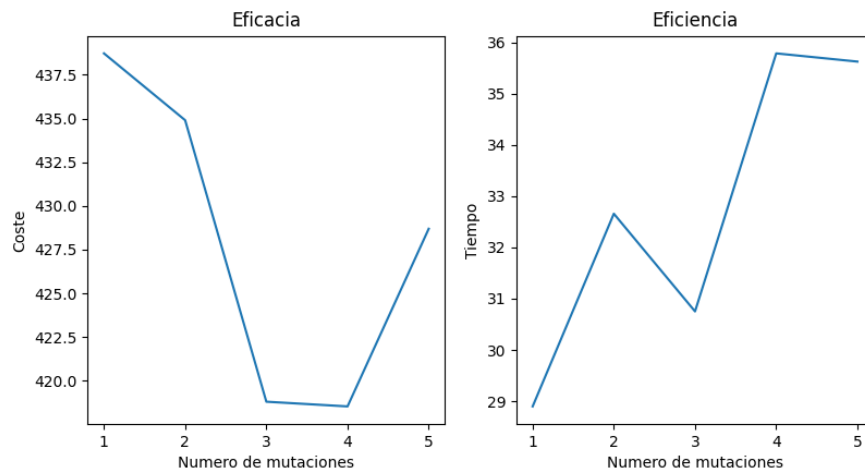
Iteraciones máximas



Usaremos 50 búsqueda locales, ya que podemos permitirnos 1 minuto por semilla. Para una ejecución de 30 minutos haríamos unas 1500 búsquedas locales, aunque en su mayoría no supondrían ninguna mejora. Sería más eficiente hacer 100 búsqueda locales en 15 semillas distintas y quedarnos con el mejor resultado

Número de intercambio

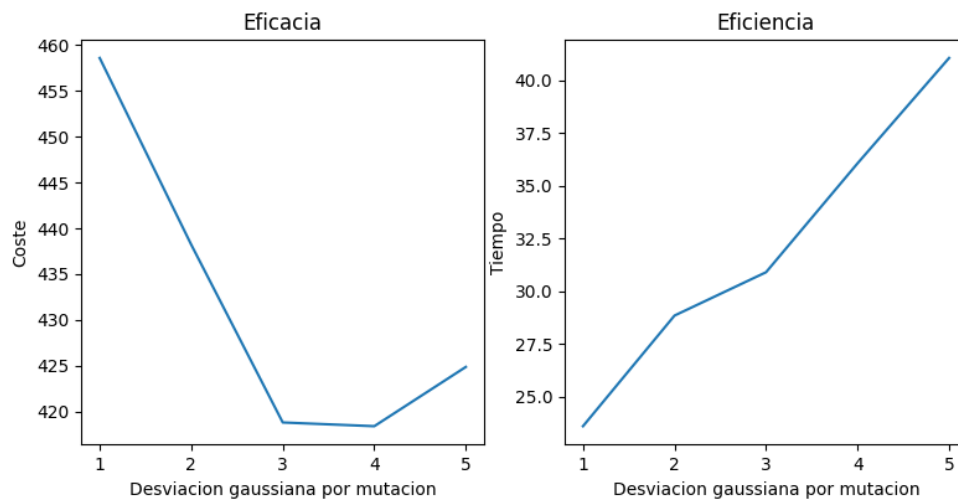
Representa el numero de cambios que se van a hacer en la sublista de tamaño fijo a la hora de variar la solución actual. Este valor esta multiplicado por k , para que sea proporcional a la longitud de la sublista



Como vemos apenas hay diferencia entre 3 y 4 cambios, sin embargo 3 es más eficiente, por lo que nos quedamos con este valor

Tamaño de cambio

Representa la desviación al hacer un cambio en la solución. Si en la estación tenemos 4 plazas sustituimos este valor por una distribución gaussiana de media 4 y desviación n .



De nuevo los mejores valores son 3 y 4 siendo 3 más eficiente.

Ejecución aleatoria

Las x representan las búsquedas locales que mejoran la solución y o las que no.

VNS

Parametros:

Numero de intercambios: 3

Tamaño de cambio: 3

Iteraciones maximas: 50

Semilla: 240

Inicial

Solucion: [5 5 17 10 25 17 5 17 7 12 5 17 22 15 25 10] Coste: 758.4294975967458 Plazas: 214

Ejecutando busquedas locales: xxxxxooxxooxoo

Final

Solucion: [19 12 21 14 19 13 11 11 12 14 18 25 2 13 17 12] Coste: 410.0171522473521 Plazas: 233

Semilla: -965

Inicial

Solucion: [18 3 12 18 7 16 14 14 16 7 12 16 16 18 5 18] Coste: 648.4822017118598 Plazas: 210

Ejecutando busquedas locales: xxxxxoo

Final

Solucion: [18 11 22 18 19 14 11 14 10 14 17 26 2 19 4 14] Coste: 417.0226428554572 Plazas: 233

Semilla: -668

Inicial

Solucion: [6 14 20 14 6 14 10 18 20 14 10 10 18 18 12 10] Coste: 682.9832508209157 Plazas: 214

Ejecutando busquedas locales: xxooxxoxxxxxxooxxoo

Final

Solucion: [18 12 20 14 20 14 10 12 12 14 18 25 2 27 4 12] Coste: 410.225565010202 Plazas: 234

```

-----
Semilla: -141
-----

Inicial
Solucion: [17 10 10 19 17 12 14 14 14 14 17  8  4 17 21  4]      Coste: 737.6248462974434      Plazas: 212

Ejecutando busquedas locales: xxxxxxxxxxoooooooooooooooooooooooooooooooooooooooooooo

Final
Solucion: [19 12 22 23 19 14 11 12 10 15 19 26  2 12  5 14]      Coste: 416.82280793975343      Plazas: 235

```

```

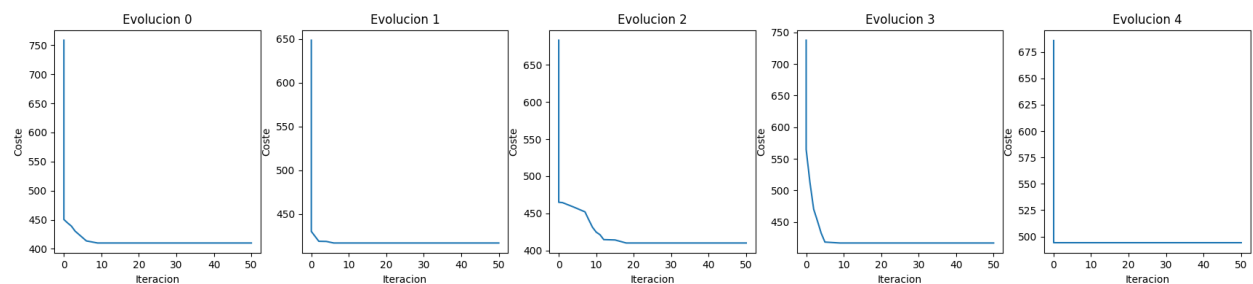
-----
Semilla: -537
-----

Inicial
Solucion: [ 4 14 18 16 23 23  4 16  4 14  4 16 21 11  9 14]      Coste: 686.1648001337597      Plazas: 211

Ejecutando busquedas locales: xxxoooooooooooooooooooooooooooooooooooooooooooooooooooo

Final
Solucion: [10 12 20 14 19 13 12  8 12 14 14 24  3 11 15 12]      Coste: 493.86566781574976      Plazas: 213

```



Podemos ver que la mayor mejora viene de la primera búsqueda local. Sobre todo se puede observar en la última ejecución, que el VNS no presenta ninguna mejora con respecto a la búsqueda local.

GENÉTICO BÁSICO

Se ha implementado un algoritmo genético **estacionario**. Tanto para la **selección** como el **reemplazo** se ha utilizado **torneo**. La **condición de parada** es un número de iteraciones máximas que no se puede superar. Además, estamos mirando las veces que el mejor individuo de la población no mejora, si ocurren un cuarto de las iteraciones máximas sin mejora el algoritmo acaba. Si se produce mejora en el último cuarto de la ejecución las iteraciones máximas aumentan, de forma que siempre debe haber el primer cuarto de ejecuciones sin mejora antes de que acabe.

Para la **mutación** se han probado varias estrategias:

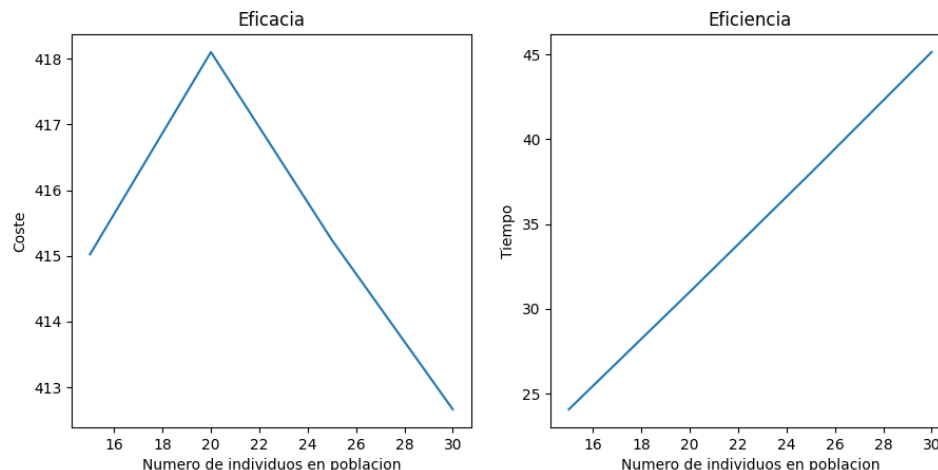
- **Mutación** con intercambio. La mutación ocurre con el traspaso de un número de plazas de una estación a otra. Ya que al hacer un intercambio se modifican 2 genes debemos dividir nuestro porcentaje de mutación entre 2. Los costes de usar la mutación con intercambio son:
[422.53516746 421.49633734 420.63909062 416.35206818 420.69402101]
Media: 420.3433369229225
- **Mutación** individual con genes a mutar fijo. Usando la misma estrategia que en el VSN, por cada gen que se vaya a mutar ejecutamos una distribución gaussiana de media el valor anterior y desviación 3. El número de genes es un número fijo dado por el porcentaje de mutación por la longitud de la solución. Los costes son:
[413.7126029 413.31819133 418.17506189 421.29510224 422.67388967]
Media: 417.8349696090594
- **Mutación** individual con número de genes aleatorio por individuo. Igual que el anterior pero el número de genes a mutar es variable. Por cada gen se lanza un dado entre 0 y 99, si este número es menor o igual que el porcentaje a mutar el gen muta. Este ofrece unos costes muy similares al anterior, pero al ser un número variable de genes algunos individuos saldrán más parecidos a los padres y otros menos. Por lo que puede dar lugar a más diversidad, por eso es algo mejor. Costes:
[416.92223608 413.09447801 418.30883452 417.7195358 421.02230916]
Media: 417.41347871288406
- También se ha probado a forzar que las mutaciones generen soluciones con al menos 205 plazas, pero esto supone un empeoramiento de los resultados. Las soluciones por debajo de 205 tienen un coste de 1000 así que no superarán el torneo del reemplazo. Costes:
[416.92223608 420.77068345 423.59343449 421.29453246 421.02230916]
420.72063912673894

Finalmente usamos la **mutación** individual con número de genes aleatorio por individuo.

Para el **cruce** también hemos probado distintos métodos:

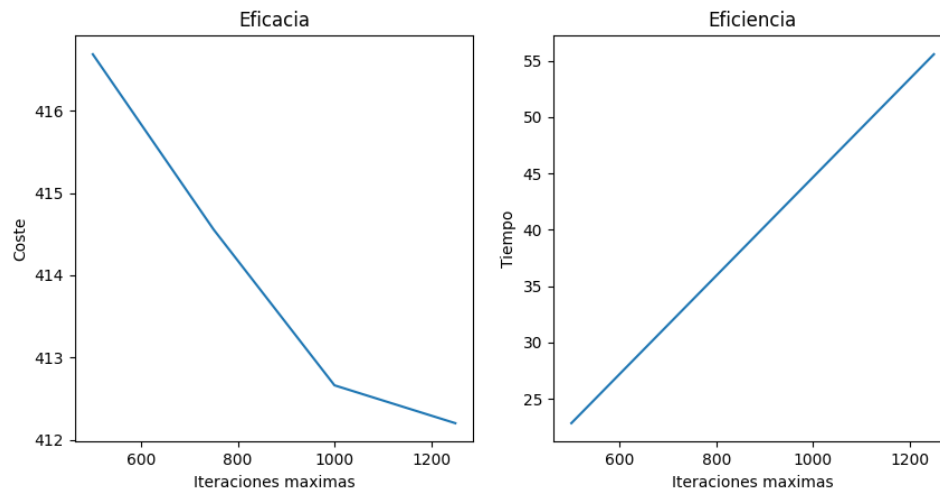
- **Cruce por corte.** Se eligen 2 puntos aleatorios del individuo, los genes que no se encuentran en el tramo pertenecen a un padre y los que sí al otro. Costes:
[416.92223608 413.09447801 418.30883452 417.7195358 421.02230916]
Media: 417.41347871288406
- **Cruce individual.** Al no existir ninguna relación entre genes continuos podemos cruzarlos de forma individual. Por cada gen hay un 50% de posibilidades de que pertenezca a un padre o a otro. Costes:
[414.13959031 415.71269351 410.32202425 418.93262673 412.02073776]
Costes: 414.2255345127038
- **Cruce individual con límite inferior de 205 plazas.** De esta forma aseguramos que los dos hijos vayan a cumplir las limitaciones de plazas. Sin embargo, esto ofrece un resultado peor posiblemente por restringir el espacio de búsqueda. Al ser los hijos individuos complementarios cuando tenemos un hijo con menos de 205 plazas el otro posee unos genes que no podría poseer si forzamos los 205 mínimos. Costes:
[414.51150091 412.80443757 414.9110361 418.93262673 423.38347716]
Media: 416.90861569526953

Número de individuos



Podemos ver que cuantos más individuos más espacio de búsqueda y se obtienen mejores resultados. La población será de 30 individuos.

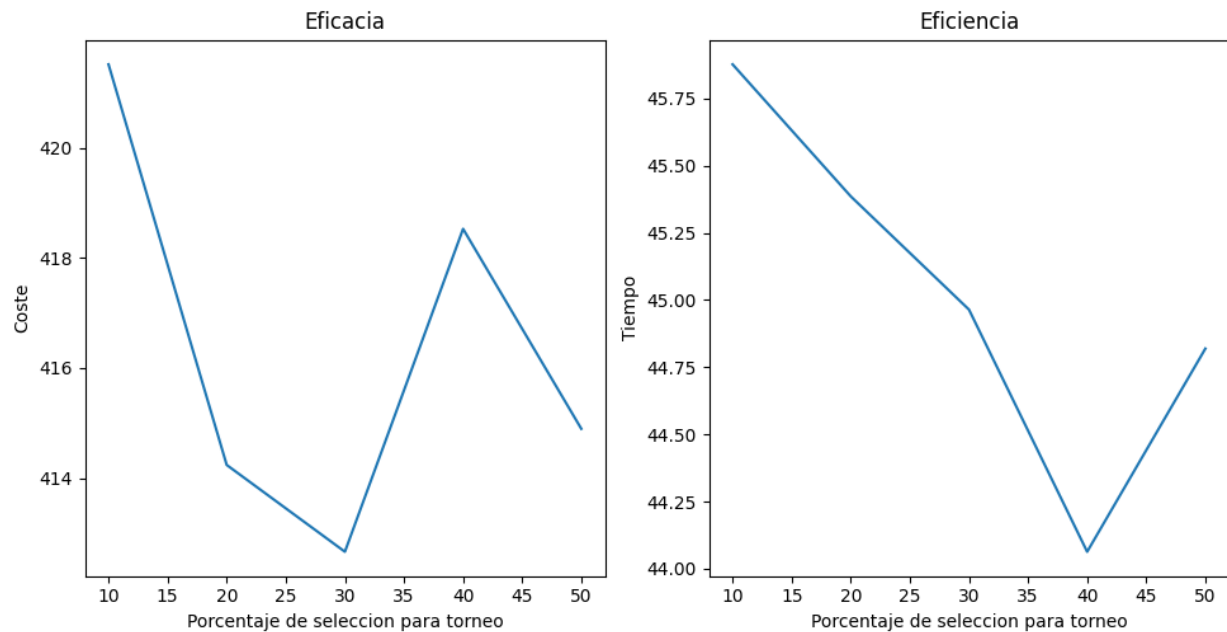
Iteraciones máximas



Costes: [416.68449992634726, 414.5552916293902, 412.663437539818, 412.2034935993867]

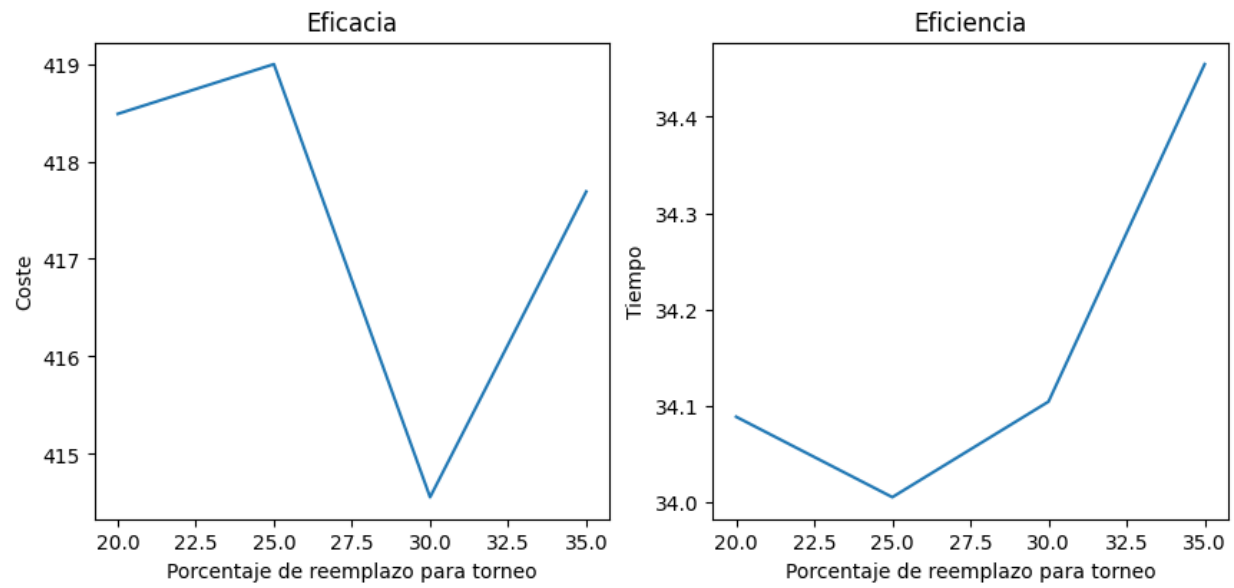
Aunque con 1200 iteraciones baja algo no es eficiente (mejora de menos de 1 punto), nos quedamos con 1000 iteraciones

Porcentaje selección



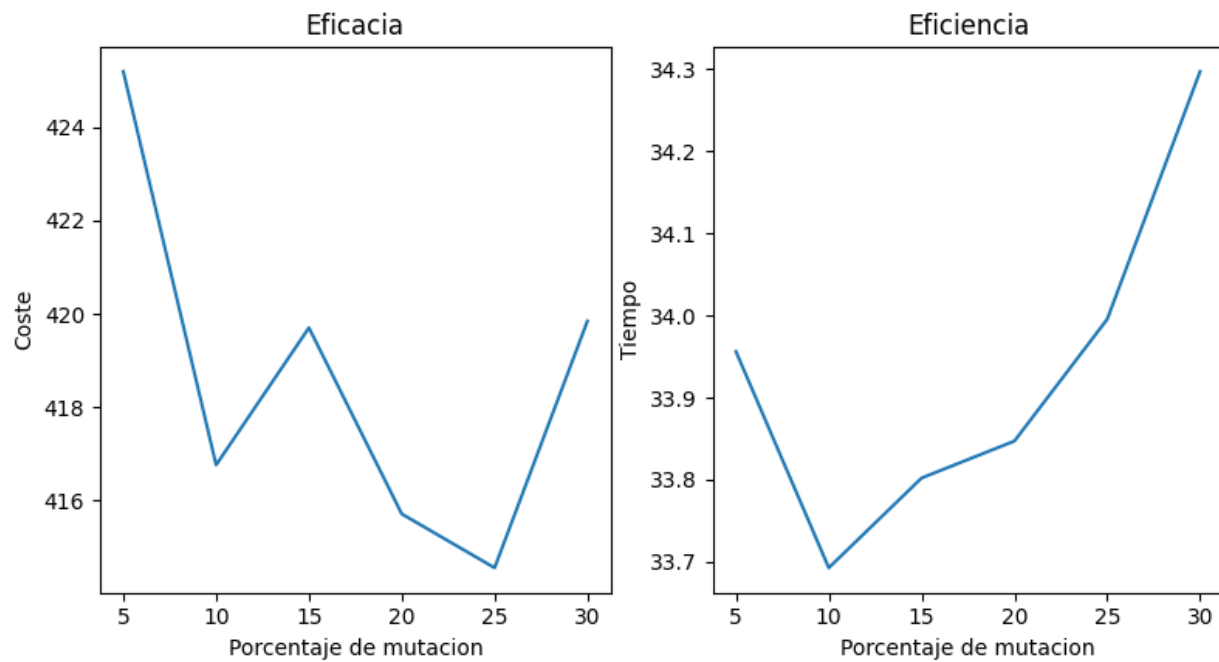
El porcentaje para el torneo de selección óptimo es 30%

Porcentaje reemplazo



Al igual que la selección el porcentaje óptimo es 30%

Porcentaje mutación



Es mejor 25% pero la practica nos obliga a usar como máximo 20%

Ejecución aleatoria

GENETICO BASICO

Parametros:

Numero de individuos: 30
Iteraciones maximas: 1000
Porcentaje de seleccion: 30
Porcentaje de mutacion: 20
Porcentaje de reemplazo: 30

Semilla: 939

Iteracion inicial. Mejor individuo 565.9861370329921 - [21 4 19 12 17 14 8 10 17 21 8 21 4 14 8 14]

Veces que mejora: 00000000000000000000

Iteracion final. Mejor individuo 418.5751480998792 - [21 12 21 18 17 13 12 13 10 16 17 23 2 9 15 14]

Semilla: 561

Iteracion inicial. Mejor individuo 582.2140505239403 - [5 5 17 15 15 5 19 17 17 11 9 19 19 7 19 15]

Veces que mejora: 00000000000000000000000000

Iteracion final. Mejor individuo 418.88237293973043 - [14 12 19 17 20 13 9 16 12 11 18 29 2 23 7 11]

Semilla: -113

Iteracion inicial. Mejor individuo 555.3922116680022 - [18 20 13 18 18 4 12 4 12 13 13 16 20 12 16 8]

Veces que mejora: 00000000000000000000000000000000

Iteracion final. Mejor individuo 411.17590050126535 - [18 12 17 14 20 13 12 12 11 16 18 27 2 13 17 13]

```
-----  
Semilla: 235  
-----
```

```
Iteracion inicial. Mejor individuo 553.5119002576105 - [ 8 16 20 18 13  4 16 18  8  8  8 20 13 10 20 18]
```

```
Veces que mejora: 00000000000000000000000000000000
```

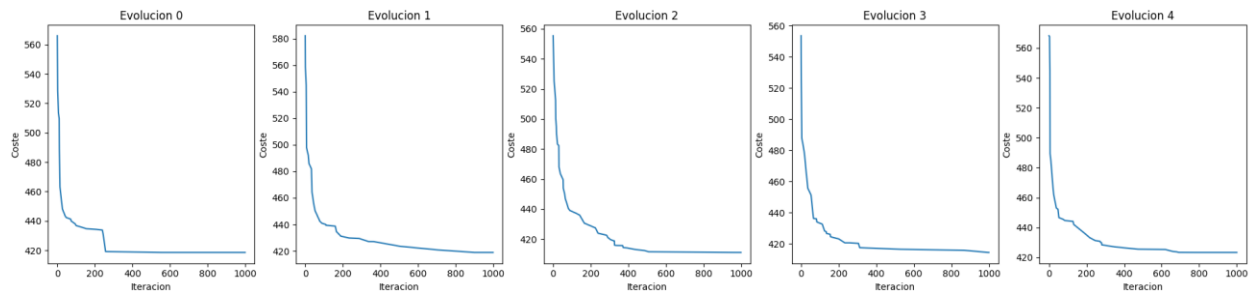
```
Iteracion final. Mejor individuo 414.35602390744396 - [18 12 19 25 20 14 14 12 10 14 18 25  2  8  8 14]
```

```
-----  
Semilla: -816  
-----
```

```
Iteracion inicial. Mejor individuo 568.2271451024152 - [12  6 20  6 18  4 10 16 14 16 18 20 18 20  6 10]
```

```
Veces que mejora: 00000000000000000000000000000000
```

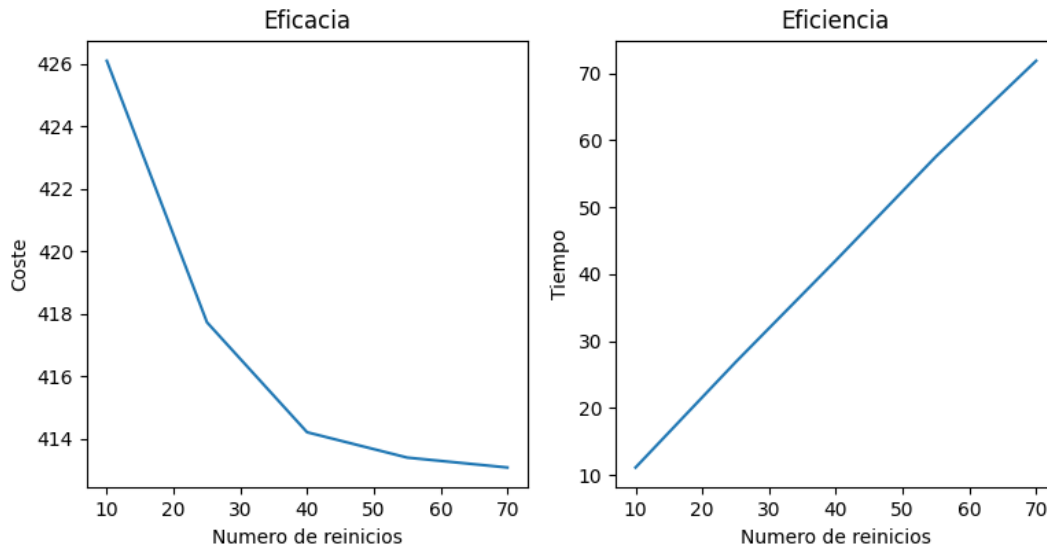
```
Iteracion final. Mejor individuo 423.29864761372596 - [16 12 19 15 20 16 18  6 12 16 18 26  2 20  9 10]
```



CHC

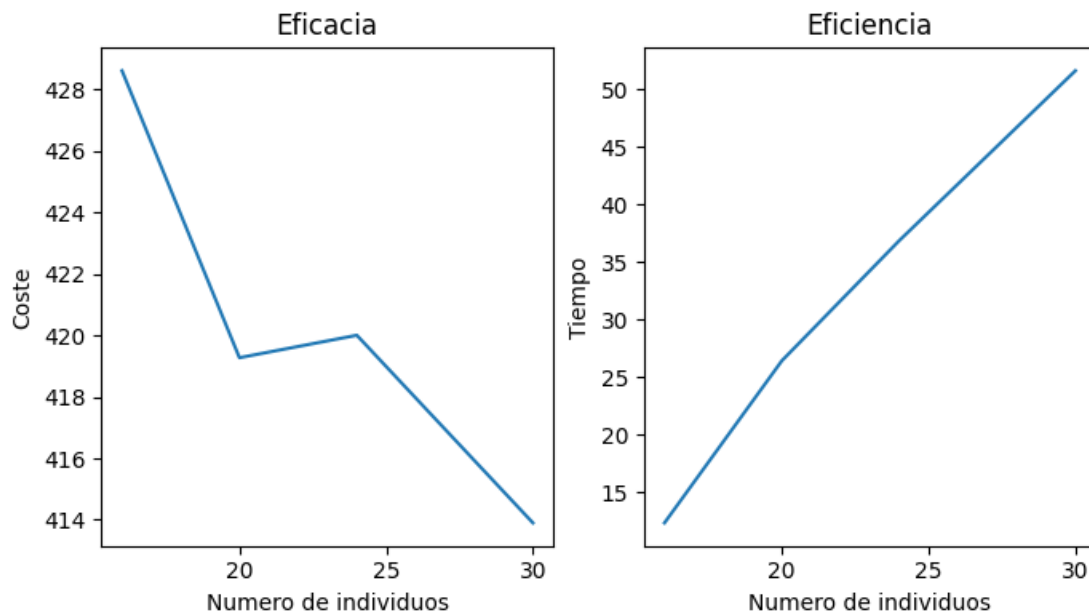
Para el CHC hemos aplicado la mutación parent centered además de la misma distribución gaussiana, cambiando la distribución de 2 a 3 ya que ofrece mejores resultados. Para el cruce intercambiamos la mitad de los valores distintos de los padres.

Número de reinicios



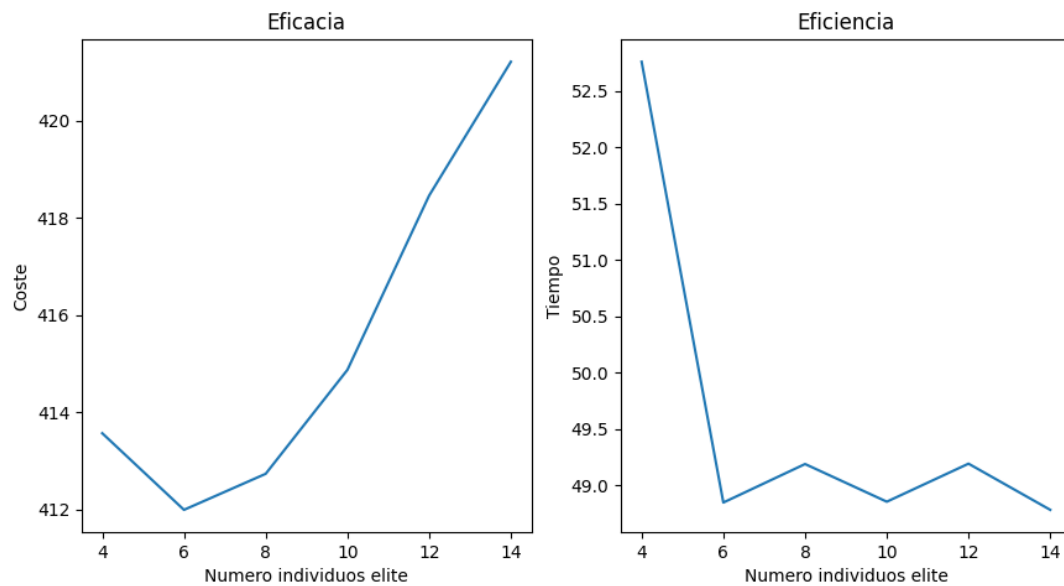
Elegimos 75 reinicios, ya que no es muy costoso temporalmente y aseguramos una buena mejora.

Número de individuos



Para el número de individuos al igual que en genético básico es mejor 30.

Número de individuos elite



El número de individuos elite óptimo para 30 individuos es 6

Ejecución aleatoria

CHC

Parametros:

Numero de individuos: 30

Reinicios maximos: 75

Individuos elites: 6

Semilla: 135

Iteracion inicial. Mejor individuo 569.1972061007934 - [11 4 15 19 19 6 11 22 17 6 13 22 8 17 8 15]

Reinicios: *****

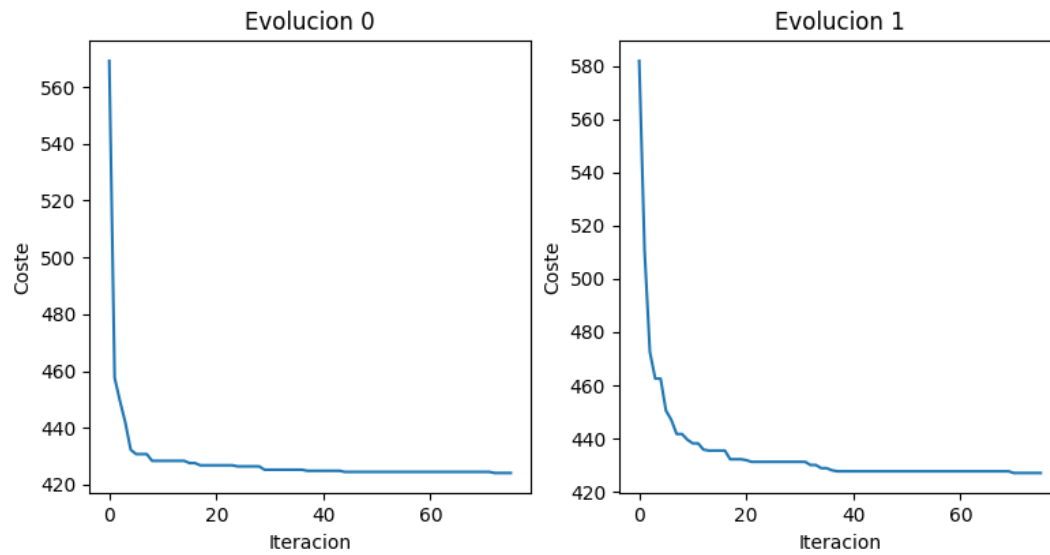
Iteracion final. Mejor individuo 424.09787391923885 - [12 13 19 15 20 15 18 10 12 14 18 26 4 10 17 12]

```
-----  
Semilla: 261  
-----
```

```
Iteracion inicial. Mejor individuo 582.007362782796 - [ 7 11 13 17 15  7 13 13  5 17 19 15 13 17 13 19]
```

```
Reinicios: *****
```

```
Iteracion final. Mejor individuo 427.0252071680402 - [20 12 23 16 18 10 12 12  8 18 18 30  2 17 13 16]
```

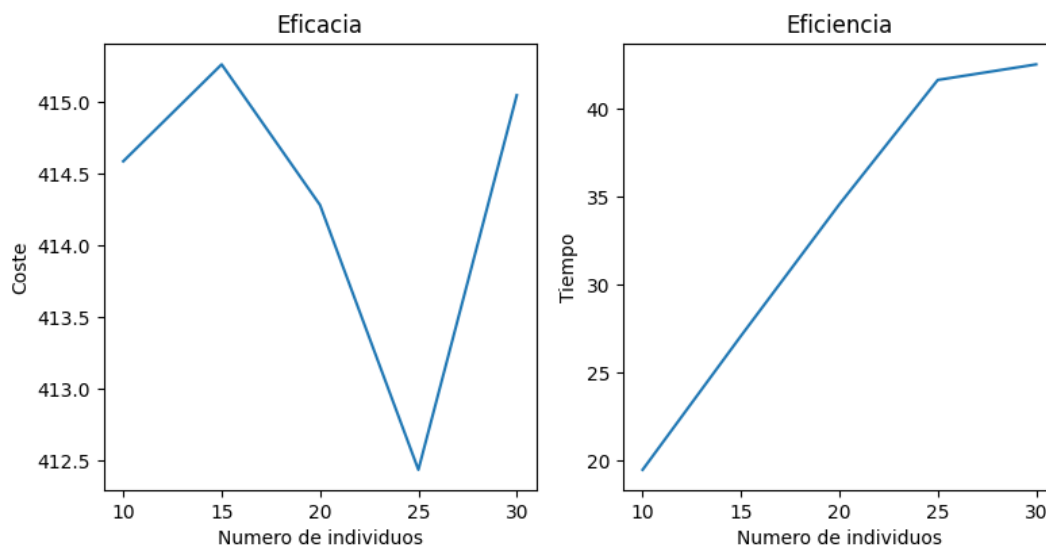


MULTIMODAL

Partimos del algoritmo genético básico, estacionario, con selección por torneo, etc. Por lo que implementamos el multimodal usando clearing.

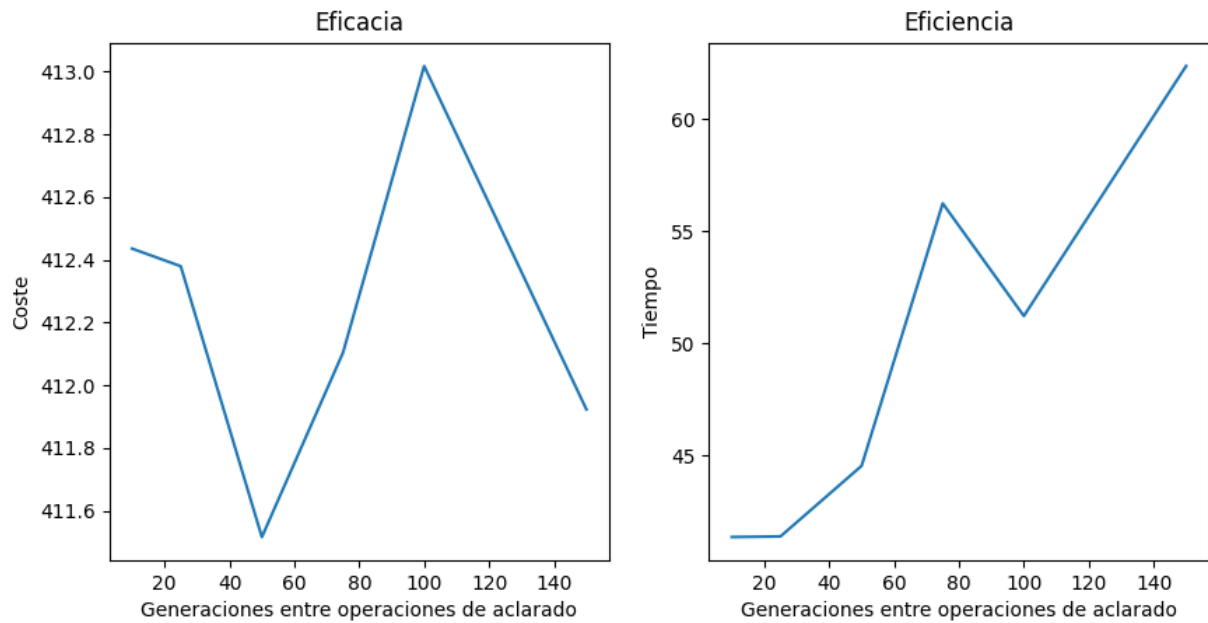
Al igual que en otros métodos vamos a hacer un estudio de parámetros, aunque el más importante va a ser el radio para la distancia Hamming. El algoritmo va a seguir corriendo hasta que se encuentre con 250 generaciones que no mejoran, por los parámetros elegidos en el genético básico. Hay mejora si la media de evaluaciones de la población supera en 0.1 puntos la media anterior.

Número de individuos



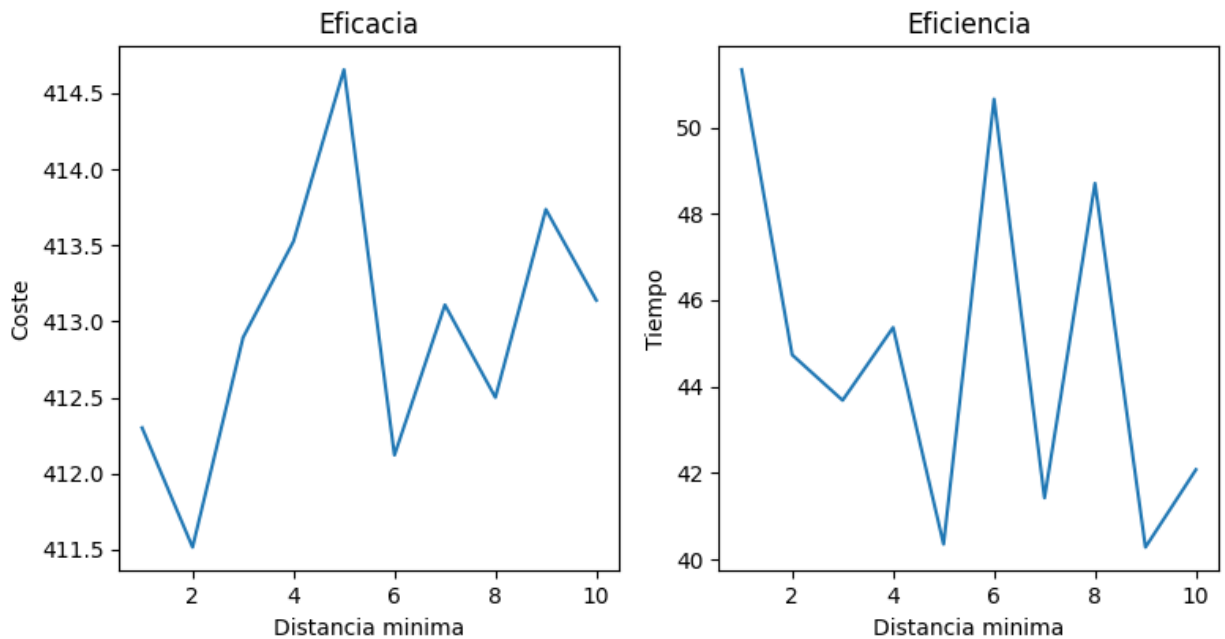
El número óptimo de individuos es 25, aunque realmente no hay demasiada diferencia con los demás valores, una diferencia de coste menor a 3 puntos.

Generaciones



El número de generaciones entre operaciones clearing óptimo es 50 generaciones.

Distancia mínima



Para la distancia mínima 2 es la que ofrece mejores resultados

Ejecución aleatoria

GENETICO MULTIMODAL

Parametros:

Numero de individuos: 25

Generaciones entre operaciones de aclarado: 50

Distancia minima: 2

Semilla: 257

Iteracion inicial. Mejor individuo 592.1921993037781 - [25 7 10 10 20 15 5 10 17 5 12 25 15 5 10 25]

Iteracion final. Mejor individuo 412.82216504753956 - [21 10 21 16 17 14 12 12 12 10 18 26 2 19 11 12]

Semilla: 474

Iteracion inicial. Mejor individuo 544.6230763292122 - [12 6 8 18 18 13 12 20 6 13 12 16 20 16 16 12]

Iteracion final. Mejor individuo 412.96058534185795 - [18 12 23 21 20 13 11 12 10 13 18 26 2 10 10 14]

Semilla: -899

Iteracion inicial. Mejor individuo 573.8805479415824 - [12 18 18 8 10 18 12 16 20 14 14 14 4 12 16 10]

Iteracion final. Mejor individuo 413.0453922864346 - [18 12 17 14 19 13 11 11 12 18 17 25 2 12 18 11]

```
-----  
Semilla: 384  
-----
```

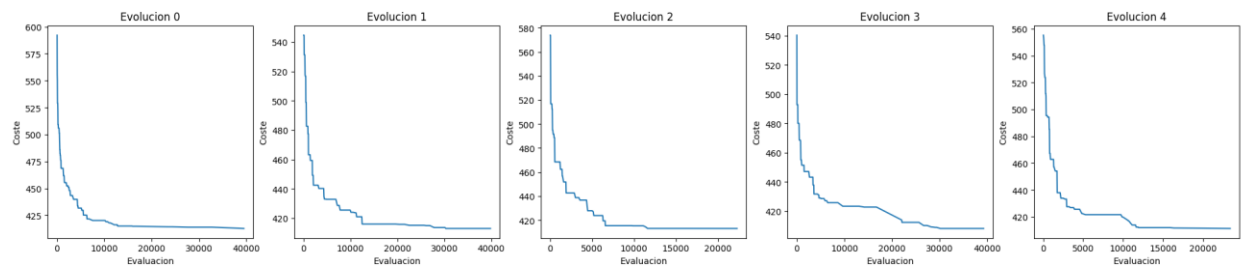
```
Iteracion inicial. Mejor individuo 540.2845349846443 - [18 10 18 12 18 12 20 10 18 18 12 13  4 10 18  8]
```

```
Iteracion final. Mejor individuo 408.1985703789265 - [18 12 18 14 20 14 10 12 12 15 18 26  2  9 21 12]
```

```
-----  
Semilla: 443  
-----
```

```
Iteracion inicial. Mejor individuo 555.2080297349894 - [10  6 17 19 19 10 19  4 14 10 17 19 19  6  4 21]
```

```
Iteracion final. Mejor individuo 411.28971854822476 - [17 12 19 16 20 14 10 12 12 11 18 26  2  3 27 13]
```



COMPARACIÓN

Algoritmo	Media Ev.	Mejor Ev.	Desv. Ev.	Media Coste	Mejor Coste	Desv. Coste
Búsqueda Local	1848	1340	463.66	474.75	453.24	21.5
VNS	35129	29161	3630	267.67	239.78	14.16
Genético Básico	62058.8	34270	28529.34	274.38	265.8	5.96
CHC	45700	40568	3965.14	274.99	267.54	9.6
Genético Multimodal	32711.2	23353	7583.47	265.51	258.76	5.47

Durante la implementación y las ejecuciones que se han mostrado de los algoritmos estamos hablando de coste por puntos, es decir la evaluación modificada, que tiene en cuenta el número de plazas. Sin embargo, en la comparación no tiene sentido usar esta evaluación de coste, ya que la anterior evaluación solo se usaba para no superar demasiado las plazas, pero no representa el coste realista final. Al entregar estas soluciones tendríamos que negociar que el número de plazas fuese superior a 220, en nuestro caso la media está en 230 plazas, por lo que no supondría un problema. Es por eso que los costes en la tabla de comparación es menor comparado con los costes vistos anteriormente.

El algoritmo **VNS** ofrece muy buenos resultados comparados con la búsqueda local. Esto se debe a que en la **búsqueda local** cuando generábamos vecinos lo hacíamos con el intercambio de plazas entre estaciones, sin embargo, en el **VNS** modificamos este número de plazas por lo que tenemos mayor espacio de búsqueda. Además, de la comparación entre distintas búsquedas locales. Si hubiésemos usado una generación de vecinos como en la **búsqueda local** hubiésemos tenido también resultados muy parecidos a la **búsqueda local**, mejorando sólo un poco. Ofrece mejor exploración.

El **genético básico** ofrece también muy buenos resultados, en este caso con mayor estabilidad, ya que tiene una desviación menor. Por semilla obtendrá de forma más consistente mejores resultados, aunque no baje algunas soluciones tanto como el **VNS**. Esto podría haberse mejorado probando distintas estrategias de selección, reemplazo, cruce y mutación. Además de comparar los resultados con un genético generacional.

El **CHC** ofrece costes muy parecidos con respecto al **genético básico**, sin embargo, este lo hace con menos evaluaciones, por lo que es más eficiente. Esto se puede deber a los reinicios y tener mayor exploración.

El **genético multimodal** es el que ofrece la mejor media de costes, también con mucha consistencia teniendo una desviación muy baja. Además, también es el que menos evaluaciones requiere. Estos buenos resultados pueden ir dados por la búsqueda de varios óptimos, no como los demás que convergen en cualquier óptimo.