

Índice

Introducción 4

Resultados obtenidos por cada Algoritmo 5

Greedy 5

Busqueda Aleatoria 6

Búsqueda Local 7

El Mejor 7

El Primer Mejor 8

El Mejor Aleatorizado 9

Enfriamiento Simulado 10

Búsqueda Tabú 11

Tabla global de resultados por datasets 12

Resumen st70 12

Gráficas st70 13

Resumen ch130 15

Gráficas ch130 16

Resumen a280 18

Gráficas a280 19

Resumen p654 21

Gráficas p654 22

Resumen vm1084 24

Gráficas vm1084 25

Resumen m1748 27

Gráficas vm1748 28

Análisis de los resultados obtenidos 30

Eficiencia temporal 30

Calidad y robustez 32

Número de evaluaciones 34

Mejor resultado individual obtenido 36

Ajuste de Parámetros 37

Ajuste del Enfriamiento Simulado 37

Mejora ES con dataset st70 37

Mejora ES con dataset ch130 37

Mejora ES con dataset a280 38

Mejora ES con dataset p654 38

Mejora ES con dataset vm1084 38

Mejora ES con dataset vm1784 38

Ajuste del Búsqueda Tabú 41

Mejora BT con dataset st70 41

Mejora BT con dataset ch130 42

Mejora BT con dataset a280 42

Mejora BT con dataset p654 42

Mejora BT con dataset vm1084 43

Mejora BT con dataset vm1784 43

Índice de Tablas

[Tabla 1. Tabla de Costes Greedy 5](#_Toc70254172)

[Tabla 2. Tabla de Costes Búsqueda Aleatoria 6](#_Toc70254173)

[Tabla 3. Tabla de Costes Búsqueda Local el Mejor 7](#_Toc70254174)

[Tabla 4. Tabla de Costes Búsqueda Local el Primer Mejor 8](#_Toc70254175)

[Tabla 5. Tabla de Costes Búsqueda Local el Mejor Aleatorizado 9](#_Toc70254176)

[Tabla 6. Tabla de Costes Enfriamiento Simulado 10](#_Toc70254177)

[Tabla 7. Tabla de Costes Búsqueda Tabú 11](#_Toc70254178)

[Tabla 8. Resumen resultados para el dataset st70 12](#_Toc70254179)

[Tabla 9. Resumen resultados para el dataset ch130 15](#_Toc70254180)

[Tabla 10. Resumen resultados para el dataset a280 18](#_Toc70254181)

[Tabla 11. Resumen resultados para el dataset p654 21](#_Toc70254182)

[Tabla 12. Resumen resultados para el dataset vm1084 24](#_Toc70254183)

[Tabla 13. Resumen resultados para el dataset vm1748 27](#_Toc70254184)

[Tabla 14. Mejora ES con dataset st70 37](#_Toc70254185)

[Tabla 15. Mejora ES con dataset ch130 37](#_Toc70254186)

[Tabla 16. Mejora ES con dataset a280 38](#_Toc70254187)

[Tabla 17. Mejora ES con dataset p654 38](#_Toc70254188)

[Tabla 18. Mejora ES con dataset vm1084 38](#_Toc70254189)

[Tabla 19. Mejora ES con dataset vm1784 38](#_Toc70254190)

[Tabla 20. Mejora BT con dataset st70 41](#_Toc70254191)

[Tabla 21. Mejora BT con dataset ch130 42](#_Toc70254192)

[Tabla 22. Mejora BT con dataset a280 42](#_Toc70254193)

[Tabla 23. Mejora BT con dataset p654 42](#_Toc70254194)

[Tabla 24. Mejora BT con dataset vm1084 43](#_Toc70254195)

[Tabla 25. Mejora BT con dataset vm1784 43](#_Toc70254196)

Índice de Gráficas

[Ilustración 1. Coste Medio st70 13](#_Toc70254197)

[Ilustración 2. Coste Mejor st70 13](#_Toc70254198)

[Ilustración 3. Evaluaciones Media st70 14](#_Toc70254199)

[Ilustración 4. Tiempo Medio st70 14](#_Toc70254200)

[Ilustración 5. Coste Medio ch130 16](#_Toc70254201)

[Ilustración 6. Coste Mejor ch130 16](#_Toc70254202)

[Ilustración 7. Evaluaciones Media ch130 17](#_Toc70254203)

[Ilustración 8. Tiempo Medio ch130 17](#_Toc70254204)

[Ilustración 9. Coste Medio a280 19](#_Toc70254205)

[Ilustración 10. Coste Mejor a280 19](#_Toc70254206)

[Ilustración 11. Evaluaciones Media a280 20](#_Toc70254207)

[Ilustración 12. Tiempo Medio a280 20](#_Toc70254208)

[Ilustración 13. Coste Medio p654 22](#_Toc70254209)

[Ilustración 14. Coste Mejor p654 22](#_Toc70254210)

[Ilustración 15. Evaluaciones Media p654 23](#_Toc70254211)

[Ilustración 16. Tiempo Medio p654 23](#_Toc70254212)

[Ilustración 17. Coste Medio vm1084 25](#_Toc70254213)

[Ilustración 18. Coste Mejor vm1084 25](#_Toc70254214)

[Ilustración 19. Evaluaciones Media vm1084 26](#_Toc70254215)

[Ilustración 20. Tiempo Medio vm1084 26](#_Toc70254216)

[Ilustración 21. Coste Medio vm1748 28](#_Toc70254217)

[Ilustración 22. Coste Mejor vm1748 28](#_Toc70254218)

[Ilustración 23. Evaluaciones Media vm1748 29](#_Toc70254219)

[Ilustración 24. Tiempo medio vm1748 29](#_Toc70254220)

[Ilustración 25. Eficiencia temporal de los Algoritmos 30](#_Toc70254221)

[Ilustración 26. Coste Medio Normalizado de los Algoritmos 32](#_Toc70254222)

[Ilustración 27. Desviación Típica Normalizada de los Algoritmos 32](#_Toc70254223)

[Ilustración 28. . Número de evaluaciones Normalizadas de los Algoritmos 34](#_Toc70254224)

[Ilustración 29. Coste Mejor Normalizado de los Algoritmos 36](#_Toc70254225)

[Ilustración 30. Coste Medio Normalizado ES Mejorado 39](file:///C:\Users\Ricar\Dropbox\Universidad\Grado\Asignaturas\4%20Cuarto\Modelos%20Bioinspirados%20y%20Heurísticas%20de%20Búsquedas\Practica\Practica%201\Memoria\Memoria.docx#_Toc70254226)

[Ilustración 31. Desviación Típica Normalizada ES Mejorado 39](file:///C:\Users\Ricar\Dropbox\Universidad\Grado\Asignaturas\4%20Cuarto\Modelos%20Bioinspirados%20y%20Heurísticas%20de%20Búsquedas\Practica\Practica%201\Memoria\Memoria.docx#_Toc70254227)

[Ilustración 32. Coste Mejor Normalizado ES Mejorado 40](file:///C:\Users\Ricar\Dropbox\Universidad\Grado\Asignaturas\4%20Cuarto\Modelos%20Bioinspirados%20y%20Heurísticas%20de%20Búsquedas\Practica\Practica%201\Memoria\Memoria.docx#_Toc70254228)

[Ilustración 33. Coste Medio Normalizado BT Mejorado 44](file:///C:\Users\Ricar\Dropbox\Universidad\Grado\Asignaturas\4%20Cuarto\Modelos%20Bioinspirados%20y%20Heurísticas%20de%20Búsquedas\Practica\Practica%201\Memoria\Memoria.docx#_Toc70254230)

[Ilustración 34. Coste Mejor Normalizado BT Mejorado 45](file:///C:\Users\Ricar\Dropbox\Universidad\Grado\Asignaturas\4%20Cuarto\Modelos%20Bioinspirados%20y%20Heurísticas%20de%20Búsquedas\Practica\Practica%201\Memoria\Memoria.docx#_Toc70254231)

[Ilustración 35. Desviación Típica Normalizada BT Mejorado 44](file:///C:\Users\Ricar\Dropbox\Universidad\Grado\Asignaturas\4%20Cuarto\Modelos%20Bioinspirados%20y%20Heurísticas%20de%20Búsquedas\Practica\Practica%201\Memoria\Memoria.docx#_Toc70254229)

Introducción

El objetivo de esta práctica es estudiar el funcionamiento de los siguientes algoritmos:

* Greedy
* Búsqueda Aleatoria
* Búsqueda Local
  + El Mejor
  + El Primer Mejor
  + El Mejor Aleatorizado
* Enfriamiento Simulado
* Búsqueda Tabú

Para ello, se implementará estos algoritmos para resolver el problema del *Viajante de Comercio*. Este problema que llamaremos a partir de ahora *TSP*, dada una serie de ciudades, consiste en encontrar el circuito de menor coste que parta de una ciudad concreta, pase por todas las demás una sola vez y retome a la ciudad de origen.

En nuestro caso, trabajaremos con 6 datasets del problema obtenidas de la biblioteca *TSPLIB* que serán las siguientes:

* St70 -> con un tamaño de 70 ciudades y un coste óptimo de 675.
* Ch130 -> con un tamaño de 130 ciudades y un coste óptimo de 6.110.
* A280 -> con un tamaño de 280 ciudades y un coste óptimo de 2.579.
* P654 -> con un tamaño de 654 ciudades y un coste óptimo de 34.643.
* Vm1084 -> con un tamaño de 1084 ciudades y un coste óptimo de 239.297.
* Vm1748 -> con un tamaño de 1748 ciudades.

Se estudiará cómo de buenas son las soluciones que ofrece cada algoritmo, cuántas veces se llama a la función de evaluación, que en nuestro caso es la función *calculaCoste*, el número de iteraciones que realiza y el tiempo que tarda en realizarlas.

Resultados obtenidos por cada Algoritmo

Greedy

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | St70 | | Ch130 | | A280 | | P654 | | Vm1084 | | Vm1748 | |
|  | **Coste** | **#EV** | **Coste** | **#EV** | **Coste** | **#EV** | **Coste** | **#EV** | **Coste** | **#EV** | **Coste** | **#EV** |
| Ejecución 1 | 830 | 1 | 7.579 | 1 | 3.157 | 1 | 43.457 | 1 | 301.476 | 1 | 408.101 | 1 |
| Media | 830 | 1 | 7.579 | 1 | 3.157 | 1 | 43.457 | 1 | 301.476 | 1 | 408.101 | 1 |
| Desviación Típica | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Tabla 1. Tabla de Costes Greedy

Búsqueda Aleatoria

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | St70 | | Ch130 | | A280 | | | P654 | | Vm1084 | | Vm1748 | | |
|  | **Coste** | **#EV** | **Coste** | **#EV** | | **Coste** | **#EV** | **Coste** | **#EV** | **Coste** | **#EV** | **Coste** | **#EV** |
| Semilla 0 | 2868 | 112.000 | 38.925 | 208.000 | | 29.664 | 448.000 | 1.794.114 | 1.046.400 | 8.009.568 | 1.734.400 | 14.134.673 | 2.796.800 |
| Semilla 1 | 2770 | 112.000 | 38.437 | 208.000 | | 29.496 | 448.000 | 1.809.713 | 1.046.400 | 8.017.444 | 1.734.400 | 14.134.601 | 2.796.800 |
| Semilla 2 | 2809 | 112.000 | 38.240 | 208.000 | | 29.546 | 448.000 | 1.798.827 | 1.046.400 | 7.969.912 | 1.734.400 | 14.128.347 | 2.796.800 |
| Semilla 3 | 2875 | 112.000 | 38.898 | 208.000 | | 29.174 | 448.000 | 1.806.317 | 1.046.400 | 7.958.466 | 1.734.400 | 14.163.069 | 2.796.800 |
| Semilla 4 | 2842 | 112.000 | 38.616 | 208.000 | | 29.467 | 448.000 | 1.803.585 | 1.046.400 | 7.963.922 | 1.734.400 | 14.131.611 | 2.796.800 |
| Semilla 5 | 2812 | 112.000 | 39.038 | 208.000 | | 29.598 | 448.000 | 1.807.866 | 1.046.400 | 7.939.222 | 1.734.400 | - | 2.796.800 |
| Semilla 6 | 2780 | 112.000 | 38.459 | 208.000 | | 29.771 | 448.000 | 1.807.271 | 1.046.400 | 7.945.424 | 1.734.400 | - | 2.796.800 |
| Semilla 7 | 2850 | 112.000 | 38.584 | 208.000 | | 29.469 | 448.000 | 1.817.079 | 1.046.400 | 7.992.968 | 1.734.400 | - | 2.796.800 |
| Semilla 8 | 2806 | 112.000 | 39.108 | 208.000 | | 29.799 | 448.000 | 1.819.544 | 1.046.400 | 7.985.354 | 1.734.400 | - | 2.796.800 |
| Semilla 9 | 2823 | 112.000 | 38.603 | 208.000 | | 29.353 | 448.000 | 1.810.228 | 1.046.400 | 7.967.929 | 1.734.400 | - | 2.796.800 |
| Media | 2823,5 | 112.000 | 38.690,8 | 208.000 | | 29.533,7 | 448.000 | 1.807.454,4 | 1.046.400 | 7.975.020,9 | 1.734.400 | 14.138.460,2 | 2.796.800 |
| Desviación Típica | 35,087 | 0 | 286,445 | 0 | | 188,577 | 0 | 7.604,478 | 0 | 25.934,269 | 0 | 14.000,158 | 0 |

Tabla 2. Tabla de Costes Búsqueda Aleatoria

Búsqueda Local

Búsqueda Local el Mejor

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | St70 | | Ch130 | | A280 | | P654 | | | Vm1084 | | | Vm1748 | | |
|  | **Coste** | **#EV** | **Coste** | **#EV** | **Coste** | **#EV** | **Coste** | **#EV** | **Coste** | | **#EV** | **Coste** | | **#EV** |
| Semilla 0 | 1.189 | 112.000 | 23.938 | 208.000 | 26.991 | 448.000 | 1.947.268 | 1.046.400 | 8.483.753 | | 1.734.400 | 14.684.089 | | 2.796.800 |
| Semilla 1 | 1.267 | 112.000 | 22.697 | 208.000 | 26.657 | 448.000 | 1.928.171 | 1.046.400 | 8.357.385 | | 1.734.400 | 14.952.989 | | 2.796.800 |
| Semilla 2 | 1.144 | 112.000 | 23.819 | 208.000 | 28.456 | 448.000 | 1.926.013 | 1.046.400 | 8.491.845 | | 1.734.400 | 15.106.312 | | 2.796.800 |
| Semilla 3 | 1.266 | 112.000 | 24.761 | 208.000 | 27.242 | 448.000 | 1.956.574 | 1.046.400 | 8.447.783 | | 1.734.400 | 14.861.355 | | 2.796.800 |
| Semilla 4 | 1.270 | 112.000 | 23.218 | 208.000 | 26.300 | 448.000 | 1.903.639 | 1.046.400 | 8.365.862 | | 1.734.400 | 14.962.200 | | 2.796.800 |
| Semilla 5 | 1.358 | 112.000 | 23.262 | 208.000 | 26.787 | 448.000 | 1.975.520 | 1.046.400 | 8.518.458 | | 1.734.400 | 14.757.511 | | 2.796.800 |
| Semilla 6 | 1.204 | 112.000 | 23.738 | 208.000 | 26.619 | 448.000 | 1.887.608 | 1.046.400 | 8.539.773 | | 1.734.400 | 14.822.860 | | 2.796.800 |
| Semilla 7 | 1.104 | 112.000 | 23.022 | 208.000 | 26.315 | 448.000 | 1.877.150 | 1.046.400 | 8.456.359 | | 1.734.400 | 14.769.694 | | 2.796.800 |
| Semilla 8 | 1.161 | 112.000 | 23.752 | 208.000 | 25.229 | 448.000 | 1.972.958 | 1.046.400 | 8.280.363 | | 1.734.400 | 14.609.878 | | 2.796.800 |
| Semilla 9 | 1.305 | 112.000 | 23.460 | 208.000 | 28.413 | 448.000 | 1.947.949 | 1.046.400 | 8.519.567 | | 1.734.400 | 14.903.149 | | 2.796.800 |
| Media | 1.226,8 | 112.000 | 23.566,7 | 208.000 | 26.900,9 | 448.000 | 1.932.285 | 1.046.400 | 8.446.114,8 | | 1.734.400 | 14.843.003,7 | | 2.796.800 |
| Desviación Típica | 79,275 | 0 | 574,499 | 0 | 970,726 | 0 | 34150,197 | 0 | 84.855,028 | | 0 | 146.166,027 | | 0 |

Tabla 3. Tabla de Costes Búsqueda Local el Mejor

Búsqueda Local el Primer Mejor

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | St70 | | Ch130 | | | A280 | | | P654 | | | Vm1084 | | | Vm1748 | | |
|  | **Coste** | **#EV** | **Coste** | **#EV** | **Coste** | | **#EV** | **Coste** | | **#EV** | **Coste** | | **#EV** | **Coste** | | **#EV** |
| Semilla 0 | 1.293 | 112.000 | 15.464 | 208.000 | 9.917 | | 448.000 | 478.747 | | 1.046.400 | 3.017.146 | | 1.734.400 | 7.992.956 | | 2.796.800 |
| Semilla 1 | 1.148 | 61.412 | 16.271 | 208.000 | 9.321 | | 448.000 | 507.788 | | 1.046.400 | 3.455.899 | | 1.734.400 | 7.776.885 | | 2.796.800 |
| Semilla 2 | 1.081 | 84.250 | 16.212 | 208.000 | 10.675 | | 448.000 | 545.978 | | 1.046.400 | 2.916.217 | | 1.734.400 | 7.946.362 | | 2.796.800 |
| Semilla 3 | 1.058 | 81.856 | 15.758 | 208.000 | 9.065 | | 448.000 | 455.930 | | 1.046.400 | 3.733.810 | | 1.734.400 | 7.950.083 | | 2.796.800 |
| Semilla 4 | 1.192 | 66.207 | 14.469 | 208.000 | 8.350 | | 448.000 | 807.625 | | 1.046.400 | 3.763.698 | | 1.734.400 | 8.218.826 | | 2.796.800 |
| Semilla 5 | 1.026 | 86.453 | 14.916 | 208.000 | 9.248 | | 448.000 | 753.523 | | 1.046.400 | 3.910.507 | | 1.734.400 | 7.865.409 | | 2.796.800 |
| Semilla 6 | 973 | 98.853 | 14.272 | 208.000 | 9.838 | | 448.000 | 654.645 | | 1.046.400 | 3.217.647 | | 1.734.400 | 8.096.013 | | 2.796.800 |
| Semilla 7 | 1.075 | 84.283 | 13.875 | 208.000 | 8.408 | | 448.000 | 474.710 | | 1.046.400 | 3.200.086 | | 1.734.400 | 8.042.342 | | 2.796.800 |
| Semilla 8 | 966 | 58.654 | 14.803 | 208.000 | 8.348 | | 448.000 | 469.655 | | 1.046.400 | 3.357.816 | | 1.734.400 | 7.553.040 | | 2.796.800 |
| Semilla 9 | 1.069 | 67.221 | 13.875 | 208.000 | 8.480 | | 448.000 | 481.185 | | 1.046.400 | 4.391.955 | | 1.734.400 | 8.005.377 | | 2.796.800 |
| Media | 1.088,1 | 80.118,9 | 14.991,5 | 208.000 | 9.165 | | 448.000 | 562.978,6 | | 1.046.400 | 3.496.478,1 | | 1.734.400 | 7.944.729,3 | | 2.796.800 |
| Desviación Típica | 99,929 | 17.047,326 | 898,305 | 0 | 796,781 | | 0 | 128.945,285 | | 0 | 453.867,677 | | 0 | 183.018,885 | | 0 |

Tabla 4. Tabla de Costes Búsqueda Local el Primer Mejor

Búsqueda Local el Mejor Aleatorizado

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | St70 | | Ch130 | | A280 | | P654 | | Vm1084 | | | Vm1748 | | |
|  | **Coste** | **#EV** | **Coste** | **#EV** | **Coste** | **#EV** | **Coste** | **#EV** | **Coste** | **#EV** | **Coste** | | **#EV** |
| Semilla 0 | 1.383 | 4.095 | 17.350 | 10.465 | 8.609 | 76.440 | 413.659 | 480.363 | 1.634.774 | 1.531.692 | 3.030.078 | | 2.796.800 |
| Semilla 1 | 1.298 | 3.920 | 13.363 | 18.135 | 9.015 | 77.840 | 376.858 | 559.170 | 1.531.399 | 1.642.802 | 2.972.055 | | 2.796.800 |
| Semilla 2 | 1.273 | 4.515 | 13.358 | 15.535 | 9.346 | 75.320 | 411.435 | 683.757 | 1.581.377 | 1.589.144 | 3.014.399 | | 2.796.800 |
| Semilla 3 | 1.330 | 4.480 | 14.824 | 13.325 | 9.413 | 67.200 | 400.989 | 535.953 | 1.578.159 | 1.412.452 | 3.015.841 | | 2.796.800 |
| Semilla 4 | 1.440 | 3.325 | 13.759 | 14.885 | 8.900 | 64.120 | 345.298 | 646.479 | 1.702.386 | 1.455.270 | 3.139.447 | | 2.796.800 |
| Semilla 5 | 1.543 | 2.695 | 14.604 | 14.885 | 8.462 | 82.740 | 415.186 | 544.128 | 1.607.308 | 1.480.744 | 2.949.476 | | 2.796.800 |
| Semilla 6 | 1.628 | 2.135 | 14.349 | 18.330 | 10.170 | 48.860 | 399.931 | 501.291 | 1.600.985 | 1.734.400 | 3.120.978 | | 2.796.800 |
| Semilla 7 | 1.311 | 3.430 | 12.493 | 15.275 | 8.801 | 73.220 | 425.366 | 509.466 | 1.662.859 | 1.444.430 | 3.000.969 | | 2.796.800 |
| Semilla 8 | 1.390 | 4.165 | 13.312 | 17.680 | 8.500 | 63.140 | 368.881 | 693.894 | 1.460.801 | 1.597.274 | 2.845.352 | | 2.796.800 |
| Semilla 9 | 1.513 | 3.185 | 14.599 | 15.145 | 10.021 | 43.680 | 416.883 | 513.063 | 1.535.821 | 1.594.564 | 2.959.868 | | 2.796.800 |
| Media | 1.410,9 | 3.594,5 | 14.201,1 | 15.392 | 9.123,7 | 67.256 | 397.448,6 | 566.756,4 | 1.589.586,9 | 1.548.277,2 | 3.004.846,3 | | 2.796.800 |
| Desviación Típica | 118,019 | 783,154 | 1.331,786 | 2.365,301 | 604,681 | 12.726,779 | 25.622,367 | 78.628,493 | 69.632,907 | 101.384,171 | 84.373,737 | | 0 |

Tabla 5. Tabla de Costes Búsqueda Local el Mejor Aleatorizado

Enfriamiento Simulado

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | St70 | | | Ch130 | | | A280 | | | P654 | | | Vm1084 | | | Vm1748 | | |
|  | **Coste** | **#EV** | **Coste** | | **#EV** | **Coste** | | **#EV** | **Coste** | | **#EV** | **Coste** | | **#EV** | **Coste** | | **#EV** |
| Semilla 0 | 1.263 | 112.000 | 10.025 | | 208.000 | 6.577 | | 448.000 | 301.305 | | 1.046.400 | 1.244.556 | | 1.734.400 | 2.451.541 | | 2.796.800 |
| Semilla 1 | 913 | 112.000 | 10.858 | | 208.000 | 5.493 | | 448.000 | 331.266 | | 1.046.400 | 1.300.555 | | 1.734.400 | 2.157.007 | | 2.796.800 |
| Semilla 2 | 1.081 | 112.000 | 9.945 | | 208.000 | 6.438 | | 448.000 | 332.744 | | 1.046.400 | 1.376.859 | | 1.734.400 | 2.295.151 | | 2.796.800 |
| Semilla 3 | 941 | 112.000 | 11.370 | | 208.000 | 6.127 | | 448.000 | 341.234 | | 1.046.400 | 1.233.964 | | 1.734.400 | 2.419.678 | | 2.796.800 |
| Semilla 4 | 1.041 | 112.000 | 11.998 | | 208.000 | 5.755 | | 448.000 | 324.331 | | 1.046.400 | 1.290.710 | | 1.734.400 | 2.271.689 | | 2.796.800 |
| Semilla 5 | 1.109 | 112.000 | 10.413 | | 208.000 | 6.166 | | 448.000 | 378.196 | | 1.046.400 | 1.351.564 | | 1.734.400 | 2.317.825 | | 2.796.800 |
| Semilla 6 | 1.016 | 112.000 | 11.770 | | 208.000 | 5.594 | | 448.000 | 329.041 | | 1.046.400 | 1.257.896 | | 1.734.400 | 2.373.466 | | 2.796.800 |
| Semilla 7 | 1.030 | 112.000 | 12.260 | | 208.000 | 5.871 | | 448.000 | 305.838 | | 1.046.400 | 1.236.590 | | 1.734.400 | 2.250.417 | | 2.796.800 |
| Semilla 8 | 900 | 112.000 | 9.567 | | 208.000 | 6.032 | | 448.000 | 305.848 | | 1.046.400 | 1.308.672 | | 1.734.400 | 2.199.872 | | 2.796.800 |
| Semilla 9 | 1.118 | 112.000 | 10.619 | | 208.000 | 5.958 | | 448.000 | 327.907 | | 1.046.400 | 1.334.309 | | 1.734.400 | 2.299.196 | | 2.796.800 |
| Media | 1.041,2 | 112.000 | 10.882,5 | | 208.000 | 6.001,1 | | 448.000 | 327.771 | | 1.046.400 | 1.293.567,5 | | 1.734.400 | 2.303.584,2 | | 2.796.800 |
| Desviación Típica | 109,771 | 0 | 931,091 | | 0 | 344,768 | | 0 | 22.202,27 | | 0 | 50.243,484 | | 0 | 92.274,215 | | 0 |

Tabla 6. Tabla de Costes Enfriamiento Simulado

Búsqueda Tabú

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | St70 | | | Ch130 | | | A280 | | | P654 | | | Vm1084 | | | Vm1748 | | |
|  | **Coste** | **#EV** | **Coste** | | **#EV** | **Coste** | | **#EV** | **Coste** | | **#EV** | **Coste** | | **#EV** | **Coste** | | **#EV** |
| Semilla 0 | 931 | 112.000 | 12.432 | | 208.000 | 3.204 | | 448.000 | 108.089 | | 1.046.400 | 1.938.944 | | 1.734.400 | 3.369.023 | | 2.796.800 |
| Semilla 1 | 1.029 | 112.000 | 12.368 | | 208.000 | 2.976 | | 448.000 | 122.300 | | 1.046.400 | 1.843.295 | | 1.734.400 | 3.149.172 | | 2.796.800 |
| Semilla 2 | 1.018 | 112.000 | 12.475 | | 208.000 | 3.026 | | 448.000 | 108.059 | | 1.046.400 | 1.770.743 | | 1.734.400 | 3.407.101 | | 2.796.800 |
| Semilla 3 | 923 | 112.000 | 12.258 | | 208.000 | 3.334 | | 448.000 | 118.714 | | 1.046.400 | 1.862.866 | | 1.734.400 | 3.221.902 | | 2.796.800 |
| Semilla 4 | 1.035 | 112.000 | 11.951 | | 208.000 | 3.223 | | 448.000 | 111.457 | | 1.046.400 | 1.755.767 | | 1.734.400 | 3.405.218 | | 2.796.800 |
| Semilla 5 | 993 | 112.000 | 11.610 | | 208.000 | 3.154 | | 448.000 | 109.016 | | 1.046.400 | 1.785.529 | | 1.734.400 | 3.200.750 | | 2.796.800 |
| Semilla 6 | 1.001 | 112.000 | 11.925 | | 208.000 | 3.726 | | 448.000 | 114.339 | | 1.046.400 | 1.889.042 | | 1.734.400 | 3.302.593 | | 2.796.800 |
| Semilla 7 | 969 | 112.000 | 12.158 | | 208.000 | 3.575 | | 448.000 | 108.905 | | 1.046.400 | 1.801.132 | | 1.734.400 | 3.268.092 | | 2.796.800 |
| Semilla 8 | 945 | 112.000 | 12.328 | | 208.000 | 3.436 | | 448.000 | 108.388 | | 1.046.400 | 1.910.291 | | 1.734.400 | 3.417.383 | | 2.796.800 |
| Semilla 9 | 967 | 112.000 | 12.126 | | 208.000 | 3.518 | | 448.000 | 110.058 | | 1.046.400 | 1.862.887 | | 1.734.400 | 3.250.772 | | 2.796.800 |
| Media | 981,1 | 112.000 | 12.163,1 | | 208.000 | 3.317,2 | | 448.000 | 111.932,5 | | 1.046.400 | 1.842.049,6 | | 1.734.400 | 3.299.200,6 | | 2.796.800 |
| Desviación Típica | 40,322 | 0 | 270,13 | | 0 | 244,456 | | 0 | 4.977,913 | | 0 | 61.932,684 | | 0 | 96.162,588 | | 0 |

Tabla 7. Tabla de Costes Búsqueda Tabú

Tabla global de resultados por datasets

Resumen st70

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Coste Medio | Mejor  (675) | Semilla | Desviación Típica | #EV  Media | Tiempo Medio |
| Greedy | 830 | 830 | - | 0 | 1 | 0,0039s |
| Búsqueda Aleatoria | 2823,5 | 2770 | 1 | 35,087 | 112.000 | 12,52s |
| BL El Mejor | 1.226,8 | 1.104 | 7 | 79,275 | 112.000 | 0,493s |
| BL El Primer Mejor | 1.088,1 | 966 | 8 | 99,929 | 80.118,9 | 0,41s |
| BL El Mejor Aleatorizado | 1.410,9 | 1.273 | 2 | 118,019 | 3.594,5 | 0,03s |
| Enfriamiento Simulado | 1.041,2 | 900 | 8 | 109,771 | 112.000 | 1,08s |
| Búsqueda Tabú | 981,1 | 923 | 3 | 40,322 | 112.000 | 1,22s |

*Tabla 8. Resumen resultados para el dataset st70*

Gráficas st70

Ilustración 1. Coste Medio st70

Ilustración 2. Coste Mejor st70

Ilustración 3. Evaluaciones Media st70

Ilustración 4. Tiempo Medio st70

Resumen ch130

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Coste Medio | Mejor  (6.110) | Semilla | Desviación Típica | #EV  Media | Tiempo Medio |
| Greedy | 7.579 | 7.579 | - | 0 | 1 | 0,0069s |
| Búsqueda Aleatoria | 38.690,8 | 38.240 | 2 | 286,445 | 208.000 | 41,36s |
| BL El Mejor | 23.566,7 | 22.697 | 1 | 574,499 | 208.000 | 1,07s |
| BL El Primer Mejor | 14.991,5 | 13.875 | 7 | 898,305 | 208.000 | 1,19s |
| BL El Mejor Aleatorizado | 14.201,1 | 12.493 | 7 | 1.331,786 | 15.392 | 0,149s |
| Enfriamiento Simulado | 10.882,5 | 9.567 | 8 | 931,091 | 208.000 | 2,00s |
| Búsqueda Tabú | 12.163,1 | 11.610 | 5 | 270,13 | 208.000 | 2,92s |

*Tabla 9. Resumen resultados para el dataset ch130*

Gráficas ch130

Ilustración 5. Coste Medio ch130

Ilustración 6. Coste Mejor ch130

Ilustración 7. Evaluaciones Media ch130

Ilustración 8. Tiempo Medio ch130

Resumen a280

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Coste Medio | Mejor  (2.579) | Semilla | Desviación Típica | #EV  Media | Tiempo Medio |
| Greedy | 3.157 | 3.157 | - | 0 | 1 | 0,01s |
| Búsqueda Aleatoria | 29.533,7 | 29.174 | 3 | 188,577 | 448.000 | 186,97s |
| BL El Mejor | 26.900,9 | 25.229 | 8 | 970,726 | 448.000 | 2,475s |
| BL El Primer Mejor | 9.165 | 8.348 | 8 | 796,781 | 448.000 | 2,653s |
| BL El Mejor Aleatorizado | 9.123,7 | 8.462 | 5 | 604,681 | 67.256 | 0,598s |
| Enfriamiento Simulado | 6.001,1 | 5.493 | 1 | 344,768 | 448.000 | 6,21s |
| Búsqueda Tabú | 3.317,2 | 2.976 | 1 | 244,456 | 448.000 | 9,26s |

*Tabla 10. Resumen resultados para el dataset a280*

Gráficas a280

Ilustración 9. Coste Medio a280

Ilustración 10. Coste Mejor a280

Ilustración 11. Evaluaciones Media a280

Ilustración 12. Tiempo Medio a280

Resumen p654

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Coste Medio | Mejor  (34.643) | Semilla | Desviación Típica | #EV  Media | Tiempo Medio |
| Greedy | 43.457 | 43.457 | - | 0 | 1 | 0,076s |
| Búsqueda Aleatoria | 1.807.454,4 | 1.794.114 | 0 | 7.604,478 | 1.046.400 | 1123,51s |
| BL El Mejor | 1.932.285 | 1.877.150 | 7 | 34.150,197 | 1.046.400 | 7,52s |
| BL El Primer Mejor | 562.978,6 | 455.930 | 3 | 128.945,285 | 1.046.400 | 9,23s |
| BL El Mejor Aleatorizado | 397.448,6 | 345.298 | 4 | 25.622,367 | 566.756,4 | 6,07s |
| Enfriamiento Simulado | 327.771 | 301.305 | 0 | 22.202,27 | 1.046.400 | 17,50s |
| Búsqueda Tabú | 111.932,5 | 108.059 | 2 | 4.977,913 | 1.046.400 | 42,24s |

*Tabla 11. Resumen resultados para el dataset p654*

Gráficas p654

Ilustración 13. Coste Medio p654

Ilustración 14. Coste Mejor p654

Ilustración 15. Evaluaciones Media p654

Ilustración 16. Tiempo Medio p654

Resumen vm1084

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Coste Medio | Mejor  (239.297) | Semilla | Desviación Típica | #EV  Media | Tiempo Medio |
| Greedy | 301.476 | 301.476 | - | 0 | 1 | 0,26s |
| Búsqueda Aleatoria | 7.975.020,9 | 7.939.222 | 5 | 25.934,269 | 1.734.400 | 4.920,7s |
| BL El Mejor | 8.446.114,8 | 8.280.363 | 8 | 84.855,028 | 1.734.400 | 14,3s |
| BL El Primer Mejor | 3.496.478,1 | 2.916.217 | 2 | 453.867,677 | 1.734.400 | 19,69s |
| BL El Mejor Aleatorizado | 1.589.586,9 | 1.460.801 | 8 | 69.632,907 | 1.548.277,2 | 22,49s |
| Enfriamiento Simulado | 1.293.567,5 | 1.233.964 | 3 | 50.243,484 | 1.734.400 | 32,37s |
| Búsqueda Tabú | 1.842.049,6 | 1.755.767 | 4 | 61.932,684 | 1.734.400 | 110,75s |

*Tabla 12. Resumen resultados para el dataset vm1084*

Gráficas vm1084

Ilustración 17. Coste Medio vm1084

Ilustración 18. Coste Mejor vm1084

Ilustración 19. Evaluaciones Media vm1084

Ilustración 20. Tiempo Medio vm1084

Resumen vm1748

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Coste Medio | Mejor | Semilla | Desviación Típica | #EV  Media | Tiempo Medio |
| Greedy | 408.101 | 408.101 | - | 0 | 1 | 0,7s |
| Búsqueda Aleatoria | 14.138.460,2 | 14.128.347 | 2 | 14.000,158 | 2.796.800 | 8.233,17s |
| BL El Mejor | 14.843.003,7 | 14.609.878 | 8 | 146.166,027 | 2.796.800 | 27,24s |
| BL El Primer Mejor | 7.944.729,3 | 7.553.040 | 8 | 183.018,885 | 2.796.800 | 38,6s |
| BL El Mejor Aleatorizado | 3.004.846,3 | 2.845.352 | 8 | 84.373,737 | 2.796.800 | 48,65s |
| Enfriamiento Simulado | 2.303.584,2 | 2.157.007 | 1 | 92.274,215 | 2.796.800 | 56,82s |
| Búsqueda Tabú | 3.299.200,6 | 3.149.172 | 1 | 96.162,588 | 2.796.800 | 212s |

*Tabla 13. Resumen resultados para el dataset vm1748*

Gráficas vm1748

Ilustración 21. Coste Medio vm1748

Ilustración 22. Coste Mejor vm1748

Ilustración 23. Evaluaciones Media vm1748

Ilustración 24. Tiempo medio vm1748

Análisis de los resultados obtenidos

Para realizar el análisis de los resultados obtenidos, vamos a realizar una comparación entre los algoritmos basándonos en la información obtenida de las tablas y de las gráficas.

La comparación de los algoritmos vamos a hacerla en base a los siguientes criterios:

* Eficiencia temporal: compararemos los algoritmos en relación del tiempo medio que tarda en arrojar una solución en función del tamaño del dataset.
* Calidad y robustez del algoritmo: en este aspecto nos basaremos en el resultado medio y la desviación típica.
* Número de evaluaciones, es decir, llamadas a la función de evaluación que en nuestro caso es *calculaCoste*().
* Mejor resultado individual obtenido, aquí revisaremos el mejor resultado obtenido de cada algoritmo.

Eficiencia temporal

La eficiencia de un algoritmo la basamos en el tiempo medio que tarda en dar una solución. Hemos creado una gráfica donde se puede ver el tiempo medio por cada algoritmo y por cada dataset.

Ilustración 25. Eficiencia temporal de los Algoritmos

En este caso se puede observar perfectamente que el orden de eficiencia es el siguiente:

1. Algoritmo Greedy
2. BL El Mejor
3. BL El Primer Mejor
4. BL El Mejor Aleatorizado
5. Enfriamiento Simulado
6. Búsqueda Tabú
7. Búsqueda Aleatoria

Era obvio que el Greedy iba a ser el mejor en este aspecto ya que este algoritmo no realiza un procedimiento iterativo de aproximación hacia una solución, simplemente en cada posición busca cuál es la ciudad más cercana sin volver a evaluar este camino nunca más.

En el caso de la búsqueda aleatoria, era de esperar que el tiempo iba a ser muchísimo mayor que el resto incluso se puede observar que los últimos datasets no se pueden ver en la gráfica, debido a que su función calculaCoste() no estaba optimizada debido a que en cada evaluación debía calcular completamente todo el coste de la solución. En las demás búsquedas, como sólo debíamos calcular el coste de un vecino es más eficiente y menos costos.

Si dejamos de evaluar el Algoritmo Greedy, las búsquedas locales son las más eficientes. Hay que aclarar que esto ocurre debido a que nuestros algoritmos están “capados” por un número de iteraciones, en un caso real sólo pararían si no encuentran un vecino mejor, es decir, no paran hasta llegar a un óptimo local, este aspecto lo estudiaremos en un apartado más adelante.

Entre los algoritmos de búsqueda local, observamos que el más eficiente es el de BL del Mejor ya que tienen mayor tendencia a estancarse en óptimos locales ya que en cada situación busca el menor vecino de todos.

Calidad y robustez

Para estudiar la calidad y robustez de los algoritmos, nos fijaremos en el coste medio y en la desviación típica

Ilustración 26. Coste Medio Normalizado de los Algoritmos

Ilustración 27. Desviación Típica Normalizada de los Algoritmos

Para que sea más representativo y podamos comparar todos los datasets se ha normalizado los valores entre 0 a 1 de la siguiente manera:

Gracias a esto, como hemos comentado antes hemos podido comparar el resultado de todos los algoritmos y todos los datasets en la misma escala.

Diremos que un algoritmo arroja una solución de calidad y robusta cuando esta solución tenga un cote medio bajo y además la desviación típica del coste también es baja.

Cuando observamos el algoritmo Greedy, observamos que siempre que se ejecuta devuelve la misma solución, por lo que el coste medio es el más bajo y la desviación típica es 0.

En el caso de la búsqueda aleatoria, podemos observar que nos devuelve los peores resultados en cuanto a coste medio de la solución, como era de esperar. No obstante podemos ver en la gráfica que es de los algoritmos que tiene menor desviación típica, esto ocurre porque es muy difícil que consiga dar con una solución buena y todas sus soluciones se mantiene dentro del mismo rango de malas soluciones donde no hay grandes diferencias.

En el caso de las búsquedas locales, podemos ver que la calidad de las soluciones va empeorando en función del tamaño del dataset, es decir, mientras vaya aumentando el tamaño del dataset, peor coste medio ira arrojando el algoritmo. Esto también se debe a que como los algoritmos están “capados” no terminan de llegar a una solución óptima o mejor en un tiempo razonable.

Dentro de estas búsquedas locales, el que obtiene mejor resultado medio es el de búsqueda local Aleatorizado, pero al tener una desviación típica tan alta no se puede considerar una solución de calidad.

En el Enfriamiento Simulado, al igual que en la búsqueda local Aleatorizado, podemos ver que el coste medio de los algoritmos es de los mejores, sólo por detrás del Greedy, pero vemos que la desviación típica no es constante y es grande, por lo que no se consideraría un resultado de calidad.

Por último observamos la búsqueda Tabú, en la cual da una solución de calidad y robustez al darnos unos resultamos medios bajos y constantes, aunque un poco peores que en el Enfriamiento Simulado, pero debido la desviación típica es mejor sin importar el tamaño del datasets.

Una vez observado los algoritmos, quedaría de la siguiente manera:

1. Algoritmo Greedy
2. Búsqueda Tabú
3. Enfriamiento Simulado
4. BL El Mejor Aleatorizado
5. BL El Primer Mejor
6. BL El Mejor
7. Búsqueda Aleatoria

Número de evaluaciones

En este apartado evaluaremos el número de evaluaciones, de tal manera que contamos las veces que se llama a la función de evaluación, en nuestro caso, calculaCoste().

Ilustración 28. . Número de evaluaciones Normalizadas de los Algoritmos

En este apartado también hemos normalizado de para poder tener todos los resultados de los datasets en la misma escala.

El orden de los mejores algoritmos en función de este aspecto sería:

1. Algoritmo Greedy
2. BL El Mejor Aleatorizado
3. BL El Primer Mejor
4. Búsqueda Aleatoria, BL El Mejor, Enfriamiento Simulado, Búsqueda Tabú

Podemos ver que el Greedy es el mejor ya que el algoritmo sólo realiza una evaluación en cada datasets que es cuando construye la solución que se va a devolver.

En el caso del BL El Mejor Aleatorizado, poder finalizar cuando en X iteraciones no encuentre a nadie, podemos ver que en datasets pequeños no realiza el máximo de iteraciones pero observamos que mientras va aumentando el tamaño del dataset, se va acercando al máximo de iteraciones.

Por último, en el caso del primer mejor ocurre algo parecido al Aleatorizado pero en menor escala debido a que puede salir si no mejora la solución, pero en datasets pequeños ocurre, en el momento que empieza a crecer es muy difícil que no encuentre nadie mejor por lo que llega al máximo.

Como se comentó anteriormente, el número de evaluaciones en el caso de la búsqueda local es engañoso ya que hemos controlado el límite máximo de iteraciones posibles, pero si dejáramos que el criterio de las búsquedas locales fuera el de convergencia, tanto el número de evaluaciones como la eficiencia sería de los peores algoritmos.

Mejor resultado individual obtenido

Por último, compararemos los algoritmos en función de los mejores resultados obtenidos. Esta gráfica también esta normalizada de 0-1 por el mismo motivo que en los apartados anteriores, para poder comparar los resultados en la misma escala.

*Ilustración 29. Coste Mejor Normalizado de los Algoritmos*

Ha dado la casualidad, que en todos los datasets si aplicamos el algoritmo Greedy partiendo desde la primera posición, este da las mejores soluciones en casi todos los datasets.

Cómo era de esperar, la búsqueda aleatoria es la que arroja los peores resultados de todos los algoritmos. Vemos que en todos los datasets, su mejor resultado es el peor de todos.

Cuando llegamos a las búsquedas locales, en el caso del mejor, aunque en datasets pequeños de resultados bastantes buenos, vemos que mientras aumenta el tamaño del datasets, el resultado empieza a empeorar, esto también puede ser debido el número de evaluaciones, que debería ser mayor para que nos dé una mejor solución. Lo mismo ocurre con el primer mejor.

En cambio, cuando observamos la búsqueda local del mejor aleatorizado, que para ser una búsqueda local, nos devuelve muy buenos resultados sin importar el tamaño del dataset, por lo que este algoritmo sería el mejor dentro de las búsquedas locales.

Cuando llegamos al enfriamiento simulado, observamos que los resultados son muy parecidos a los de la búsqueda tabú que son muy buenos e incluso en algunos casos mejorando al algoritmo Greedy. Pero hemos de mencionar que los parámetros utilizados en estos dos algoritmos son los parámetros por defecto en la práctica para que controlar el tiempo de estos algoritmos. En el siguiente apartado modificaremos estos algoritmos para ver si mejora o no los resultados.

Ajuste de Parámetros

Después de haber analizado todos los algoritmos y con la idea de hacer un estudio más completo, vamos a modificar algunos parámetros de los algoritmos de Enfriamiento Simulado y de Búsqueda Tabú, ya que estos tienen mucho margen de mejora.

Ajuste del Enfriamiento Simulado

En este algoritmo, tenemos varias posibilidades:

* Modificar µ o Φ. Estos parámetros son utilizados para calcular la temperatura inicial. Si modificamos estos parámetros haremos que el algoritmo sea más permisivo con soluciones peores que la actual.
* Modificar L o número de vecinos. Si se incrementa el valor del número de vecinos, aumentamos la exploración y con ello conseguir llegar a un mínimo.

Las pruebas que vamos a realizar serían dos:

1. Mismo L pero incrementando µ y Φ de 0,3 a 0,6.
2. Incrementar L de 20 a 50 e incrementar µ y Φ de 0,3 a 0,6

Obteniendo los siguientes resultados separados por datasets:

Mejora ES con dataset st70

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Coste Medio | Mejor  (675) | Semilla | Desviación Típica | #EV  Media | Tiempo Medio |
| Enfriamiento Simulado | 1.041,2 | 900 | 8 | 109,771 | 112.000 | 1,08s |
| ES Mejora 1 | 948 | 887 | 1 | 51,955 | 112.000 | 0.91s |
| ES Mejora 2 | 868,4 | 785 | 6 | 46,171 | 280.000 | 2,387s |

*Tabla 14. Mejora ES con dataset st70*

Mejora ES con dataset ch130

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Coste Medio | Mejor  (6.110) | Semilla | Desviación Típica | #EV  Media | Tiempo Medio |
| Enfriamiento Simulado | 10.882,5 | 9.567 | 8 | 931,091 | 208.000 | 2,00s |
| ES Mejora 1 | 9922,7 | 8.896 | 6 | 611,758 | 208.000 | 2,12s |
| ES Mejora 2 | 9492,8 | 9.194 | 2 | 288,656 | 520.000 | 5,53s |

*Tabla 15. Mejora ES con dataset ch130*

Mejora ES con dataset a280

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Coste Medio | Mejor  (2.579) | Semilla | Desviación Típica | #EV  Media | Tiempo Medio |
| Enfriamiento Simulado | 6.001,1 | 5.493 | 1 | 344,768 | 448.000 | 6,21s |
| ES Mejora 1 | 6.020,9 | 5.711 | 7 | 163,189 | 448.000 | 4,60s |
| ES Mejora 2 | 5.595,8 | 5.200 | 3 | 249,541 | 1.120.000 | 12,04s |

*Tabla 16. Mejora ES con dataset a280*

Mejora ES con dataset p654

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Coste Medio | Mejor  (34.643) | Semilla | Desviación Típica | #EV  Media | Tiempo Medio |
| Enfriamiento Simulado | 327.771 | 301.305 | 0 | 22.202,27 | 1.046.400 | 17,50s |
| ES Mejora 1 | 308.983,5 | 263.555 | 6 | 21.919,75 | 1.046.400 | 15,58s |
| ES Mejora 2 | 280.889 | 236.175 | 6 | 21.547,99 | 2.616.000 | 40,80s |

*Tabla 17. Mejora ES con dataset p654*

Mejora ES con dataset vm1084

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Coste Medio | Mejor  (239.297) | Semilla | Desviación Típica | #EV  Media | Tiempo Medio |
| Enfriamiento Simulado | 1.293.567,5 | 1.233.964 | 3 | 50.243,484 | 1.734.400 | 32,37s |
| ES Mejora 1 | 1.241.259,6 | 1.177.058 | 0 | 42.105,569 | 1.734.400 | 29,29s |
| ES Mejora 2 | 1.078.739,6 | 1.017.962 | 7 | 44.801,72 | 4.336.000 | 75,74s |

*Tabla 18. Mejora ES con dataset vm1084*

Mejora ES con dataset vm1784

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Coste Medio | Mejor | Semilla | Desviación Típica | #EV  Media | Tiempo Medio |
| Enfriamiento Simulado | 2.303.584,2 | 2.157.007 | 1 | 92.274,215 | 2.796.800 | 56,82s |
| ES Mejora 1 | 2.226.744,6 | 2.122.169 | 0 | 67.242,248 | 2.796.800 | 48,32s |
| ES Mejora 2 | 1.958.859 | 1.823.355 | 1 | 81.504,814 | 6.992.000 | 160,71s |

*Tabla 19. Mejora ES con dataset vm1784*

Ilustración 30. Coste Medio Normalizado ES Mejorado

Ilustración 31. Desviación Típica Normalizada ES Mejorado

Ilustración 32. Coste Mejor Normalizado ES Mejorado

Observando los resultados podemos observar aumentando la vecindad, es decir, explorando más se obtienen resultados mucho mejores, en nuestro caso, lo hemos realizado con valores fijos, pero la idea ideal sería ajustar ese parámetro en función del tamaño del dataset ( *n/2*, *n/4*).

Al modificar los valores µ o Φ por lo general mejoramos los resultados ya que acepta valores peores para después intensificarlos que nos puede llevar a otro mínimo o no.

Ajuste del Búsqueda Tabú

En la búsqueda tabú tenemos los siguientes parámetros que podemos modificar:

* Número de iteraciones del algoritmo
* Número de reinicializaciones: este parámetro implica el número de veces que el algoritmo va a reinicializar la solución actual, esta nueva solución actual podría ser:
  + Una completamente aleatoria, lo cual implica diversificar por otra zona donde nunca hayamos explorado.
  + La mejor solución hasta el momento, que implicaría intensificar más dónde creemos que esta el mínimo.
  + Greedy de la memoria de frecuencias: esta solución también implicaría diversificar pero justamente en espacios donde nunca hayamos estado, la solución aleatoria puede dar el caso de que caiga en una zona ya explorada, pero este caso es justo irnos a la zona que menos se ha explorado.
* Por último, podemos modificar el número de vecinos que analizamos en cada solución.

Las mejoras que vamos a reinicializar son las siguientes:

1. Incrementar el número de iteraciones de 40 a 60 manteniendo los demás valores por defecto.
2. Incrementar el número de reinicializaciones de 4 a 150 manteniendo los demás valores por defecto.
3. Incrementar el número de vecinos de 40 a 70 manteniendo los demás valores por defecto.
4. Ejecutar el algoritmo con los siguientes parámetros:
   1. Número de iteraciones: 60
   2. Número de reinicios: 10
   3. Número de vecinos: 70

Mejora BT con dataset st70

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Coste Medio | Mejor  (675) | Semilla | Desviación Típica | #EV  Media | Tiempo Medio |
| Búsqueda Tabú | 981,1 | 923 | 3 | 40,322 | 112.000 | 1,22s |
| BT Mejora 1 | 963,9 | 907 | 4 | 39,085 | 168.000 | 1,74s |
| BT Mejora 2 | 1.079,4 | 982 | 5 | 105,35 | 112.000 | 1s |
| BT Mejora 3 | 911,4 | 853 | 0 | 31,74 | 196.000 | 1,93s |
| BT Mejora 4 | 884,8 | 787 | 8 | 56,07 | 294.000 | 2,97s |

*Tabla 20. Mejora BT con dataset st70*

Mejora BT con dataset ch130

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Coste Medio | Mejor  (6.110) | Semilla | Desviación Típica | #EV  Media | Tiempo Medio |
| Búsqueda Tabú | 12.163,1 | 11.610 | 5 | 270,13 | 208.000 | 2,92s |
| BT Mejora 1 | 11.406,5 | 10.814 | 0 | 345,87 | 312.000 | 3,88s |
| BT Mejora 2 | 12.726,4 | 12.232 | 2 | 248,70 | 208.000 | 2,32s |
| BT Mejora 3 | 10.155,9 | 9.417 | 5 | 628,44 | 364.000 | 4,19s |
| BT Mejora 4 | 10.268 | 9.259 | 0 | 525,16 | 546.000 | 6,09s |

*Tabla 21. Mejora BT con dataset ch130*

Mejora BT con dataset a280

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Coste Medio | Mejor  (2.579) | Semilla | Desviación Típica | #EV  Media | Tiempo Medio |
| Búsqueda Tabú | 3.317,2 | 2.976 | 1 | 244,456 | 448.000 | 9,26s |
| BT Mejora 1 | 3.537,3 | 3.041 | 1 | 467,6 | 672.000 | 13,24s |
| BT Mejora 2 | 2.870,5 | 2.808 | 7 | 64,90 | 448.000 | 6,24s |
| BT Mejora 3 | 3.304,1 | 3.085 | 2 | 255,22 | 784.000 | 13,15s |
| BT Mejora 4 | 3.123,2 | 2.875 | 0 | 240,27 | 1.176.000 | 22,33s |

*Tabla 22. Mejora BT con dataset a280*

Mejora BT con dataset p654

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Coste Medio | Mejor  (34.643) | Semilla | Desviación Típica | #EV  Media | Tiempo Medio |
| Búsqueda Tabú | 111.932,5 | 108.059 | 2 | 4.977,913 | 1.046.400 | 42,24s |
| BT Mejora 1 | 127.576,8 | 108.115 | 2 | 46,407 | 1.569.600 | 65,64s |
| BT Mejora 2 | 107.758,7 | 107.523 | 0 | 113,38 | 1.046.400 | 30,71s |
| BT Mejora 3 | 128.245,5 | 100.171 | 0 | 63.555,74 | 1.831.200 | 73,24s |
| BT Mejora 4 | 109.759,1 | 102.888 | 1 | 3.526,82 | 2.746.800 | 105,4s |

*Tabla 23. Mejora BT con dataset p654*

Mejora BT con dataset vm1084

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Coste Medio | Mejor  (239.297) | Semilla | Desviación Típica | #EV  Media | Tiempo Medio |
| Búsqueda Tabú | 1.842.049,6 | 1.755.767 | 4 | 61.932,684 | 1.734.400 | 110,75s |
| BT Mejora 1 | 1.757.517,8 | 1.719.557 | 8 | 28.175,51 | 2.601.600 | 205,49s |
| BT Mejora 2 | 1.931.999,3 | 1.870.560 | 1 | 37.370,74 | 1.734.400 | 72,55s |
| BT Mejora 3 | 1.466.973,3 | 1.336.343 | 5 | 104.056,45 | 3.035.200 | 249,24s |
| BT Mejora 4 | 1.405.757,1 | 1.323.983 | 5 | 48.177,94 | 4.552.800 | 328,83s |

*Tabla 24. Mejora BT con dataset vm1084*

Mejora BT con dataset vm1784

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Coste Medio | Mejor | Semilla | Desviación Típica | #EV  Media | Tiempo Medio |
| Búsqueda Tabú | 3.299.200,6 | 3.149.172 | 1 | 96.162,588 | 2.796.800 | 212s |
| BT Mejora 1 | 3.204.416,6 | 3.059.339 | 2 | 100.277,39 | 4.195.200 | 490,26s |
| BT Mejora 2 | 3.475.641,4 | 3.350.393 | 3 | 92.812,59 | 2.796.800 | 119,18s |
| BT Mejora 3 | 2.607.726,3 | 2.440.445 | 1 | 126.664,95 | 4.894.400 | 342,07s |
| BT Mejora 4 | 2.576.108,2 | 2.396.420 | 1 | 144.706,91 | 7.341.600 | 448,41s |

*Tabla 25. Mejora BT con dataset vm1784*

Ilustración 33. Desviación Típica Normalizada BT Mejorado

Ilustración 34. Coste Medio Normalizado BT Mejorado

Una vez hemos experimentado variando los parámetros del algoritmo, podemos ver que modificando tanto el número de iteraciones o el número de reinicializaciones independientemente, es decir, sin modificar ningún parámetro más, los valores obtenidos son incluso peores que teniendo los valores por defecto.

Ilustración 35. Coste Mejor Normalizado BT Mejorado

En cambio, si vemos que sólo variamos el número de vecinos que evaluamos conseguimos un coste medio de más robusto ya que conseguimos un mejor resultado como media, aunque la desviación típica aumenta.

Por último, si aumentamos todos los parámetros obtenemos los mejores resultados y mejores resultados medios, aunque la desviación típica aumenta. La idea ideal sería aumentar los parámetros de manera proporcional al tamaño del dataset.