

# Práctica 1 de Inteligencia Artificial: Búsqueda Local

Roberto Amat Alins  
Marc Camarillas Parés  
Alejandro Salvat Navarro  
**Grupo 13**

# Índice

## **1. Descripción del problema y justificaciones de su solución**

## **2. Experimentos**

**Experimento 1: Determinar qué conjunto de operadores es mejor.**

**Experimento 2: Determinar la estrategia de la solución inicial**

**Experimento 3: Determinar los mejores parámetros para el Simulated Annealing**

**Experimento 4: Evolución del tiempo de ejecución**

**Experimento 6: Efecto de aumentar el coste por km**

**Experimento 7: Modificación horas de trabajo**

**Experimento 8: Experimento Especial**

## **3. Conclusión**

## **4. Introducción al trabajo de innovación**

## 1.Descripción del problema y justificaciones

### Descripción y justificación de la implementación del estado

El estado tiene presente los siguientes atributos:

- Ruta: vector de enteros indican el orden en el que el camión debe visitar las gasolineras. Usamos un -1 para representar una vuelta al centro.
- Distancia Recorrida Total: entero que representa la suma de las distancias que han recorrido los camiones.
- Porcentaje Ganancias: entero que representa la suma de los porcentajes que se obtienen al entregar las peticiones.
- Pérdidas: entero que representa la diferencia entre el porcentaje máximo de ganancias y el porcentaje de ganancias.
- Ganancias:  $1000 * (\text{Porcentaje Ganancias})/100 - 2 * \text{Distancia}$ .

Usamos esta solución debido a que estos elementos son aquellos variables que influyen en nuestro objetivo para la solución final. La ruta de cada camión influye en la distancia recorrida total ( elemento que queremos minimizar) y en el porcentaje de ganancias, influyendo este último en las pérdidas (elemento a minimizar) y en las ganancias (elemento a maximizar)

### Descripción y justificación de los operadores utilizados

- Intercambio(g1, g2, c1, c2): intercambia las gasolinera g1 que pertenece al camión c1 con la gasolinera g2 que pertenece al camión c2, siempre y cuando se respeten las restricciones de distancia máxima para un camión.
- intercambioGasolineraFueraDeRuta(g1, g2, c1): intercambia la gasolinera g1 que pertenece al camión g1 con la gasolinera g2 que no pertenece a ningún camión, siempre que se respeten las restricciones de distancia máxima para un camión.

Hemos escogido estos operadores porque con ellos somos capaces de recorrer todo el espacio de soluciones, es decir todas las gasolineras pueden estar asignadas a todos los centros de distribución. Principalmente elegimos hacer el operador insertar paquete, pero este nos generaba no-soluciones.

### **Descripción y justificación de las estrategias para hallar la solución inicial**

- SoluciónInicial1: Asignar la gasolinera al camión que tenga más cerca. Empezar con las que tienen más de dos peticiones y luego hacer lo mismo con las que tienen una sola petición. Si el número de tanques del camión es 0 vuelva al centro a suministrar, sinó sigue en la ruta.
- SoluciónInicial2: Asignar la gasolinera a un camión random Empezar con las que tienen más de dos peticiones y luego hacer lo mismo con las que tienen una sola petición. Si el número de tanques del camión es 0 vuelva al centro a suministrar, sinó sigue en la ruta.

Hemos decidido implementar estas soluciones porque inicialmente nos generan una ruta completa, es decir, cada camión inicialmente entrega 2 tanques antes de volver al centro y recorre 5 viajes que es el máximo de viajes que puede hacer en un día. Además hemos hecho la distinción de que la segunda te genere la solución de manera aleatoria por qué puede que de este modo, al ser peor solución, al usar el hill climbing no tenga el problema de encontrarse en un óptimo local.

### **Descripción y justificación de las funciones heurísticas**

- HeuristicFuncion: el valor del heurístico, viene determinado por la diferencia entre la distancia total recorrida y las pérdidas, ambas multiplicadas por un factor de ponderación.

Hemos decidido usar este heurístico debido a que queremos minimizar la distancia pero a su vez queremos aumentar las pérdidas ya que así el día siguiente estas no son mayores.

## 2. Experimentos

### Experimento 1

Determinar qué conjunto de operadores da mejores resultados para una función heurística que optimice el criterio de calidad del problema con un escenario en el que el número de centros de distribución es 10, hay un camión en cada centro y el número de gasolineras es 100. Deberéis usar el algoritmo de Hill Climbing.

Escoged una de las estrategias de inicialización de entre las que proponéis. A partir de estos resultados deberéis fijar los operadores para el resto de experimentos.

#### Observación:

Puede haber unos operadores que funcionen mejor con el heurístico que otros.

#### Planteamiento:

Probamos el programa con distintos conjuntos de operadores y analizamos los resultados.

#### Hipótesis:

- **[H0]** Hay algunos operadores mejores que otros.
- **[H1]** Todos los operadores son igual de eficientes.

#### Método:

- Utilizamos 10 seeds diferentes.
- Realizamos una ejecución para cada par de seeds y operadores.
- Realizamos el experimento con el Algoritmo Hill Climbing.
- Realizamos el experimento con Número de Gasolineras = 100 y Número de Centros de Distribución = 10.

Conjunto OP 1: [intercambio]

Conjunto OP 2: [intercambio, intercambioGasolineraFueraDeRuta]

SEED	BENEFICIOS		TIEMPO EJECUCIÓN (MS)	
	CONJUNTO 2	CONJUNTO 1	CONJUNTO 2	CONJUNTO 1
6666	90356	90356	1338	417
9999	92390	91388	1331	309
9876	93636	93636	1548	458
1234	94804	93772	830	240
4774	91764	91764	981	368
5983	94036	92868	1146	349
5729	93618	93618	1181	367
7777	91352	90396	1426	498
8007	90892	89788	1057	306
DESVIACIÓN	1551,16	1571,92	228,26	80,21

### Conclusión:

Tras realizar las pruebas, nos damos cuenta que la variación en los beneficios al pasar del conjunto 2 al 1 es mínima, mientras que en cambio se reduce el tiempo de ejecución significativamente, con lo cual nos quedamos con la opción del conjunto 1.

*\* Al implementar el operador 2 hemos tenido problemas que han resultado en un mal funcionamiento del mismo. Debido a eso, aunque no exploraremos todo el espacio de soluciones, solo usaremos el operador 1 para que nuestros experimentos sean correctos.*

## Experimento 2

Determinar qué estrategia de generación de la solución inicial da mejores resultados para la función heurística usada en el apartado anterior, con el escenario del apartado anterior y usando el algoritmo de Hill Climbing. A partir de estos resultados deberéis fijar también la estrategia de generación de la solución inicial para el resto de experimentos.

### Observación:

Una de las estrategias implementadas puede ser mejor que la otra.

### Planteamiento:

Ejecutamos el programa con las dos soluciones iniciales y analizamos los resultados.

### Hipótesis:

- **[H0]** Ambas soluciones iniciales muestran el mismo rendimiento.
- **[H1]** Una solución inicial muestra mejor rendimiento que la otra.

### Método:

- Utilizamos 10 seeds diferentes.
- Realizamos una ejecución para cada par de seeds y soluciones iniciales.
- Realizamos el experimento con el Algoritmo Hill Climbing.
- Realizamos el experimento con Número de Gasolineras = 100 y Número de Centros de Distribución = 10.

SEED	BENEFICIO INICIAL		BENEFICIO FINAL		TIEMPO DE EJECUCIÓN (MS)	
	SOL INI 1	SOL INI 2	SOL INI 1	SOL INI 2	SOL INI 1	SOL INI 2
6666	88762	75458	90356	82258	1200	1301
9999	90632	78688	92390	86746	934	1366
9876	92224	79860	93636	87094	1262	1175
1234	93108	79080	94804	86004	649	1160
4774	90540	74444	91764	81360	699	1161
5983	91928	77984	94036	85398	966	1295
5729	92276	79458	93618	86610	977	1382
7777	88966	75986	91352	84858	1025	1492
8007	88888	79090	90892	85224	845	1146
DESVIACIÓN	1661,04 1072	1973,35 9853	1551,15988 9	1999,891247	203,510305 5	122,859676 1

### **Conclusión:**

La solución inicial 2 nos proporciona una solución en la que el beneficio inicial es menor, lo que nos supondrá mayores beneficios al día siguiente. Por esta razón, nos quedaremos con la solución inicial 2 para el resto de experimentos de nuestro proyecto

### **Experimento 3**

**Determinar los parámetros que dan mejor resultado para Simulated Annealing con el mismo escenario, usando la misma función heurística, los operadores y la estrategia de generación de la solución inicial escogidos en los experimentos anteriores.**

#### **Observación:**

Ha de existir una combinación de los parámetros de utiliza el Simulated Annealing que sea mejor que las demás.

#### **Planteamiento:**

Vamos realizando pruebas iterativamente con diversas combinaciones de parámetros para ir acotandolos en función de su resultado.

#### **Hipótesis:**

- **[H0]** Encontramos una combinación de los parámetros de entrada del Simulated Annealing que nos proporciona unos resultados iguales que los que nos proporciona el Hill Climbing.
- **[H1]** Encontramos una combinación de los parámetros de entrada del Simulated Annealing que nos proporciona unos resultados mejores que los que nos proporciona el Hill Climbing.

#### **Método:**

Valores probados:

$K \rightarrow 1, 5, 10, 25, 50, 125, 250$

$\text{Lambda} \rightarrow 1.0, 0.1, 0.01, 0.001, 0.0001$

Núm Iteraciones  $\rightarrow$  Escogemos un número relativamente grande de iteraciones para comprobar cómo afecta la decisión de escoger el resto de variables.



K	Lambda	Beneficios AVG
1	1	87416
1	0,1	88128
1	0,01	88112
1	0,001	88114
1	0,0001	88142
5	1	87480
5	0,1	88056
5	0,01	88228
5	0,001	88166
5	0,0001	88056
25	1	87768
25	0,1	88094
25	0,01	88296
25	0,001	88084
25	0,0001	88022
125	1	87108
125	0,1	87996
125	0,01	88174
125	0,001	87644
125	0,0001	87108

### Conclusión:

Tras realizar las pruebas observamos que la combinación de  $k = 125$  y  $\lambda = 0,0001$  nos otorga los mejores resultados. Siendo que Beneficios AVG sale la media de la respuesta obtenida al llamar al programa con los mismos parámetros y 10 seeds diferentes.

## Experimento 4

Dado el escenario de los apartados anteriores, estudiar cómo evoluciona el tiempo de ejecución para hallar la solución para valores crecientes de los parámetros del problema siguiendo la proporción 10:100 para centros y gasolineras. Comenzad con 10 centros de distribución e incrementar el número de 10 en 10 hasta que se vea la tendencia. Usad el algoritmo de Hill Climbing y Simulated Annealing y la misma función heurística que antes. Comprobad que los parámetros para Simulated Annealing que habéis hallado en el apartado anterior siguen dando buenos resultados al aumentar el tamaño del problema.

### Observación:

El tiempo de ejecución será mayor cuantos más centros tengamos, ya que nuestro algoritmo se ve forzado a recorrerlos todos, haciendo que el coste temporal crezca

### Planteamiento:

Generamos diversas soluciones, una para cada número de centros y observamos los resultados

### Hipótesis:

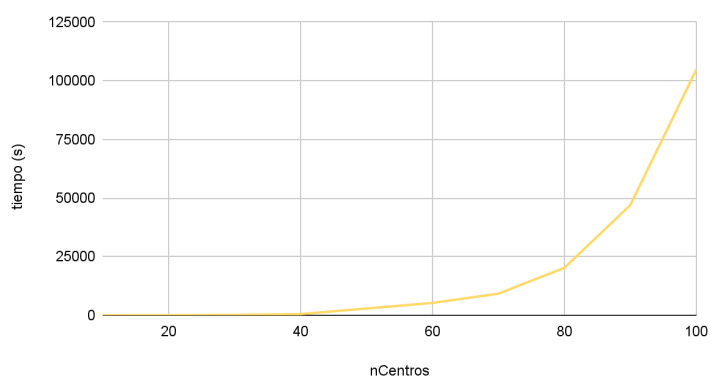
- [H0] El tiempo de ejecución se mantiene constante independientemente del número de centros
- [H1] Según crezca el número de centros, mayor será el tiempo de ejecución

### Método:

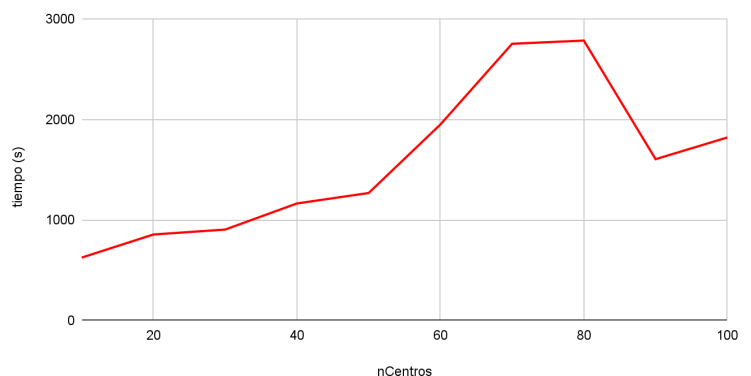
Asignaremos el mismo valor que en los apartados anteriores a todas las variables excepto al número de centros. Ejecutaremos el código sumando 10 centros en cada ejecución y manteniendo la proporción de 100 gasolineras por cada 10 centros, repitiendo esto un total de 10 veces.

**Conclusión:** Podemos ver que el coste de ejecutar el hill climbing es exponencial cuando aumentamos el tamaño de la entrada, en cambio con el simulated annealing podemos ver que prácticamente no varía el tiempo de ejecución, debido a que coje valores estocásticos. Por lo tanto cuando la entrada aumente preferiremos usar simulated annealing.

Evolución de tiempo en Función de nCentros con Hill Climbing



Evolución de tiempo en Función de nCentros con Simulated Annealing



nCentros	TIEMPO DE EJECUCIÓN (ms)	
	HillClimbing	Simulated Annealing
10	1501	625
20	4026	855
30	15308	904
40	51080	1164
50	288707	1268
60	527048	1950
70	922285	2754
80	527048	2786
90	4702278	1605
100	10478610	1821

## Experimento 6

Hemos asumido un coste fijo de 2 por kilómetro recorrido. Usad el algoritmo de Hill Climbing para estudiar cómo afecta al número de peticiones servidas el aumentar el coste por kilómetro usando el escenario del primer apartado. Id doblando el coste hasta que podáis ver la tendencia. ¿Tiene algún efecto en la proporción de peticiones servidas según el tiempo que llevan pendientes?

### Observación:

Puede que el precio por kilómetro afecta en la proporción de peticiones servidas

### Planteamiento:

Ejecutamos el programa doblando el coste por kilómetro a cada ejecución y analizamos los resultados.

### Hipótesis:

- **[H0]** El precio por kilómetro no afecta en la proporción de peticiones servidas.
- **[H1]** El precio por kilómetro afecta en la proporción de peticiones servidas.

### Conclusión:

El coste de los kilómetros recorridos no afecta al número de peticiones totales realizadas, ya que no tenemos un operador que elimine peticiones de la ruta de los camiones.

## Experimento 7

Manteniendo el límite de 5 viajes diarios por cisterna, comprobad qué efecto tendría el aumentar y disminuir en una hora el horario de trabajo de las cisternas con el escenario del primer apartado, es decir, aumentando/disminuyendo los kilómetros que podemos recorrer en un día. Usad el algoritmo de Hill Climbing para comprobar el efecto sobre el beneficio.

### Observación:

Puede que al aumentar o disminuir el número de horas diarias en 1 cambie el beneficio.

### Planteamiento:

Ejecutar el programa disminuyendo en 80 el número de kilómetros máximos y lo volvemos a ejecutar sumándole 80 número de kilómetros máximos inicial y analizar los resultados.

### Hipótesis:

- **[H0]** Hay algunos operadores mejores que otros.
- **[H1]** Todos los operadores son igual de eficientes.

### Método:

- Utilizamos 10 seeds diferentes.
- Realizamos una ejecución para cada par de seeds y operadores.
- Realizamos el experimento con el Algoritmo Hill Climbing.
- Realizamos el experimento con Número de Gasolineras = 100 y Número de Centros de Distribución = 10.

SEED	BENEFICIOS		
	7h trabajo/día	8h trabajo/día	9h trabajo/día
6666	89604	90356	90356
9999	91388	91388	91388
9876	93636	93636	93636
1234	93772	93772	93772
4774	91764	91764	91764
5983	90824	92868	92868
5729	93618	93618	93618
7777	90396	90396	90396
8007	87852	89788	89788
DESVIACIÓN	2025,61	1571,92	1571,92

**Conclusión:** Podemos ver que al aumentar el número de horas de los trabajadores no influye debido a que inicialmente hacen 5 viajes con 2 peticiones cada viaje. En cambio al disminuir la jornada en una hora puede que en algún punto superen la distancia máxima que pueden recorrer en un día y por lo tanto no entreguen tantos pedidos, por eso el número de pérdidas es menor.

## Experimento 8

**Experimento Especial:** Para fomentar el trabajo continuado en la practica siguiendo la planificación, asignaremos un punto extra sobre la nota de la práctica a los grupos que envíen durante la semana del 18 de octubre un correo con el resultado del beneficio que se obtiene en el escenario con 10 centros de distribución, un camión por centro y 100 gasolineras y cuánto tiempo se tarda en hallar la solución en milisegundos (aproximadamente). Para que todos los experimentos sean con las mismas condiciones usaremos como semilla del generador de números aleatorios 1234, para centros y gasolineras. Deberéis usar los operadores, inicialización y heurística escogidos con los experimentos 1 y 2.

Con solución inicial 2 y el conjunto de operadores 1.

**Beneficio:** 83034€

**Tiempo:** 404ms

## 3.Conclusión

El trabajo lo realizamos sin fijarnos bien del todo en los experimentos, centrándonos más en el enunciado inicial, lo que ha hecho que luego algún experimento haya dado problemas para poder realizarse correctamente. Por otro lado, la programación de un operador nos dió problemas, lo que retrasó bastante la práctica e hizo que no la pudiéramos presentar de la manera que teníamos prevista, teniendo esta algunos pequeños fallos aun con un buen planteamiento y resolución.

## 4. Introducción al trabajo de innovación

Para el trabajo de innovación hemos elegido un proyecto que intenta unir una tecnología emergente como es el mundo blockchain con el de la Inteligencia Artificial. SingularityNET busca crear una red descentralizada de servicios de IA basada en Ethereum y sus Smart Contracts.