



---

# *PRACTICA 3*

---

Jesús Campos Márquez



2018/2019

MODELOS DE BUSQUEDA Y HEURISTICAS DE BUSQUEDA  
Universidad de Huelva



---

# Índice

---

## **1. Definición teórica de los algoritmos**

- 1.1. Sistema basado en Hormigas (SH)**
- 1.2. Sistema basado en Hormigas Elitistas (SHE)**
- 1.3. Sistema basado en Colonia de Hormigas (SCH)**

## **2. Resultados de los algoritmos**

- 2.1. Sistema basado en Hormigas (SH)**
- 2.2. Sistema basado en Hormigas Elitistas (SHE)**
- 2.3. Sistema basado en Colonia de Hormigas (SCH)**

## **3. Resultados Globales**

## **4. Conclusiones**

## Definición teórica de los algoritmos

- Sistema basado en Hormigas (SH)

Es un método que se basa en la construcción de un camino gracias a un mecanismo probabilístico condicionado por feromonas y distancia de caminos, que se asimila al comportamiento de las hormigas reales. (Figura 1.1)

El sistema aplicado al TSP, en cada paso estando en una ciudad  $p_1$ , elige a una ciudad  $p_2$  entre todas las posibles sin visitar. Tras esta elección se transita y se da lugar a una evaporación de las feromonas y a un aporte de feromonas a aquellas aristas que se han visitado.

- Sistema basado en Hormigas Elitistas (SHE)

Sigue la heurística del sistema de hormigas normal, pero en su regla de actualización utiliza además de la actualización del sistema anterior, un refuerzo adicional a los buenos arcos, es decir, a los que están en la mejor solución global, con lo que se consigue una convergencia hacia una buena solución más rápida. ( pero cayendo en mas óptimos locales)

- Sistema basado en Colonia de Hormigas (SCH)

Su funcionamiento es basado en el SH pero de forma mejorada. La primera mejora que incluye es un mecanismo para hacer una exploración mas amplia a nuevos arcos, para ello se hace una actualización local, para que los arcos visitados sean menos posibles de coger y poder diversificar y, además, se hace una actualización global para generar una solución de calidad.

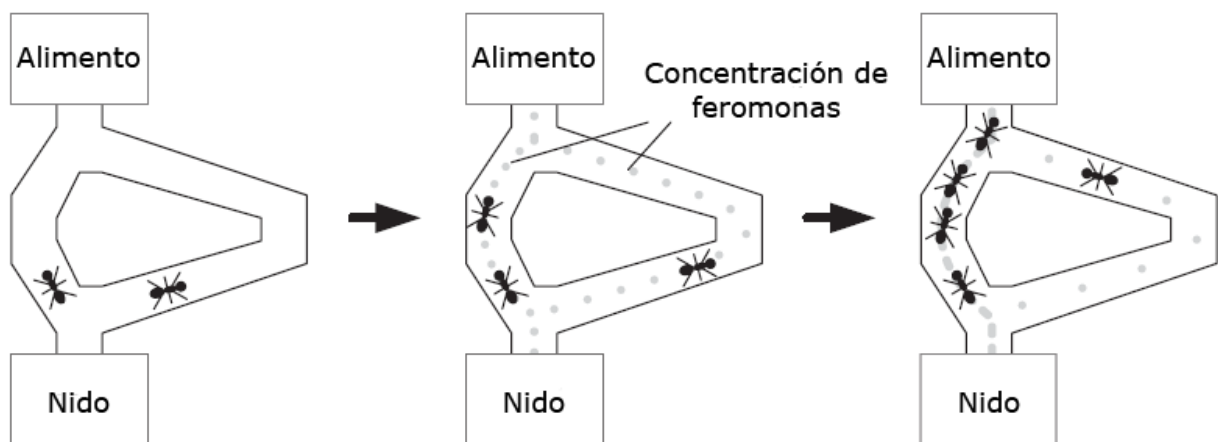


Figura 1.1

## Resultados de los algoritmos

Se mostrarán tablas gráficas del estudio realizada para cada algoritmo:

- Sistema basado en Hormigas (SH)

#Ejecución y Seed	St70		Ch130		A280		p654		Vm1084	
	Coste	#Ev	Coste	#Ev	Coste	#Ev	Coste	#Ev	Coste	#Ev
1	722	330160	6707	153300	3219	34900	53100	5020	382126	1540
2	716	318340	6783	141940	3146	35480	54311	5020	379835	1560
3	721	330520	6691	147060	3178	33780	54479	4620	375955	1580
4	713	301200	6845	150140	3244	32920	52130	3880	391050	1520
5	723	321860	6749	144960	3240	32940	54612	3820	381764	1520
Media	719	320416	6755	147480	3205,4	34004	53726,4	4472	382146	1544
Desviación Típica	3,84707681	10702,2046	55,35341	3953,89428	37,818514	1033,31699	962,634011	528,787292	4961,788831	23,3238076

Del sistema implementado se puede ver a simple vista que los resultados son bastante buenos, sin embargo, tiene que hacer un gran número de evaluaciones para conseguirlo, esto es por que la convergencia ha estado limitada por tiempo, en el que para el st70.tsp se le ha puesto 5 minutos, aumentando a 5 minutos mas cada fichero respecto el anterior.

- Sistema basado en Hormigas Elitista (SHE)

#Ejecución y Seed	St70		Ch130		A280		p654		Vm1084	
	Coste	#Ev	Coste	#Ev	Coste	#Ev	Coste	#Ev	Coste	#Ev
1	723	362000	6658	157340	3214	30580	55969	4020	391393	1480
2	716	351780	6720	154620	3192	30480	53472	4020	394205	1880
3	722	366880	6657	157860	3336	30660	54606	3860	371310	1420
4	720	367300	6806	156120	3263	30680	55176	4020	390715	1660
5	720	351300	6833	157460	3221	30680	55484	4020	380184	2120
Media	720,2	359852	6734,8	156680	3245,2	30616	54941,4	3988	385561,4	1712
Desviación Típica	2,4	7039,5068	73,3250298	1182,67493	50,8936145	77,3563184	857,559701	64	8570,855327	259,414726

Da resultados muy buenos, mejorando en algunos casos al sistema de hormigas tradicional, pero en otros casos empeorándolo, la diferencia radica en que este sistema alcanza más rápidamente una buena solución, pero al mejorarse los buenos arcos no se capaz de diversificar por lo que es muy propenso a caer en óptimos locales, y solo debería de utilizarse para ver que se alcanza más rápidamente una buena solución, en el caso de no estar limitado por tiempo si no por evaluaciones.

## MB Y HB

- Sistema basado en Colonia de Hormigas (SCH)

#Ejecución y Seed	St70		Ch130		A280		p654		Vm1084	
	Coste	#Ev	Coste	#Ev	Coste	#Ev	Coste	#Ev	Coste	#Ev
1	676	651480	6309	308200	3050	57260	42745	7040	308332	2540
2	700	697920	6251	308140	2931	57280	40771	7180	312175	2620
3	682	653640	6779	325380	2976	63040	41153	7240	290897	2460
4	678	653580	6222	324660	3036	62260	38213	7520	293594	2440
5	682	653880	6216	310540	2919	60300	38199	7680	298831	2440
Media	683,6	662100	6355,4	315384	2982,4	60028	40216,2	7332	300765,8	2500
Desviación Típica	8,5229103	17930,9074	214,344209	7918,50895	53,1887206	2422,44835	1769,86625	233,786227	8246,073087	70,4272674

Sus resultados son extraordinariamente buenos, con un número de evaluaciones mayor a los anteriores sistemas. La diferencia entonces, es que hace una actualización local de la feromona permitiendo a las demás hormigas poder buscar por otros lados, diversificando la transición a nodos no visitados, y con ello a la obtención de buenos resultado cayendo menos en óptimos locales, y además, su cálculo de la transición a otro nodo es más rápido por su nueva heurística implementada, y por ello consigue evaluar más.

## Resultados globales y comparacion entre algoritmos

Modelo	St70 (675)				Ch130 (6110)				A280 (2579)			
	Media	Mejor	Desviación Típica	Ev-Media	Media	Mejor	Desviación Típica	Ev-Media	Media	Mejor	Desviación Típica	Ev-Media
<b>Greedy</b>	847,1	815	473,4996938	1	7752,2	7358	473,4996938	1	3128,9	2975	94,44622809	1
SH	719	713	3,847076812	320416	6755	6691	55,35341001	147480	3205,4	3146	37,81851398	34004
SHE	720,2	716	2,4	359852	6734,8	6657	73,32502983	156680	3245,2	3192	50,89361453	30616
SCH	683,6	676	8,522910301	662100	6355,4	6216	214,3442092	315384	2982,4	2919	53,18872061	60028

Modelo	P654 (34.643)				Vm1084 (239.297)			
	Media	Mejor	Desviación Típica	Ev-Media	Media	Mejor	Desviación Típica	Ev-Media
<b>Greedy</b>	45616,6	43210	1570,208088	1	426730,4	413657	8117,460862	1
SH	53726,4	52130	962,6340114	4472	382146	375955	4961,788831	1544
SHE	54941,4	53472	857,5597005	3988	385561,4	371310	8570,855327	1712
SCH	40216,2	38199	1769,866255	7332	300765,8	290897	8246,073087	2500

Las tablas anteriores muestran los resultados globales, con los mejores resultados tras 5 ejecuciones de cada algoritmo y con cada fichero, para poder hacer una buena comparación se utilizarán resultados del algoritmo Greedy de la práctica 1, y que se ha utilizado en esta práctica como uno de los parámetros para inicializar la feromona.

Los resultados muestran un claro ‘Vencedor’ el cuál es el Sistema basado en Colonias de Hormigas (SCH), y es gracias a su capacidad de diversificar y buscar por todo el problema de forma más eficiente, y además gracias a que su convergencia a estado limitada por tiempo, con 5 minutos para el fichero más pequeño y sumándole 5 a cada fichero siguiente.

Debido a lo anterior, el algoritmo es capaz de mejorar al que en la práctica 2 se consideraba como el mejor algoritmo que se había implementado (GRASP Extendido) en prácticas hasta la fecha.

Primero de todo, y viendo sus resultados, no podemos considerar que el algoritmo SH sea malo debido a que la solución dada ya es mejor que todos los algoritmos anteriores implementados, y que puede ser mejorada dado un tiempo de convergencia mayor.

Por otra parte, el algoritmo SHE tampoco es malo, ya que mejora también a todos los anteriores, aunque está por debajo de los otros dos implementados en esta práctica, debido a que su actualización de feromonas lo que hace es mejorar y darle más importante a los arcos buenos, por lo que la siguiente hormiga tiende a pasar por ahí más que por otra, esto desde el punto de vista biológico es lo normal, pero algorítmicamente hace que no se diversifique, por lo que capta la solución a los mejores nodos, como ocurría con el Greedy, pero con una probabilidad de salto hacia algunos peores, lo que hace que la solución sea buena e incluso mejor que el Greedy, pero no la mejor.

Por lo tanto, podemos decir que por ahora el problema se resuelve de buena forma mediante Hormigas, pero cuyos resultados podrían verse aun más mejorados con mayores tiempos de ejecución o mayor número de Hormigas.

Puede verse que, en los resultados globales, la mejor solución obtenida en el fichero más pequeño es 676, a 1 del óptimo global conocido, por lo que se puede decir que el SCH da capacidad de alcanzar óptimos globales, saltando óptimos locales, pero que pueden mejorarse estos resultados con mayor tiempo de ejecución, ya que se realizarían mayores evaluaciones y con ello se podría diversificar más.

Otra opción sería probarlo en una máquina más potente para que haya más evaluaciones en el mismo tiempo de ejecución, consiguiendo mejores resultados posiblemente si el número de evaluaciones es mucho mayor.

Para demostrar esto último, se ha creado una tabla que resume un pequeño estudio de las mejoras propuestas, mediante la ejecución en una máquina más potente de sobremesa, en la que se han realizado 5 ejecuciones del algoritmo y con dos ficheros, el

## MB Y HB

más pequeño y el más grande, utilizando el algoritmo SCH para poder ver si se podría alcanzar el óptimo global en problemas de mayor envergadura. (Figura 2.1 y 2.2)

Fichero	St70			
SCH	Coste-Medio	Evaluaciones	Hormigas	Tiempo (minutos)
	675,4	730360	20	5
	680,1	294920	40	2

Figura 2.1

Fichero	Vm1084			
SCH	Coste-Mejor	Evaluaciones	Hormigas	Tiempo (minutos)
	296619	3420	20	25
	296120	8340	20	50
	291963	1560	40	12
	290891	6920	40	50
	285146	3360	80	25

Figura 2.2

En la Figura 2.1 puede verse que con poco tiempo y un buen numero de hormigas, el sistema es capaz de encontrar casi un optimo con un tiempo de ejecución de 2 minutos, y que con una máquina más potente y con el mismo numero de Hormigas y tiempo de ejecución, se puede llegar al óptimo, y esto se debe a la cantidad de evaluaciones hechas.

Por último, en la Figura 2.2, se ha creado mediante un estudio basado en si es mejor mas tiempo solamente, o si las hormigas influyen en la solución de un problema grande, y vemos que el mejor resultado lo da un numero de hormigas de 80, con el mismo tiempo de ejecución, quedándose a 40000 del optimo, lo cuál no está mal viendo sus soluciones predecesoras.

En definitiva, el problema está en ajustar el problema a un numero de hormigas bueno dependiendo del problema al que se va a enfrentar, y ajustando su tiempo de ejecución a ese numero de hormigas y a su complejidad, para intentar minimizar el gasto en computo y en tiempo, es decir, intentar minimizar los tiempos de ejecución gracias a un estudio de los parámetros propuestos para alcanzar mejores soluciones.



### Maquina usada: Portátil Asus

- **SO:** *Windows 10 x64*
- **Procesador:** *Intel Core i7-5500U. 2.4GHz*
- **RAM:** *8GB*
- **Gráfica:** *NVIDIA GEFORCE 820M*
- **Ventiladores:** *Default*

### Máquina más potente usada: Ordenador de sobremesa

- **SO:** *Windows 7 x64*
- **Procesador:** *Intel Core i5-4460. 3.2GHz*
- **RAM:** *8GB*
- **Gráfica:** *NVIDIA GTX 690 2GB RAM*
- **Ventiladores:** *Nox x3*
- **Disipador:** *Cooler Master Hyper 12 TX3 EVO*

---

## Conclusiones

---

Tras el estudio de los algoritmos con los distintos data sets, tras el estudio de las mejoras, y tras ver la comparativa global de los algoritmos con Greedy de la práctica 1, puede decirse que el sistema basado en hormigas es mejor que Greedy, ya que tiene probabilidades de saltar a los mejores locales, que conlleven a la consecución de la mejor solución, aunque para ello necesita de bastantes evaluaciones (cuya convergencia está limitada por tiempo), y con las que los algoritmos van bastante bien en cuanto a soluciones, alcanzándose incluso caso el mejor global en el fichero más pequeño.

Para conseguir esto en los ficheros mas grandes y por lo cual, problemas más complejos, necesitarían de un mayor tiempo de ejecución, una máquina más potente y/o un mayor número de hormigas, por lo cual la dificultad para resolver este problema reside en la capacidad del programador para minimizar los costes de computo gracias a la definición de buenos parámetros.