



**UNIVERSIDAD
DE GRANADA**

METAHEURÍSTICA

Práctica Final.

Uso de una distinta Metaheurística

Lukas Häring García

1. Introducción.	2
2. Descripción de los algoritmos.	4
3. Análisis de los resultados	8
3.1 Dataset "Rand".	8
3.1.1. Análisis Tabla Colonia de Abejas.	8
3.1.2. Análisis Tabla Colonia de Abejas Mejorado.	9
3.1.3. Análisis Tabla de Resultados Medios.	10
3.2 Dataset "Iris".	11
3.2.1. Análisis Tabla Colonia de Abejas.	11
3.2.2. Análisis Tabla Colonia de Abejas Mejorado.	12
3.2.3. Análisis Tabla de Resultados Medios.	13
3.3 Dataset "Ecoli".	14
3.3.1. Análisis Tabla Colonia de Abejas.	14
3.3.2. Análisis Tabla Colonia de Abejas Mejorado.	15
3.3.3. Análisis Tabla de Resultados Medios.	16
3.4 Dataset "Newthyroid".	17
3.4.1. Análisis Tabla Colonia de Abejas.	17
3.4.2. Análisis Tabla Colonia de Abejas Mejorado.	18
3.4.3. Análisis Tabla de Resultados Medios.	19
3.5 Análisis Global de los resultados.	20
4. Bibliografía	21

1. Introducción.

Para la realización de esta última práctica, se ha elegido un algoritmo de **Inteligencia de Enjambres** (SI), introducido por Gerardo Beni y Jing Wan en 1989, estos algoritmos están basados en comportamientos colectivos descentralizados y auto-organizados.

En particular, el algoritmo elegido dentro de esta colección es el **Sistema de Abejas de 1997**, desarrollado en la Universidad de Keiō por dos investigadores, Tomoya Sato y Masafumi Hagiwara del departamento de Ingeniería Eléctrica, inspirándose en el comportamiento colectivo de las abejas.

El comportamiento que estos investigadores observaron sobre un grupo de abejas era como primero algunas de estas encontraban comida y seguidamente era notificado a demás abejas utilizando un baile que las hacía trabajar en equipo.

Para la aproximación, utilizaron **dos técnicas de búsqueda** mediante una **metaheurística basada en genética** como la realizada en la práctica 2, aunque estos han utilizado una representación real.

Entre las técnicas de búsqueda encontramos dos, una **búsqueda global**, que simula la búsqueda de posible comida (o puntos fuertes de posible convergencia) y una **búsqueda local** que ayuda a las abejas atraídas (por el baile) a converger hacia el lugar de la comida (solución). Ambas técnicas son implementadas utilizando un esquema evolutivo genético, proponiendo un algoritmo de cruce específico para el problema.

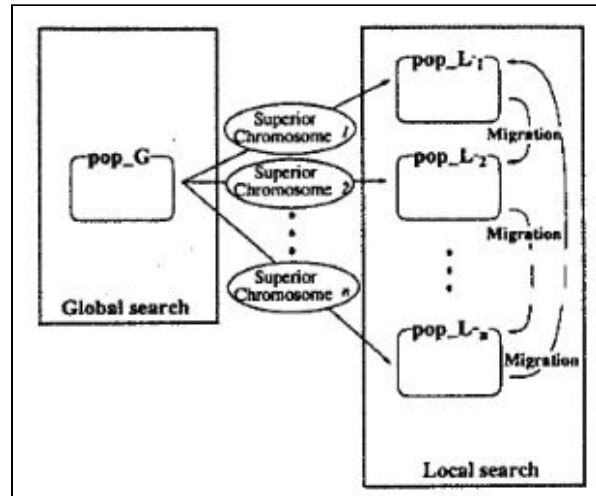
Se ha elegido esta metaheurística debido a que engloba estrategias vistas en prácticas anteriores, así como posibles mejoras inspiradas en uno de los animales más importantes de nuestro planeta y nuestra existencia. Además, se trata de un algoritmo relativamente antiguo, aunque muy estudiado años siguientes, del que podemos modelar y proponer mejoras.

Cabe destacar que estos algoritmos han sido implementados desde el principio debido a la escasez de código en el modelo inicial, ya que estos se fueron implementando en 2005 con un algoritmo más consistente ABC[\[1\]](#).

2. Descripción de los algoritmos.

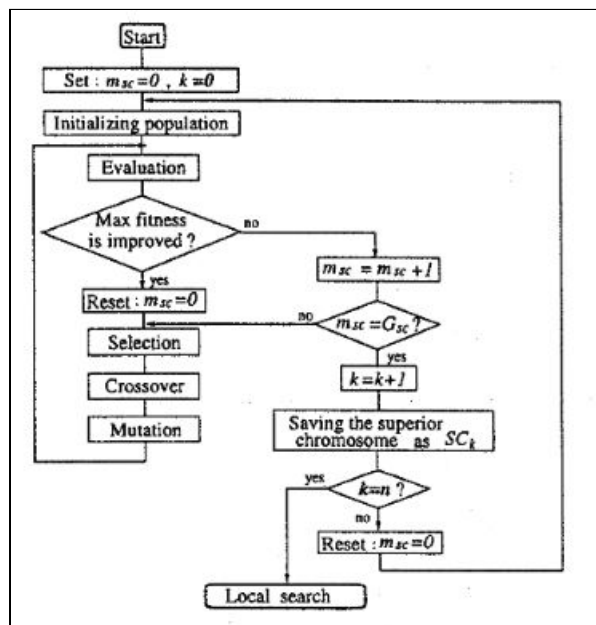
Como se ha comentado anteriormente, el algoritmo hace uso de una búsqueda global inicial, de éste, tomará “n” cromosomas superiores, se considerará un cromosoma superior a otro si no se ha encontrado otro mejor en un número prefijado de iteraciones, se ofrecerá la implementación de éste algoritmo más adelante.

Los cromosomas superiores serán utilizados para la generación inicial utilizada en la búsqueda local. Para cada iteración de la búsqueda local, el algoritmo permite migrar individuos a otras poblaciones.



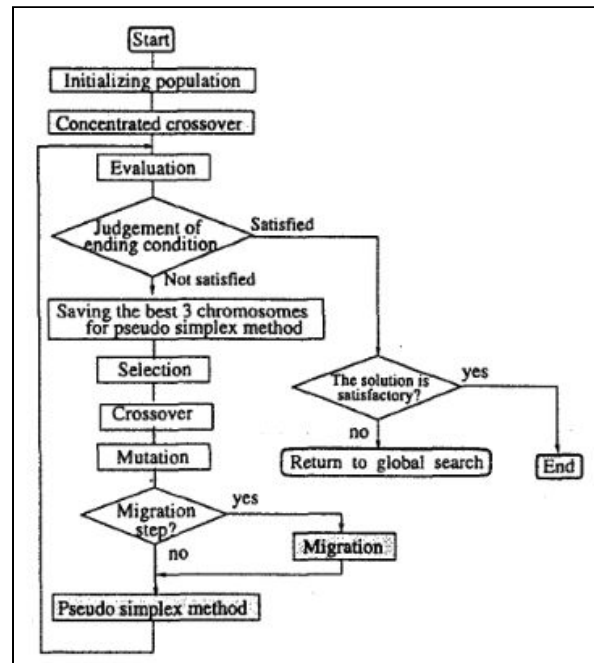
En primer lugar, se va a comentar el algoritmo de búsqueda global, aunque a simple vista parezca complejo, se trata de un algoritmo naïve, basado en los algoritmos genéticos.

En primer lugar, crearemos nuestra población actual, para cada iteración del algoritmo genético, tendremos un contador de veces que no ha mejorado desde la última modificación de nuestra población, según el paper, el umbral es de **20 iteraciones** sin ser modificado, si es cierto, nos quedaremos el mejor cromosoma, este proceso lo realizaremos tantas veces como deseemos, en particular, “n” veces como se ha comentado anteriormente, el paper considera suficiente para sus problemas, un número total de **tres** cromosomas superiores.



La búsqueda local es algo más compleja, genera una población inicial y utiliza los cromosomas superiores obtenidos por la búsqueda global para cruzarlos entre estos, para cada población local k-ésima, utiliza el cromosoma superior k-ésimo y cruza todos los de la población con este, consiguiendo así elementos cercanos a dicho cromosoma superior.

Seguidamente, repetirá el mismo esquema evolutivo que el utilizado en el algoritmo global, aunque previamente, vamos a almacenar los tres mejores cromosomas los cuales serán utilizados para lo que ellos llaman “Pseudo Simplex Method”, que veremos posteriormente.



Una vez realizada la pasada por el esquema genético, tomaremos un elemento aleatorio que será transferido a otra población local. Seguidamente a los tres mejores cromosomas de la población anterior, aplicaremos la técnica “Pseudo Simplex Method” e introduciremos los cromosomas resultantes de esta técnica en nuestra nueva población.

Repetiremos el proceso hasta completar las

$$\frac{(100.000 - evaluaciones_{globales})}{n} \text{ iteraciones}$$

para cada población local, donde 100.000 es el umbral del número de evaluaciones de la función objetivo, especificado en las prácticas. $evaluaciones_{globales}$, el número de evaluaciones de la función objetivo realizado por el algoritmo de búsqueda global. n es el número de cromosomas superiores.

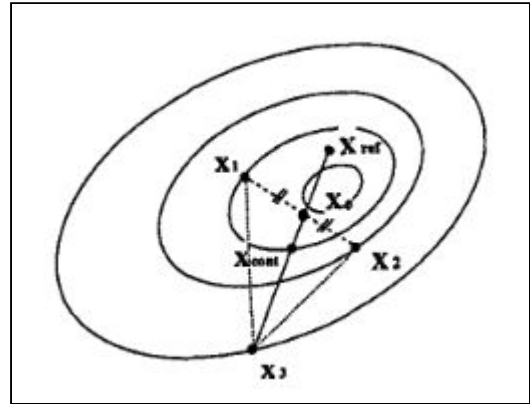
Para modelar este problema y simular la “migración”, se elige de forma no determinista la siguiente población local a realizar una pasada por el algoritmo descrito, así hasta que todos cumplan la satisfactoriedad de la solución, que es haber realizado el número de iteraciones expuesto.

A continuación se va a presentar el algoritmo “Pseudo Simplex Method”, este hace uso de los 3 mejores cromosomas (reales) de la iteración anterior, X_1 , X_2 y X_3 y dos variables α y β , que será utilizados como moduladores de importancia.

Definimos,

$$X_o = \frac{X_1 + X_2}{2}$$

que se trata del cromosoma intermedio del segmento X_1 y X_2 , a partir de este, definimos como los cromosomas resultantes.



Un primer cromosoma, $X_{cont} = (1 - \beta)X_o + \beta X_3$, como $\beta \in [1, 0]$ dónde es fácil observar que X_{cont} es una interpolación lineal de X_o hasta X_3 . Si $\beta = \frac{1}{3}$, es fácil observar que se trata del centro del triángulo formado por los tres cromosomas $X_{cont} = \frac{X_1 + X_2 + X_3}{3}$, favoreciendo la convergencia hacia la solución.

El otro cromosoma será $X_{ref} = (1 + \alpha)X_o - \alpha X_3$, que se trata de una extrapolación. De manera equivalente al anterior, si $\alpha = \frac{1}{3}$, se tratará del centro del triángulo formado por los tres cromosomas donde X_3 está reflejado sobre X_1 y X_2 , $X_{ref} = \frac{2(X_1 + X_2) - X_3}{3}$ este ayudará a salir de la solución si esta no fuera global.

Finalmente, devolveremos tres cromosomas, X_1 , X_{cont} y X_{ref} , como vemos, hemos introducido X_1 , que es el mejor cromosoma de la iteración anterior, por lo que este algoritmo en su naturaleza, también trabaja como elitismo, así, no perderemos nunca la mejor solución encontrada hasta ahora.

El paper nos recomienda unos parámetros para $\alpha = 0.4$ y $\beta = 0.4$, aunque en realidad se trabaje con números reales, nosotros estamos trabajando con números enteros, por lo que debemos asegurarnos de que los valores siempre están en el rango de trabajo, es fácil observar que X_{cont} lo está, ya que es la media, mientras que X_{ref} no, por eso, vamos a realizar la técnica de “clamping”, todos aquellos valores que se salgan del extremo, serán desplazados al extremo más cercano.

Además se aplica un corrector sobre este algoritmo ya que puede ocurrir que no los conjuntos se queden vacíos, por lo que mutará un valor que obligue a tener al menos una vez dicho valor.

Para los algoritmos genéticos en las iteraciones globales y locales, se ha utilizado como algoritmo de cruce, el mejor obtenido en la segunda práctica, el cruce uniforme.

Se han elegido una probabilidad de mutación de 5 por ciento por cada gen y la probabilidad de cruce del 70 por ciento. Aunque en el paper se utiliza una constante de 0.5% para la probabilidad de mutación, cabe destacar que este incremento a resultado en una convergencia más rápida.

Puesto que el algoritmo per se tiene incluido una hibridación, ya que combina búsquedas con algoritmos genéticos, no se han realizado mejoras sobre este más que un “tanteo” de valores para la mutación, ya que el la probabilidad de cruce ha resultado ser buena. También se ha probado en pocas iteraciones el resultado del cruce por segmento fijo, el cual ha obtenido peores resultados que el utilizado, el cruce uniforme.

Las mejoras han sido bastante simples, pero eficaces, ya que se trata de un algoritmo muy completo. Se han utilizado políticas de reemplazamiento, esto quiere decir que, tanto la migración como “Pseudo Simplex Method”, reemplazan siempre de manera aleatoria cualquier individuo de la población, haciendo que el mejor individuo pudiera ser reemplazado.

Para la migración es simple, compara el peor individuo con el migrante, si el migrante es mejor, entonces el peor individuo se ve reemplazado. Por otro lado, “Pseudo Simplex Method” reemplaza 3 individuos, los cuales no tienen porqué tener relación de orden, por ello, sustituiremos el k-ésimo peor individuo de la población actual, si es posible, ya que deben ser mejores, asegurándonos también que ninguno de los 3 sea reemplazado por los restantes a asignar.

Para terminar, podemos ver cómo el algoritmo explota utilizando la búsqueda global, seguidamente explora utilizando la local. Esta explotación es controlada, quiere decir, los cromosomas explotados son posibles puntos de convergencia. Además, la utilización de un algoritmo genético sobre la búsqueda local ayuda a que en ciertas ocasiones, también se explote. La población utilizada para los algoritmos genéticos son de 10 individuos, para cada búsqueda local, que tras realizar las pruebas observamos que es suficiente para la convergencia.

3. Análisis de los resultados

3.1 Dataset “Rand”.

El factor obtenido es $\lambda = 0.0072$, está definido el problema para 3 clusters.

3.1.1. Análisis Tabla Colonia de Abejas.

10% CA	RAND (k = 3)			
	<i>Tasa_C</i>	<i>Tasa_inf</i>	<i>Agr.</i>	<i>T</i>
Ejecución 1	0.75	14.00	0.85	258.60s
Ejecución 2	0.74	20.00	0.88	263.93s
Ejecución 3	0.72	0.00	0.72	249.66s
Ejecución 4	0.72	0.00	0.72	255.17s
Ejecución 5	0.72	0.00	0.72	243.07s
Media	0.73	6.80	0.78	252.28s

20% CA	RAND (k = 3)			
	<i>Tasa_C</i>	<i>Tasa_inf</i>	<i>Agr.</i>	<i>T</i>
Ejecución 1	0.72	0.00	0.72	223.01s
Ejecución 2	0.72	0.00	0.72	230.88s
Ejecución 3	0.72	0.00	0.72	232.00s
Ejecución 4	0.72	0.00	0.72	242.02s
Ejecución 5	0.72	0.00	0.72	224.38s
Media	0.72	0.00	0.72	232.45s

3.1.2. Análisis Tabla Colonia de Abejas Mejorado.

10% CA-M	RAND (k = 3)			
	<i>Tasa_C</i>	<i>Tasa_inf</i>	<i>Agr.</i>	<i>T</i>
Ejecución 1	0.72	0.00	0.72	251.01s
Ejecución 2	0.72	0.00	0.72	241.03s
Ejecución 3	0.72	0.00	0.72	240.18s
Ejecución 4	0.72	0.00	0.72	241.66s
Ejecución 5	0.72	0.00	0.72	253.55s
Media	0.72	0.00	0.72	245.48s

20% CA-M	RAND (k = 3)			
	<i>Tasa_C</i>	<i>Tasa_inf</i>	<i>Agr.</i>	<i>T</i>
Ejecución 1	0.72	0.00	0.72	241.10s
Ejecución 2	0.72	0.00	0.72	242.33s
Ejecución 3	0.72	0.00	0.72	236.20s
Ejecución 4	0.72	0.00	0.72	240.62s
Ejecución 5	0.72	0.00	0.72	236.20s
Media	0.72	0.00	0.72	238.60s

3.1.3. Análisis Tabla de Resultados Medios.

10%	RAND (k = 3)			
	<i>Tasa_C</i>	<i>Tasa_inf</i>	<i>Agr.</i>	<i>T</i>
COPKM	1.02	220.40	2.61	1.30s
BL	0.72	0.00	0.72	3.00s (1529 its)
ES	0.72	0.00	0.72	124.06s
AGE-SF	0.72	0.00	0.72	232.35s
CA	0.73	6.80	0.78	105.78s
CA-M	0.72	0.00	0.72	245.48s

20%	RAND (k = 3)			
	<i>Tasa_C</i>	<i>Tasa_inf</i>	<i>Agr.</i>	<i>T</i>
COPKM	0.85	238.00	1.71	1.40s
BL	0.72	0.00	0.72	3.08s (1555 its)
ES	0.72	0.00	0.72	91.55s
AGE-SF	0.72	0.00	0.72	225.81s
CA	0.72	0.00	0.72	232.45s
CA-M	0.72	0.00	0.72	238.60s

3.2 Dataset “Iris”.

El factor obtenido es $\lambda = 0.0063$, también definido para 3 clusters.

3.2.1. Análisis Tabla Colonia de Abejas.

10% CA	IRIS (k = 3)			
	<i>Tasa_C</i>	<i>Tasa_inf</i>	<i>Agr.</i>	<i>T</i>
Ejecución 1	0.71	64.00	1.12	245.90s
Ejecución 2	0.67	0.00	0.67	253.09s
Ejecución 3	0.67	0.00	0.67	246.64s
Ejecución 4	0.67	0.00	0.67	243.43s
Ejecución 5	0.67	0.00	0.67	247.29s
Media	0.68	12.80	0.76	247.07s

20% CA	IRIS (k = 3)			
	<i>Tasa_C</i>	<i>Tasa_inf</i>	<i>Agr.</i>	<i>T</i>
Ejecución 1	0.67	0.00	0.67	231.73s
Ejecución 2	0.67	0.00	0.67	245.51s
Ejecución 3	0.71	224.00	1.42	234.44s
Ejecución 4	0.67	40.00	0.80	242.40s
Ejecución 5	0.67	0.00	0.67	248.04s
Media	0.68	52.80	0.85	240.43s

3.2.2. Análisis Tabla Colonia de Abejas Mejorado.

10% CA-M	IRIS (k = 3)			
	<i>Tasa_C</i>	<i>Tasa_inf</i>	<i>Agr.</i>	<i>T</i>
Ejecución 1	0.67	0.00	0.67	242.74s
Ejecución 2	0.67	0.00	0.67	246.99s
Ejecución 3	0.67	0.00	0.67	238.39s
Ejecución 4	0.67	0.00	0.67	237.84s
Ejecución 5	0.67	0.00	0.67	236.98s
Media	0.67	0.00	0.67	240.59s

20% CA-M	IRIS (k = 3)			
	<i>Tasa_C</i>	<i>Tasa_inf</i>	<i>Agr.</i>	<i>T</i>
Ejecución 1	0.67	0.00	0.67	237.90s
Ejecución 2	0.67	0.00	0.67	239.70s
Ejecución 3	0.67	0.00	0.67	237.88s
Ejecución 4	0.67	0.00	0.67	240.84s
Ejecución 5	0.67	0.00	0.67	243.86s
Media	0.67	0.00	0.67	240.04s

3.2.3. Análisis Tabla de Resultados Medios.

10%	IRIS (k = 3)			
	<i>Tasa_C</i>	<i>Tasa_inf</i>	<i>Agr.</i>	<i>T</i>
COPKM	0.90	492.8	3.20	1.94s
BL	0.67	0.00	0.67	3.04s (1556 its)
ES	0.67	0.00	0.67	111.14s
AGE-SF	0.67	0.00	0.67	218.00s
CA	0.68	12.80	0.76	247.07s
CA-M	0.67	0.00	0.67	240.59s

20%	IRIS (k = 3)			
	<i>Tasa_C</i>	<i>Tasa_inf</i>	<i>Agr.</i>	<i>T</i>
COPKM	0.98	727,6	4.10	1.73s
BL	0.67	0.00	0.67	3.63s (1781 its)
ES	0.67	0.00	0.67	88.79s
AGE-SF	0.67	0.00	0.67	211.27s
CA	0.68	52.80	0.85	240.43s
CA-M	0.67	0.00	0.67	240.04s

3.3 Dataset “Ecoli”.

El factor obtenido es $\lambda = 0.0268$, un problema con 8 clusters.

3.3.1. Análisis Tabla Colonia de Abejas.

10% CA	ECOLI (k = 8)			
	<i>Tasa_C</i>	<i>Tasa_inf</i>	<i>Agr.</i>	<i>T</i>
Ejecución 1	40.06	2152.00	97.80	1096.99s
Ejecución 2	40.22	2000.00	93.88	1163.28s
Ejecución 3	39.05	1708.00	84.87	1175.53s
Ejecución 4	39.95	2002.00	93.66	1109.88s
Ejecución 5	41.36	2008.00	95.23	1128.49s
Media	40.13	1974.00	93.09	1134.83s

20% CA	ECOLI (k = 8)			
	<i>Tasa_C</i>	<i>Tasa_inf</i>	<i>Agr.</i>	<i>T</i>
Ejecución 1	38.28	3384.00	83.68	894.22s
Ejecución 2	40.67	2990.00	80.78	794.99s
Ejecución 3	36.20	4166.00	92.09	768.57s
Ejecución 4	38.41	3626.00	87.05	811.25s
Ejecución 5	39.30	4210.00	95.77	898.49s
Media	38.57	3675.20	87.87	837.10s

3.3.2. Análisis Tabla Colonia de Abejas Mejorado.

10% CA-M	ECOLI (k = 8)			
	<i>Tasa_C</i>	<i>Tasa_inf</i>	<i>Agr.</i>	<i>T</i>
Ejecución 1	37.53	1840.00	86.90	790.39s
Ejecución 2	39.06	1665.00	83.71	833.15s
Ejecución 3	40.46	1674.00	85.37	816.52s
Ejecución 4	36.50	1196.00	68.59	812.10s
Ejecución 5	38.63	1720.00	84.78	821.21s
Media	38.44	1618.80	81.87	814.67s

20% CA-M	ECOLI (k = 8)			
	<i>Tasa_C</i>	<i>Tasa_inf</i>	<i>Agr.</i>	<i>T</i>
Ejecución 1	36.34	2660.00	72.02	1087.95s
Ejecución 2	37.61	1924.00	63.42	1043.60s
Ejecución 3	37.20	2026.00	64.38	1043.57s
Ejecución 4	36.89	2300.00	67.74	1121.13s
Ejecución 5	36.48	3022.00	77.01	1053.91s
Media	36.90	2386.40	68.91	1070.03s

3.3.3. Análisis Tabla de Resultados Medios.

10%	ECOLI (k = 8)			
	<i>Tasa_C</i>	<i>Tasa_inf</i>	<i>Agr.</i>	<i>T</i>
COPKM	29.16	1452.00	68.11	251.01s
BL	23.35	63.20	25.04	250.48s (22994 its)
ES	22.56	32.80	23.44	888.41s
AGE-SF	27.93	502.40	34.47	1562.72s
CA	40.13	1974.00	93.09	1134.83s
CA-M	38.44	1618.80	81.87	814.67s

20%	ECOLI (k = 8)			
	<i>Tasa_C</i>	<i>Tasa_inf</i>	<i>Agr.</i>	<i>T</i>
COPKM	30.47	2737.60	67.20	236.57s
BL	22.29	102.0	23.66	220.88s (20024 its)
ES	22.48	87.60	23.66	938.59s
AGE-SF	22.19	164.80	23.90	1567.63s
CA	38.57	3675.20	87.87	837.10s
CA-M	36.90	2386.40	68.91	1070.03s

3.4 Dataset “Newthyroid”.

El factor obtenido es $\lambda = 0.0365$, Con una agrupación de 3 clusters.

3.4.1. Análisis Tabla Colonia de Abejas.

10% CA	NEWTHYROID (k = 3)			
	<i>Tasa_C</i>	<i>Tasa_inf</i>	<i>Agr.</i>	<i>T</i>
Ejecución 1	12.98	262.00	22.56	424.28s
Ejecución 2	14.09	4.00	14.23	421.60s
Ejecución 3	16.64	242.00	25.49	405.01s
Ejecución 4	13.94	44.00	15.55	384.94s
Ejecución 5	14.27	12.00	14.71	377.35s
Media	14.39	112.80	18.51	402.64s

20% CA	NEWTHYROID (k = 3)			
	<i>Tasa_C</i>	<i>Tasa_inf</i>	<i>Agr.</i>	<i>T</i>
Ejecución 1	14.10	56.00	15.13	436.94s
Ejecución 2	14.28	126.00	16.58	375.27s
Ejecución 3	14.11	46.00	14.95	369.82s
Ejecución 4	14.56	82.00	16.05	371.55s
Ejecución 5	14.05	660.00	26.12	369.62s
Media	14.22	194.00	17.77	384.64s

3.4.2. Análisis Tabla Colonia de Abejas Mejorado.

10% CA-M	NEWTHYROID (k = 3)			
	<i>Tasa_C</i>	<i>Tasa_inf</i>	<i>Agr.</i>	<i>T</i>
Ejecución 1	12.28	176.00	18.72	457.56s
Ejecución 2	14.09	4.00	14.23	429.96s
Ejecución 3	14.09	4.00	14.23	390.02s
Ejecución 4	14.06	8.00	14.35	389.31s
Ejecución 5	14.09	4.00	14.23	392.93s
Media	13.72	39.20	15.15	411.96s

20% CA-M	NEWTHYROID (k = 3)			
	<i>Tasa_C</i>	<i>Tasa_inf</i>	<i>Agr.</i>	<i>T</i>
Ejecución 1	13.61	320	19.46	403.41s
Ejecución 2	14.26	74.00	15.61	389.15s
Ejecución 3	14.29	0.00	14.29	393.04s
Ejecución 4	14.29	0.00	14.29	394.68s
Ejecución 5	10.8	546	20.78	396.68s
Media	13.45	188.00	16.89	395.4s

3.4.3. Análisis Tabla de Resultados Medios.

10%	NEWTHYROID (k = 3)			
	<i>Tasa_C</i>	<i>Tasa_inf</i>	<i>Agr.</i>	<i>T</i>
COPKM	12.53	936.80	46.79	5.68s
BL	12.81	79.6	15.72	10.93s (3344 its)
ES	13.40	41.60	14.92	211.68s
AGE-SF	13.42	42.40	13.53	897.16s
CA	14.39	112.80	18.51	402.64s
CA-M	13.72	39.20	15.15	411.96s

20%	NEWTHYROID (k = 3)			
	<i>Tasa_C</i>	<i>Tasa_inf</i>	<i>Agr.</i>	<i>T</i>
COPKM	13.04	1726.40	44.60	5.77s
BL	12.91	184.8	16.28	8.59s (2591 its)
ES	13.58	92.0	15.27	217.79s
AGE-SF	22.19	164.80	23.90	1567.63s
CA	14.22	194.00	17.77	384.64s
CA-M	13.45	188.00	16.89	395.4s

3.5 Análisis Global de los resultados.

Como vemos, se han recolectado los resultados de alguna de las mejores metaheurísticas y algoritmos, de las prácticas anteriores para comparar, entre ellos está CPOKM que se trata de la versión greedy, BL o búsqueda local, ES o Enfriamiento Simulado y AGE-SF Algoritmo Genético Estacionario por Segmento Fijo.

Vemos que en general tenemos buenos resultados, “CA” o Colonia de Abejas, converge en muchos de los casos al valor óptimo, sobre todo en datasets simples, mientras que con “CA-M” o Colonia de Abejas Mejorado, su convergencia en estos problemas es siempre el óptimo. Esto es debido a que no se reemplaza la mejor solución y ayuda así a converger más rápidamente. Respecto a los demás algoritmos, podemos destacar que requiere de más tiempo de cómputo pero en general son resultados buenos.

Para dataset más complejos como Newthyroid o Ecoli, los resultados son mejores que el algoritmo Greedy pero peores que las demás, esto es debido a su lenta convergencia, ya que en realidad es peor en la rapidez de convergencia, no en soluciones. Cabe destacar que por problemas de tiempo, no se ha podido incrementar el número de iteraciones para estos problemas, pero es esperable una convergencia a una solución buena-óptima.

Para el dataset de Ecoli, vemos que la versión mejorada es casi 20 puntos mejor que la versión original de 1999, con el mismo número de iteraciones. Estas mejoras ayudan a mantener la mejor solución trabajando como elitismo, así siempre se cruzará una solución buena con otra peor e intentar encontrar un resultado mejor.

No se ha realizado un estudio de los parámetros “Pseudo Simplex Method”, ya que como se ha analizado antes, $\alpha = \beta = \frac{1}{3}$ ofrece unos cromosomas lo menos sesgados, donde uno trabaja como interpolador y el otro como extrapolador para así encontrar dos posibles soluciones además, sean eficientes de calcular.

Para concluir, este algoritmo mantiene en balanza la explotación y la exploración, aunque su convergencia sea lenta. Los resultados son buenos, para problemas sencillos, obtenemos el óptimo, para los datasets más complejos, necesitamos un mayor número de iteraciones. La capacidad de explorar diferentes óptimos gracias a la búsqueda de distintas regiones buenas gracias a los cromosomas superiores.

4. Bibliografía

1. *S. Tomoya, M. Hagiwara. An Efficient K-Means Clustering Algorithm.* “In this paper we propose an improved genetic algorithm named Bee System. The concept of the Bee System comes from behavior of bees: first, a bee finds feed and then it notifies the information to the other many bees by dance to work together. In the proposed Bee System, each chromosome tries to find good solution individually...”. Source: <https://ieeexplore.ieee.org/document/633289>