Carlota Valdivia Manzano
Doble Grado de Ingeniería Informática y Matemáticas
Curso 2020/21

The Bees Algorithm

Metaheurísticas: Práctica Alternativa

UNIVERSIDAD
DE GRANADA

1. Descripción de la metaheurística

La metaheurística escogida para la implementación de esta práctica es Bees Algorithm (BA).

Bees Algorithm es un algoritmo de optimización basado en enjambres (SOA) que utiliza métodos de la naturaleza para buscar una solución óptima. En concreto, este algoritmo está inspirado en el comportamiento de búsqueda de alimento de enjambres de abejas melíferas.

Dicho algoritmo, en su versión más simple, realiza una especie de búsqueda local combinada con búsqueda aleatoria, que puede utilizarse tanto para la optimización combinatoria como para la funcional. Concretamente en la versión implementada se centra en este último.

La diferencia que hay entre los algoritmos de optimización basados en enjambres y los de búsqueda directa es que los SOA utilizan una población de soluciones para cada iteración en lugar de una única solución.

A continuación, se describirá de forma más exhaustiva el comportamiento en el que se basa esta metaheurística.

1.1 Comportamiento de las abejas en la naturaleza

Una colonia de abejas melíferas se extiende a través de extensos campos en busca de buenas zonas con flores que tengan una abundante cantidad de polen. Aquellas que pueden recolectarse con un menor esfuerzo deben recibir un mayor número de abejas frente aquellas cuyas flores tienen menor polen que recolectar.

El proceso de búsqueda de alimento comienza cuando se envían abejas exploradoras a buscar zonas prometedoras. Estas se mueven de forma aleatoria de una zona a otra. Una vez regresan a la colmena, aquellas abejas que han encontrado las mejores zonas con flores con más cantidad de polen mejor recolectable, depositan su polen y proceden a comunicarles la información sobre el lugar encontrado al resto de abejas a través del “waggle dance” o “baile del meneo”.

Este baile contiene información sobre la dirección en la que se encuentra el lugar prometedor, su distancia a la colmena y su aptitud o calidad. Gracias a esta información las abejas son capaces de llegar hasta él y evaluar las distintas zonas buscadas.

Después, la abeja que ha realizado el baile regresa a la zona de flores con otras abejas. En dicho reparto se envían más abejas a las zonas más prometedoras.

Mientras las abejas extraen el néctar de las flores prometedoras siguen controlando la calidad del alimento, para que cuando regresen a la colmena puedas seguir decidiendo si la zona sigue

siendo lo suficientemente buena como para informarlo a través del baile y reclutar más abejas para que vayan a ella.

1.2 Algoritmo de abejas

El algoritmo requiere que se establezcan una serie de parámetros para su implementación:

1. **Abejas exploradoras (n)**: número de abejas exploradoras total.
2. **Número de sitios seleccionados (m)**: se seleccionan m sitios de los n totales.
3. **Número de los mejores sitios de los seleccionados (e)**: se seleccionan los e mejores de los m seleccionados.
4. **Abejas reclutadas para los mejores sitios (nep):** número de abejas que irán a los e mejores sitios seleccionados.
5. **Abejas reclutadas para los otros sitios (ngh):** número de abejas que irán a los sitios seleccionados que no son los mejores (m - e)
6. **Tamaño inicial de las zonas (ngh)**: tamaño inicial de las zonas que irá aumentado con una constante con valor entre 0 y 1.

Además requiere de un criterio de para que en este caso será un determinado número de iteraciones.

El pseudocódigo del programa es en su forma más básica el siguiente.

1. Initialise population with random solutions.
2. Evaluate fitness of the population.
3. While (stopping criterion not met)
   //Forming new population.
4. Select sites for neighbourhood search.
5. Recruit bees for selected sites (more bees for best e sites) and evaluate fitnesses.
6. Select the fittest bee from each patch.
7. Assign remaining bees to search randomly and evaluate their fitnesses.
8. End While.

Fig.1 Pseudo code of the basic bees algorithm

Nota: Pseudocódigo de The Bees Algorithm – A Novel Tool for Complex Optimisation Problems by D.T. Pham, A. Ghanbarzadeh, E. Koç, S. Otri , S. Rahim , M. Zaidi

El algoritmo empieza con n abejas exploradoras situadas en lugares que se han generado de forma aleatoria. Las calidad de la zona en la que se encuentran será evaluada mediante una función.

A continuación, señalaremos cuál es la mejor abeja y su fitness, y mientras no se cumpla el criterio de parada, que en este caso será que alcance un determinado número de iteraciones, se procederá a repetir la siguiente secuencia de pasos.

Primero seleccionamos de las n abejas las m que tienen la aptitud más apta y nos quedaremos con los sitios en los que están como “Sitios seleccionados”. Dentro de esta selección subdividiremos entre las mejores y las restantes.

Después, reclutaremos un determinado número de abejas para los mejores sitios seleccionados y otras tantas para el resto de sitios seleccionados que no son los mejores. A cada una de dicha abeja, les realizaremos unas pequeñas variaciones, que serán como ligeros movimientos en su posición para ver si mejora el fitness de las zonas en las que se encuentran.

Una vez hecho esto seleccionaremos a la abeja que haya encontrado la mejor zona y por tanto tenga el mejor fitness, y luego las abejas restantes de la población, las no seleccionadas, se asignan de forma aleatoria en el espacio de búsqueda total para buscar nuevas soluciones potenciales.

Así pues, al final de cada iteración obtendremos una colonia de abejas divida en representantes de cada zona seleccionada y otras abejas exploradoras asignadas a realizar búsquedas aleatorias.

2. Implementación de la metaheurística

La práctica se ha desarrollado en C++ y se ha hecho uso del framework de metaheurísticas: CEC 2017 Special Session and Competition on Single Objective Bound Constrained Real-Parameter Numerical Optimization. de Daniel Molina Cabrera.

Asimismo, se han utilizado distintas **librerías** como *random, vector, limits* y *algorithm.* La librería *random* ha sido empleada para poder generar de forma aleatoria los elementos de la solución, *algorithm* para utilizar una función de ordenación de vectores (sort), y librerías como *vector* para poder utilizar las distintas estructuras de datos necesitadas.

2.1 Representación de las soluciones

Para representar las soluciones del algoritmo se ha creado una clase para las abejas y un struct para la población o colonia de abejas.

2.1.1 Clase Bee

La clase abeja contiene la posición en la que se sitúa y la calidad del polen que en este caso es el fitness. Además cuenta con dos constructores, uno con parámetros y otro sin parámetros, y dos getters y dos setters para establecer, consultar o modificar los atributos de la clase.

Para definir el fitness se ha hecho uso de una función del framework empleado: **cec17_fitness.**

2.1.2 Struct Poblacion

El struct Población tiene los atributos: tamaño de la población, vector de objetos de la clase, índice de la mejor abeja y fitness de la mejor abeja.

2.2 Funciones auxiliares

Para desarrollar la metaheurística ha sido necesario implementar diversas funciones adicionales.

2.2.1 Función movimiento

La función **movimiento** es una función que se correspondiese con el ligero movimiento de una abeja.

En él se toma la posición actual de una abeja, de entre aquellas que han sido reclutadas para ir a los sitios seleccionados.

Después calculamos un entero aleatorio k y un real aleatorio t. El entero va a determinar la componente del vector dimensión a modificar. El real va a determinar lo que se mueve la abeja, está comprendido entre (-ngh,ngh).

Por último, actualizamos la posición de la abeja, sumando a la componente k-ésima del vector posición la cantidad t.

2.2.2 Función criterioOrdenacion

La función **criterioOrdenacion** sirve para hacer una comparación entre dos abejas. En este caso se ordena según su calidad o fitness, aquellas que tienen un fitness menor, se dicen que son menores que el resto de abejas.

A la hora de ordenar una población de abejas, utilizamos este criterio para ordenar de forma ascendente las abejas según la calidad de su zona. De esta forma, tendremos en la posición 0 la mejor abeja, y en la última posición la peor.

2.2.3 Función uniformDistribution

La función **uniformDistribution** es una función para calcular números reales de forma aleatoria entre dos números reales que se pasan como argumento.

2.3 Función BA

Para la implementación del algoritmo se ha seguido el pseudocódigo del algoritmo.

Hemos inicializado la población de abejas de forma aleatoria. A continuación, se repite el mismo proceso mientras el número de iteraciones sea menor que el máximo, que será equivalente a dimensión * 10000.

Dicho proceso comienza por crear nuevas abejas a partir de aquellas que han sido reclutadas y moverlas ligeramente de su posición. Con dichos movimientos nos quedaremos con el mejor y si es mejor que la mejor de la población nos la quedamos.

Al resto de abejas que no han sido seleccionadas se vuelve a calcular su posición de forma aleatoria, y se reordenan las abejas para repetir el proceso con la salvedad de que en la siguiente iteración del bucle el tamaño de la zona se reducirá a partir de una constante entre 0 y 1.

2.4 Función main

En el main se ejecuta un número de archivos y cada uno en un determinado número de ejecuciones. En este caso serán 30 archivos o funciones ejecutados 10 veces cada uno.

3. Resultados obtenidos

Para obtener los resultados y las comparaciones con diversos algoritmos hacemos uso del software proporcionado por Daniel Molina.

Primero usaremos un script de Python para generar los ficheros Excel con los resultados obtenidos en las ejecuciones. Dicho archivo se denomina **extract.py**.

A continuación, usaremos la web desarrollada también por Daniel Molina https://tacolab.org/ . En ella podemos realizar comparaciones con otras metaheurísticas.

Para realizar las comparaciones se han seleccionado dos metaheurísticas vistas en clase y con buenos resultados: Differential Evolution (DE) y Particle Swarm Optimization (PSO).

3.1 Resultados en dimensión 10

3.1.1 Evaluaciaciones 1%

	BA	DE
Best	23	7
F01	1,86E+10	2,65E+10
F02	4,72E+14	2,4E+17
F03	150510	16960,72
F04	1498,372	4561,14
F05	142,2818	164,7854
F06	82,29488	69,45639
F07	438,4333	140,8733
F08	131,1143	74,4188
F09	3531,91	1285,159
F10	2580	3038,145
F11	9009,121	3E+07
F12	1,61E+09	4,52E+09
F13	93850756	2,17E+09
F14	490854	1,62E+09
F15	3633732	2,79E+08
F16	998,2803	1112,822
F17	564,772	408,9821
F18	4,82E+08	1,17E+10
F19	41456951	9,71E+09
F20	535,2773	514,8836
F21	341,1715	600,9934
F22	1585,948	2467,55
F23	522,1796	1402,353
F24	575,0199	915,9298
F25	1817,376	2004,765
F26	2155,643	2704,509
F27	666,469	1545,307
F28	1258,925	1530,164
F29	1008,908	13680,72
F30	78501296	3,78E+08

	BA	PSO
Best	0	30
F01	1,86E+10	7,21E+09
F02	4,72E+14	1
F03	150510	31670,17
F04	1498,372	470,2425
F05	142,2818	101,3224
F06	82,29488	53,97906
F07	438,4333	176,3567
F08	131,1143	89,90927
F09	3531,91	1503,968
F10	2580	1950,94
F11	9009,121	2551,722
F12	1,61E+09	2,57E+08
F13	93850756	4721833
F14	490854	4791,901
F15	3633732	52833,05
F16	998,2803	678,6328
F17	564,772	296,675
F18	4,82E+08	28612022
F19	41456951	971843
F20	535,2773	291,292
F21	341,1715	279,1184
F22	1585,948	551,9554
F23	522,1796	426,5291
F24	575,0199	423,2538
F25	1817,376	766,362
F26	2155,643	1273,359
F27	666,469	543,3311
F28	1258,925	951,6032
F29	1008,908	615,2144
F30	78501296	11842853

	BA	DE	PSO
Best	0	4	26

3.1.2 Evaluaciaciones 100%

	BA	DE
Best	3	27
F01	5,63E+09	0
F02	6,28E+10	0
F03	18578,62	0
F04	373,9415	0,000111
F05	87,53533	115,0819
F06	49,86854	34,59707
F07	229,2824	38,48056
F08	82,9634	29,8292
F09	1431,832	193,788
F10	1854,806	359,6808
F11	593,0973	0,019419
F12	2,11E+08	4,931086
F13	2246585	5,988143
F14	2850,419	0,052398
F15	17304,73	0,060603
F16	514,1833	456,0592
F17	225,4008	23,50453
F18	4098904	0,0363
F19	58996,93	0,005192
F20	175,7806	383,6905
F21	196,3212	188,8937
F22	581,6188	100,481
F23	413,2171	809,7517
F24	421,9956	100
F25	746,2573	404,0113
F26	1134,183	270,5882
F27	485,9727	389,7283
F28	739,1424	351,7333
F29	514,0539	237,5455
F30	8844893	80512,17

	BA	PSO
Best	0	30
F01	5,63E+09	52551101
F02	6,28E+10	1
F03	18578,62	1988,879
F04	373,9415	46,84269
F05	87,53533	32,12004
F06	49,86854	10,01006
F07	229,2824	42,75206
F08	82,9634	22,03181
F09	1431,832	56,86467
F10	1854,806	1076,895
F11	593,0973	38,42737
F12	2,11E+08	2516941
F13	2246585	8408,6
F14	2850,419	99,92584
F15	17304,73	2065,841
F16	514,1833	141,4558
F17	225,4008	64,9719
F18	4098904	14840,1
F19	58996,93	3219,606
F20	175,7806	84,4391
F21	196,3212	131,952
F22	581,6188	77,40418
F23	413,2171	330,4044
F24	421,9956	181,0388
F25	746,2573	447,8069
F26	1134,183	372,9316
F27	485,9727	413,4232
F28	739,1424	469,7623
F29	514,0539	319,2936
F30	8844893	635248

	BA	DE	PSO
Best	0	21	9

3.1.3 Comparativa Ranking

	BA	DE	PSO
1	2,233333	2,633333	1,133333
2	2,3	2,566667	1,133333
3	2,6	2,266667	1,133333
5	2,9	1,733333	1,366667
10	2,9	1,366667	1,733333
20	2,9	1,333333	1,766667
30	2,9	1,333333	1,766667
40	2,9	1,333333	1,766667
50	2,9	1,366667	1,733333
60	2,9	1,4	1,7
70	2,9	1,4	1,7
80	2,9	1,4	1,7
90	2,9	1,4	1,7
100	2,9	1,4	1,7

3.1.4 Análisis de Resultados

Para llevar a cabo el análisis de resultados se han obtenido un total de siete tablas para poder llevar a cabo comparativas con los algoritmos mencionados anteriormente.

Se han obtenido tres tablas con relación al 1% de evaluaciones y otras tres con respecto al 100% de evaluaciones, y además una tabla que muestra cómo avanza el ranking de las tres metaheurísticas según el fitness consumido.

En primer lugar, se puede observar que en la tabla general con 1% de evaluaciones, el mejor con algoritmo es el PSO frente al BA que es el que peores resultados ha obtenido. No obstante, si pasamos a analizar las otras dos tablas por separado podemos apreciar que los resultados obtenidos en la mayoría de los casos son mejores en BA que en DE, pero efectivamente los de BA son peores que los de PSO.

Así bien, podemos concluir de esta información que al comienzo, cuando hemos realizado pocas evaluaciones, los resultados obtenidos por la metaheurística BA son considerablemente aceptables, ya que estos son mejores que los obtenidos con DE. Sin embargo, conforme se van aumentando las evaluaciones realizadas, en torno al 10%, se puede apreciar que nuestra metaheurística no va a ser tan buena y que podría tener demasiada exploración y que por tanto, sería necesario realizar una mayor explotación en el algoritmo.

3.2 Resultados en dimensión 30

3.2.1 Evaluaciaciones 1%

	BA	DE
Best	17	13
F01	1E+11	7,71E+10
F02	1,2E+51	1,22E+60
F03	1239623	89319,68
F04	34664,81	29757,26
F05	617,4222	477,715
F06	126,46	96,33909
F07	2371,595	787,3313
F08	575,5944	375,7772
F09	30158	11916
F10	9173,545	7712,921
F11	30676,13	1,59E+08
F12	1,71E+10	2,63E+10
F13	1,91E+10	3,84E+10
F14	24736534	7E+08
F15	3,53E+09	4,48E+09
F16	5455,94	20871,63
F17	5414,637	141470
F18	3,4E+08	3,06E+09
F19	4,57E+09	4,61E+09
F20	1618,837	1774,31
F21	740,2196	960,6489
F22	9362,158	8536,473
F23	1406,427	4232,676
F24	1644,988	2667,144
F25	12502,41	5280,954
F26	12161,52	11976,86
F27	2088,027	6075,636
F28	8382,887	6609,534
F29	10888,46	101468
F30	2,55E+09	8,72E+09

	BA	PSO
Best	0	30
F01	1E+11	3,43E+10
F02	1,2E+51	1
F03	1239623	202045
F04	34664,81	10105,39
F05	617,4222	412,0312
F06	126,46	91,38506
F07	2371,595	854,5443
F08	575,5944	374,6549
F09	30158	15082,67
F10	9173,545	8461,775
F11	30676,13	9459,415
F12	1,71E+10	7,25E+09
F13	1,91E+10	4E+09
F14	24736534	3677467
F15	3,53E+09	2,55E+08
F16	5455,94	3341,561
F17	5414,637	1661,669
F18	3,4E+08	55410886
F19	4,57E+09	3,94E+08
F20	1618,837	1200,438
F21	740,2196	617,4353
F22	9362,158	6460,223
F23	1406,427	1024,034
F24	1644,988	1086,312
F25	12502,41	2900,753
F26	12161,52	6772,138
F27	2088,027	1486,311
F28	8382,887	3758,208
F29	10888,46	3372,845
F30	2,55E+09	3,68E+08

	BA	DE	PSO
Best	0	4	26

3.2.2 Evaluaciaciones 100%

	BA	DE
Best	0	30
F01	6,72E+10	49094,72
F02	7,65E+41	1,31E+19
F03	145543	3481,079
F04	14525	84,30386
F05	491,0837	201,4981
F06	101,4875	6,320261
F07	1757,543	233,4222
F08	435,5023	189,372
F09	18291,86	65,2973
F10	7967,426	3763,561
F11	10213,83	79,5828
F12	8,83E+09	325765
F13	4,67E+09	153,6136
F14	2025127	71,00224
F15	5,1E+08	62,55673
F16	3605,785	1319,233
F17	1937,347	480,8806
F18	36131326	61,22245
F19	1,11E+09	35,72268
F20	1039,027	275,1128
F21	637,0859	325,4941
F22	6968,044	100,2235
F23	1092,145	534,6092
F24	1215,364	605,9018
F25	5992,591	387,028
F26	8109,208	403,7673
F27	1364,659	492,5198
F28	4522,745	394,2696
F29	3436,478	1025,095
F30	6E+08	3656,552

	BA	PSO
Best	0	30
F01	6,72E+10	4,18E+09
F02	7,65E+41	1
F03	145543	54534,28
F04	14525	1183,191
F05	491,0837	217,0438
F06	101,4875	36,93872
F07	1757,543	359,5821
F08	435,5023	174,5205
F09	18291,86	2841,846
F10	7967,426	6938,089
F11	10213,83	1207,135
F12	8,83E+09	3,59E+08
F13	4,67E+09	45075854
F14	2025127	305871
F15	5,1E+08	273710
F16	3605,785	1568,291
F17	1937,347	472,5476
F18	36131326	2170364
F19	1,11E+09	1260194
F20	1039,027	462,1451
F21	637,0859	411,2916
F22	6968,044	1026,573
F23	1092,145	640,4298
F24	1215,364	709,5056
F25	5992,591	686,3826
F26	8109,208	3369,357
F27	1364,659	806,8056
F28	4522,745	1105,036
F29	3436,478	1409,126
F30	6E+08	13585755

	BA	DE	PSO
Best	0	27	3

3.2.3 Comparativa Ranking

	BA	DE	PSO
1	2,433333	2,433333	1,133333
2	2,533333	2,333333	1,133333
3	2,7	2,1	1,2
5	3	1,266667	1,733333
10	3	1,133333	1,866667
20	3	1,1	1,9
30	3	1,066667	1,933333
40	3	1,066667	1,933333
50	3	1,066667	1,933333
60	3	1,066667	1,933333
70	3	1,1	1,9
80	3	1,1	1,9
90	3	1,1	1,9
100	3	1,1	1,9

3.2.4 Análisis de Resultados

Para llevar a cabo el análisis de resultados se han obtenido, al igual que para dimensión 10, un total de siete tablas para poder llevar a cabo comparativas con los algoritmos mencionados.

Se puede observar que en la tabla general con 1% de evaluaciones, el mejor con algoritmo es de nuevo el PSO frente al BA que es el que peores resultados ha obtenido también. Sin embargo, si pasamos a analizar las otras dos tablas por separado podemos observar de nuevo que los resultados obtenidos en BA son mejores que en DE, aunque esta vez el número de veces que lo ha superado es menor que en el de dimensión 10. No obstante, los resultados de BA son peores que los de PSO en todos los casos.

De esta forma, podemos llegar a la conclusión de que al comienzo, cuando hay un pequeño porcentaje de evaluaciones, los resultados obtenidos con la metaheurística BA son considerablemente aceptables al ser mejores que los obtenidos con DE, pero son peores que en dimensión 10.

Sin embargo, a medida que aumentan las evaluaciones podemos observar que nuestra metaheurística no va a ser tan buena. De hecho, ya en un 2% de evaluaciones los resultados obtenidos en DE superan ligeramente los de BA. Así pues, al igual que en dimensión 10 nuestra metaheurística podría tener demasiada exploración y necesitar mayor explotación para poder obtener mejores resultados..

4. Conclusión

A modo de conclusión, en vista de los análisis realizados anteriormente sobre las distintas dimensiones y porcentaje de evaluaciones de la metaheurísticas, podemos afirmar que la que mejores resultados nos ha proporcionado es PSO mientras que nuestro metaheurística BA se podría considerar de manera global la menos óptima.

Esto se puede deber a, como se ha comentado previamente, que no estamos aprovechando las evaluaciones del fitness a partir de un cierto porcentaje. Esto es apreciable ya que cuando el porcentaje de evaluaciones es bajo, los resultados que obtenemos no son los mejores pero si superan a los del DE, mientras que a medida que aumentan las diferencias entre BA y los otros dos algoritmos son notables.

Así pues es posible que estemos realizando demasiada exploración y desperdiciamos evaluaciones. Por lo que la solución al problema podría ser intensificar las soluciones para llegar a una mayor explotación y como consecuencia obtener unos mejores resultados. Una forma práctica de solucionarlo sería haciendo modificaciones sobre la búsqueda local, por ejemplo utilizando la que nos proporciona Daniel Molina denominada “Solis West”.