

Evolutionary Discrete Firefly Algorithm for Travelling Salesman Problem

Sergio Donís Ebri

Master en Inteligencia Artificial (MUIA), Universidad Politécnica de Madrid (UPM), España
s.donis@alumnos.upm.es

Abstract. Este paper contiene mi implementación del algoritmo de las luciérnagas, (Firefly Algorithm), FA, para resolver el problema del viajante (TSP, por sus siglas en inglés). Para ello se ha realizado una implementación del algoritmo propuesto por Gilang Kusuma Jati y Suyanto[1]. Para intentar optimizarlo, se ha intentado solucionar uno de los problemas de este papel y dar direccionalidad a las luciérnagas.

Keywords: Firefly algorithm, Travelling salesman problem, Evolutionary algorithm

1. Introducción

FA, propuesto por Xin-she Yang, [2] es normalmente usado para problemas continuos, por la dificultad que puede suponer darle direccionalidad a las luciérnagas e intentar acercar una solución a otra.. Ya hay varios intentos de resolver el TSP mediante algoritmos bio inspirados, como por ejemplo algoritmos meméticos, propuesto por Ref Luciana Buriol[3]; discrete fuzzy PSO[4], por N.Salmani Niasar; algoritmos genéticos combinado con un sistema de colonia de hormigas, por Marcian L. Pilat[5]. Todos estos algoritmos están basados en una combinación de metaheurísticas con búsqueda local.

2. Evolutionary Discrete Firefly Algorithm

2.1 Representación de una luciérnaga

Una representación de una solución posible se puede ver en la figura 1:

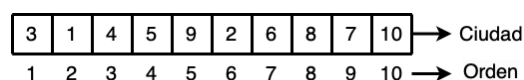


Figura 1. Representación de una solución

Hay que recordar que en el caso del TSP, después de visitar la última ciudad, hay que visitar también la primera otra vez.

2.3 Movimiento

Hay dos tipos de movimiento al comparar dos luciérnagas:

- La que brilla más(tienen una solución mejor), se mueve de forma aleatoria(movimiento más exploratorio).
- La que brilla menos(tiene una solución peor), se siente atraída por la otra que tiene una solución mejor (movimiento más explotativo).

3. Experimentos Realizados y Resultados

Se han diseñado tres experimentos, uno en el que se realiza solo movimientos aleatorios, otro en el que se realiza solo movimientos acercando las peores soluciones a las mejores y después el algoritmo ya propiamente dicho de las luciérnagas. Se ha usado para la codificación de este problema la librería TSPLIB y más en concreto el problema de berlin 52, consistente en 52 nodos con una distancia optima de 7542. Todos los experimentos se han realizado 5 veces y 8000 iteraciones cada una.

3.1 Movimiento Aleatorio

Se ha utilizado la mutación inversa: que consiste en elegir dos índices, de forma aleatoria e intercambiar todos los elementos de orden. Se ha usado esta mutación debido a que así no se destruyen posibles caminos buenos ya formados. En la figura 2, se puede ver el ejemplo de una mutación.

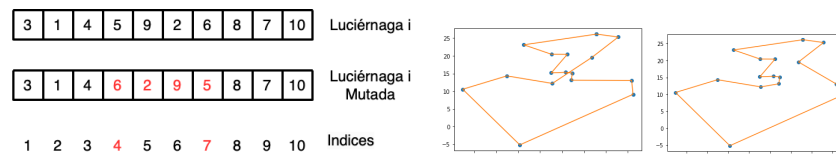


Figura 2. Representación de una mutación

Para ello se han realizado dos experimentos. Dado una población de 11 luciérnagas (caminos), cada uno muta 11 veces, produciendo 121 posibles soluciones. De esas nuevas soluciones mutadas, solo 11 pasan a la siguiente generación. Y se ha estudiado también cuántas de las luciérnagas antes de mutar pasan a la siguiente. Primero se ha estudiado la relación como de grande tienen que ser las mutaciones, que se realizan de un tamaño aleatorio entre $[2,12]$, $[2,25]$, $[2,52]$. La primera columna dice cuántas veces ha convergido el algoritmo. La segunda y la tercera, cuando el algoritmo converge en cuántas iteraciones y tiempo lo hace. Las siguientes tres, el mejor, peor y la media de los resultados. La última columna, a partir de que iteración ya no se mejora. Los resultados obtenidos son los siguientes:

	converge	iteraciones converger	tiempo converger	peor	media	mejor	iter ultima mejora
aleatorio 52	2/5	1758	36sec	7992	7782	7542	1871
aleatorio 25	1/5	1200	82sec	7892	7777	7542	1200
aleatorio 12	0/5	X	X	10507	9396	8622	2640

Figura 3. Tabla comparando las diferentes mutaciones

	Converge	media iter conver	tiempo iter conver	peor	media	mejor	iter última mejora
1	2/5	1758	36sec	7992	7710	7542	1871
3	2/5	1559	42sec	8029	7757	7542	1903
7	0/5	X	X	8195	7929	7735	2440
ninguno	0/5	X	X	7902	7874	7718	5630

Figura 4. Tabla comparando cuantas luciérnagas de la generación anterior pasan a la siguiente

Para el segundo experimento se han usado mutaciones de tamaño aleatorio entre 2 y 52. 2 de cada 5 veces se encuentra con este algoritmo el camino óptimo, y cuando no encuentra una solución bastante buena. Lo que demuestra que este algoritmo es capaz de encontrar buenas soluciones.

3.2 Movimiento con dirección

El proceso de acercar una luciérnaga a otra, se hace viendo los arcos que tienen diferentes, e intercambiando valores de estos arcos por los correctos, para que cada vez se vayan acercando más. El proceso de movimiento con direccionalidad, se puede ver con más profundidad y con un ejemplo en la siguiente figura 5.

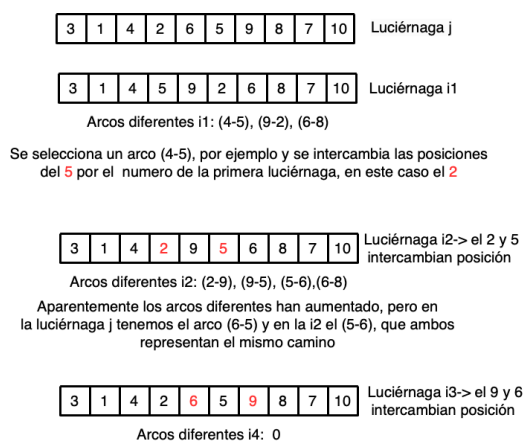


Figura 5. Movimiento con dirección

Para este experimento se han comparado todas las luciérnagas con todas. En este caso, la que tenía mejor solución no se ha mutado y a la que tenía una peor solución, se le acercaba a la otra. Al igual que en el experimento anterior, se han realizado diferentes pruebas para ver cuántas soluciones de la generación actual se guardaban también para la siguiente. Los resultados obtenidos se pueden ver en la figura:

	Peor	Media	Mejor	Iteración última mejora
solo 1	18266	16169	15061	56
solo 3	18534	16445	15035	380
solo 7	19567	17928	13386	376
ninguno	20093	18431	15688	445

Figura 6. Tabla de cuántas luciérnagas de la generación actual pasan a la siguiente con solo movimiento direccional.

Como vemos, con solo este algoritmo, no se es capaz de converger hacia una buena solución y se converge de forma bastante prematura, pero la combinación de ambas técnicas puede producir resultados satisfactorios.

3.3 Algoritmo de las luciérnagas

Ahora el algoritmo es el mismo que en el de las luciérnagas. Se comparan todas las luciérnagas con todas. La mejor adaptada se mueve de forma aleatoria y la peor adaptada se mueve de forma direccional hacia la otra. Como en todos los experimentos anteriores, se ha mostrado que es beneficiario quedarse con solo la mejor solución de la generación actual para la siguiente, aparte

de las nuevas; en este experimento solo se han realizado esas pruebas. Los resultados obtenidos han sido los siguientes.

	peor	media	mejor	media iter que ya no mejora
solo 1	8062	7951	7775	700

Figura 7. Resultados del algoritmo de las luciérnagas

El algoritmo no ha sido capaz de converger ninguna de las 5 veces que se ha probado, mientras que solo el movimiento aleatorio, ha convergido 2 de 5 veces.

3.4 Otros Mapas.

Vamos a estudiar también nuestro algoritmo en otros mapas, para ver cómo se comporta y cuantas veces converge.

Ulysses 16: converge 5 de 5 veces

Ulysses 22: converge 5 de 5 veces

Gr202: converge 0 de 5 veces: camino medio encontrado: 43132, lo que supone un 93% del óptimo

Pcb442 converge 0 de 5 veces, camino medio encontrado: 56640->89,65%, lo que mejora el algoritmo del paper que se estaba intentando seguir

3.5 Entrenamiento en Dos Fases.

Analizando cómo son los caminos obtenidos, normalmente convergen a caminos con soluciones bastante buenas, como por ejemplo el camino de la siguiente figura, con una distancia de 8086, acercando un 93% de la solución óptima. Tiene un camino bastante bueno, pero cambia bastante respecto al óptimo, Figura X.

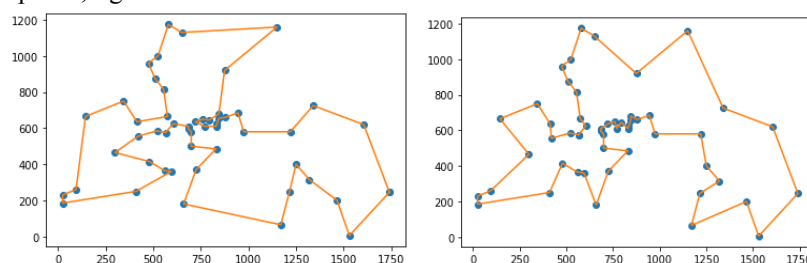


Figura 8. Óptimo local vs óptimo global

Una vez se ha llegado en un mínimo local como el anterior, es muy difícil salir de él. Analizando los caminos obtenidos y las funciones de pérdida del primer experimento, de cuando no se guardan las luciérnagas antes de mutar para la siguiente generación, el algoritmo es bastante exploratorio y los caminos tras 8000 iteraciones eran muy parecidos al óptimo. Se pueden ver en la siguiente figura:

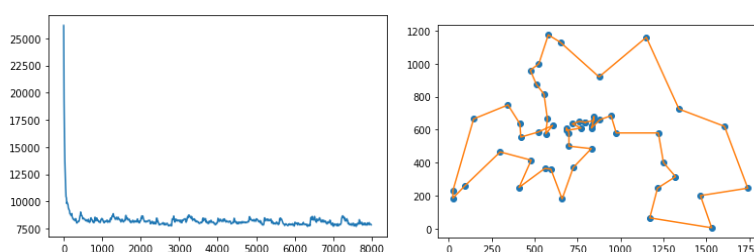


Figura 9. Gráfico de la mejor solución y camino del movimiento aleatorio cuando no se guardan las soluciones anteriores.

Los caminos obtenidos son muy parecidos al óptimo. Se ha hecho la prueba de después de eso, si ejecutar 2000 iteraciones más guardando la mejor solución de la generación actual para la siguiente y en ese caso 4 de 5 veces ha acabado convergiendo encontrando el mejor camino..

4 Conclusiones y Mejoras Futuras

4.1 Conclusiones

- El algoritmo propuesto de las luciérnagas, encuentra una muy buena solución a los problemas propuestos, en una distancia del 97% óptima para el mapa de berlin. Pero es propenso a caer en óptimos locales, dado que no ha convergido ninguna de las 5 veces en el mapa de Berlín 52, gr202 Pcb442.
- Es mejor algoritmo ha sido usando sólo mutaciones (ha convergido 2 de 5 veces), que el algoritmo de las luciérnagas.
- Es beneficioso tener dos fases, la primera más exploratoria y la segunda más explotativa. Con esto se ha conseguido encontrar 4 de cada 5 veces al camino óptimo.
- Este algoritmo ha encontrado, de media, una mejor solución al mapa de Pcb442, que el algoritmo de las luciérnagas del paper,, lo que quiere decir que este algoritmo puede resultar mejor que ese, dependiendo del tipo de mapa

4.2 Mejora futuras

- Tener diferentes tipos de luciérnagas, algunas más exploratorias y otras más explotativas en el mismo algoritmo.
- Implementar un tipo de mutación que sea la traslación. Por ejemplo, pasar de una posible solución que sea 1 2 3 4 5 6 7 8 9 a 1 2 3 7 8 9 4 5 6 . Este tipo de mutación también puede ser buena porque tampoco se rompen caminos buenos ya formados.
- Tener un entrenamiento en varias fases, al principio más exploratorio e ir cambiando hacia una búsqueda local.
- En el algoritmo de las mutaciones, al hacer las mutaciones, no hacer las 11 mutaciones sobre la luciérnaga original sino sobre la anterior mutación. Esto sería por ejemplo, en la iteración 1, en vez de coger la primera luciérnaga y mutarla 11 veces; mutarla, y volver a mutarla sobre esa mutación. Esto puede ser bastante bueno porque a lo mejor una sola mutación a la vez no es suficiente para salir del óptimo local.
- Ir reduciendo cada ciertas iteraciones el tamaño máximo de mutación. Es muy difícil que cuando el camino ya haya convergido, se necesite una mutación grande para encontrar el camino óptimo.

References

1. Gilang Kusuma Jati and Suyanto. 2011. Evolutionary discrete firefly algorithm for travelling salesman problem. In Proceedings of the Second international conference on Adaptive and intelligent systems (ICAIS'11). Springer-Verlag, Berlin, Heidelberg, 393–403.
2. X.-S. Yang, Firefly algorithms for multimodal optimization, in: Stochastic Algorithms: Foundations and Applications, SAGA 2009, Lecture Notes in Computer Sciences, Vol. 5792, pp. 169-178 (2009).
3. Luciana, B., França, P.M., Pablo, M.: A New Memetic Algorithm for the Asymmetric Traveling Salesman Problem. Journal of Heuristics 10(5), 483–506 (2004)
4. Niasar, N.S., Shanbezade, J., Perdam, M.M.: Discrete Fuzzy Particle Swarm Optimization for Solving Travelling Salesman Problem. In: Proceedings of International Conference on Information and Financial Engineering, pp. 162–165 (2009)
5. Pilat, M.L., White, T.: Using Genetic Algorithms to Optimize ACS-TSP. In: Dorigo, M., Di Caro, G.A., Sampels, M. (eds.) Ant Algorithms 2002. LNCS, vol. 2463, pp. 12–14. Springer, Heidelberg (2002)