

Aplicación del algoritmo Ant Colony Optimization (ACO) para resolver el Traveling Salesman Problem (TSP) ^[1]

Integrantes: Jorge Valenzuela, Jose Pablo Santander

Equipo docente: Joaquín Fontbona, Camilo Carvajal, Arie Wortsman, Pablo Zúñiga

En la naturaleza, las hormigas son capaces de aproximar empíricamente soluciones a problemas del estilo "caminos óptimos" a través del uso de feromonas, un tipo de sustancia química que estas pueden liberar a voluntad, con el fin de comunicar a sus compañeras instrucciones y advertencias.

El algoritmo ACO toma esta idea y la aplica a búsquedas en grafos a través de iteraciones en donde una traza de feromonas es dejada, con el fin de que la siguiente iteración obtenga información útil y logre mejorar la solución actual al problema de búsqueda. Debido a esto el problema del vendedor viajero resulta un entorno ideal para implementar el algoritmo.

El algoritmo (ACO) en su versión más básica llamada Ant System (AS) consta de los siguientes pasos:

- 1) Se inicializan aleatoriamente m hormigas de forma uniforme sobre los vértices del grafo que componen la instancia del TSP a resolver
- 2) Para cada hormiga se busca un camino hamiltoniano a través del grafo, en donde la probabilidad de transición de un vértice a otro está dada por la formula:

$$p_{i,j}^k = \frac{[\tau_{i,j}(t)]^\alpha \cdot [\theta_{i,j}(t)]^\beta}{\sum_{l \in N_t^k} [\tau_{i,l}(t)]^\alpha \cdot [\theta_{i,l}(t)]^\beta}$$

Donde $\tau_{i,j}(t)$ corresponde a la cantidad de feromona localizada en el arco i,j en el tiempo t . $\theta_{i,j} = \frac{1}{d_{i,j}}$ es un valor heurístico. α, β son dos parámetros que determinan la influencia relativa del rastro de hormonas y la información heurística, y N_t^k es el conjunto de los vecinos factibles del vértice i para la hormiga k .

- 3) Una vez obtenidos los caminos de cada hormiga, se realiza la actualización de los rastros de feromonas con la siguiente regla:

$$\tau_{i,j}(t+1) = (1-\rho) \cdot \tau_{i,j}(t) + \sum_{k=1}^m \Delta\tau_{i,j}^k(t)$$

Donde $0 \leq \rho \leq 1$ simboliza la evaporación del rastro de feromonas.

$\Delta\tau_{i,j}^k(t)$ es la cantidad de feromona que la hormiga k pone en los arcos que esta recorrió en su camino, y se define como sigue:

$$\Delta\tau_{i,j}^k(t) = \begin{cases} 1/L^k(t) & \text{si el arco (i,j) es usado por la hormiga k} \\ 0 & \text{en otro caso} \end{cases}$$

Donde $L^k(t)$ es el largo de del camino hamiltoniano correspondiente a la hormiga k

La implementación del algoritmo fue realizada en Python, y con el uso de un Jupyter Notebook se analizó el comportamiento de este para distintas configuraciones de parámetros, junto con la manera en que los parámetros se influyen entre sí. De esto se pudieron extraer las siguientes conclusiones:

- En general un buen número de hormigas es aproximadamente el numero de ciudades de la instancia de TSP
- Las combinaciones de α , β que dan buenos resultados, son con $2 \leq \alpha \leq 10$, $2 \leq \beta \leq 10$
- El parámetro de evaporación ρ funciona mejor con $0.5 \leq \rho \leq 0.9$
- El parámetro Q funciona bien en general para $1 \leq Q$

Además, se comparó el rendimiento de ACO contra el de Simulated Annealing para las mismas instancias, resultando en una clara ventaja de ACO, obteniendo caminos más cortos en un menor tiempo de ejecución

[1] St, Thomas & Dorigo, Marco. (1999). ACO Algorithms for the Traveling Salesman Problem.