



Diseño de tuberías mediante optimización por colonia de hormigas

Daniel Baeza
Carlos Gonzalez

11 de diciembre de 2013

1. Introducción

En este reporte se muestran los avances hechos en la implementación de un algoritmo basado en Optimización por Colonia de Hormigas (ACO) para el diseño de tuberías en topologías complejas.

Los problemas de optimización combinatorial son muy frecuentes en ingeniería y en el ámbito científico. Un ejemplo de estos problemas es el asociado al ruteo, en donde el objetivo puede ser encontrar la ruta mas corta tomando distintas restricciones y no necesariamente una métrica euclidiana.

Optimización por colonia de hormigas es una solución a problemas de optimización combinatorial desarrollada en base al comportamiento natural que tienen las hormigas dentro de una colonia al momento de buscar y llevar el alimento a sus hormigueros. Este método es introducido por Dorigo en (1996) y fue propuesto como una técnica de optimización para luego ser formalizada como una metaheurística. Ejemplos de otras metaheurísticas son *simulated annealing* y *tabu search*.

Este trabajo intenta dar una solución alternativa al problema de diseño de tuberías para el transporte de fluidos en faenas mineras usando un algoritmo evolutivo como lo es ACO y además tomando en cuenta costos y restricciones asociadas a la construcción, gastos operacionales, impacto ambiental, energía de transporte de los fluidos y factibilidad del terreno para la construcción de las tuberías.

2. Descripción del algoritmo

2.1. Aspectos generales del comportamiento de las Hormigas y su análogo computacional

El uso del comportamiento de las hormigas dentro de una colonia como base de un algoritmo de optimización tiene relación con el modelamiento de agentes computacionales simples trabajando de manera cooperativa para alcanzar un objetivo en común. Esto se lleva a cabo a través de un espacio artificial de feromonas en donde se registra el éxito de cada agente al buscar el alimento para la colonia. En este sentido, un agente se mueve desde una posición S_i a una posición S_j guiado por dos factores:

- Información heurística: Es una métrica que indica la preferencia que tiene una hormiga de pasar de un estado S_i a un estado S_j . Esta información es parte de la representación del problema que se quiere resolver y no es modificada durante la ejecución del algoritmo.
- Rastro de feromona: Es una medida de la depositación de feromona que han dejado el resto de las hormigas que han pasado sobre el arco que une los estados S_i y S_j . Este rastro se puede desgastar cuando pasa una hormiga que aún no ha tenido éxito en la búsqueda del alimento o puede ser aumentado por hormigas que han encontrado el alimento y van de vuelta al hormiguero.

Las principales propiedades del modelo de una hormiga son las siguientes:

- Cada hormiga tiene una memoria interna que registra el camino que ha recorrido.
- Todas las hormigas comienzan en el estado S_0 e intentan crear una solución factible para algún problema en particular, moviéndose iterativamente por el espacio de búsqueda.
- Antes de moverse desde S_i a un estado S_j , cada hormiga aplica las reglas de transición entre estados. Estas reglas pueden incluir restricciones adicionales y la información heurística relacionadas con el problema que se quiera resolver.
- La depositación de feromona que cada hormiga hace está descrita por la regla de actualización del modelo de feromona que se utilice. Esto involucra actualizaciones hechas en cada transición de estado (online) o una vez que el agente haya llegado al estado objetivo actualizar toda la trayectoria (offline).

Además de la heurística, es posible mejorar la convergencia del algoritmo usando actualizaciones desde una perspectiva global, castigando las malas soluciones y premiando las mejores soluciones propuestas por cada hormiga, esto es disminuyendo o aumentando el rastro de feromona en cada caso. Estas funciones son ajenas a la heurística ya que no tienen una contraparte en la naturaleza, pero mejoran los tiempos y la convergencia del algoritmo en algunos problemas. Estas reglas son llamadas *Daemon Actions*.

2.2. Información heurística - Función de costos

La información heurística del modelo es todo lo que caracteriza el problema que se quiere resolver. En el caso de que se quiera encontrar el camino mas corto entre dos puntos, la información heurística es la referida a la distancia entre dos estados adyacentes y la factibilidad de ir desde un estado a otro. Por ejemplo, si se quiere encontrar el camino mas corto entre dos puntos en una ciudad se usa la métrica euclidiana y la factibilidad de ir de un estado a otro queda dado por la presencia de un camino entre esos dos estados.

2.3. Metaheurística

La metaheurística del método de optimización por colonia de hormigas está constituida por las siguientes etapas:

- Inicialización de parámetros: En esta etapa se carga la información heurística del problema, el estado inicial y el estado objetivo dentro del grafo. Por otro lado, se escogen las reglas de actualización de feromona y cuales serán las *Deamon Actions* usadas en la implementación.
- Construcción de soluciones: Mediante las reglas de transición entre estados y el rastro de feromonas dejado por el resto de la colona, cada hormiga genera soluciones factibles dentro del espacio de solucines, con el fin de unir los estados inicial y final mediante una secuencia de estados transitorios.
- Actualización de rastro de feromonas: luego de la transición de un estado S_i a un estado S_j , el rastro de feromona del arco que une estos dos estados es actualizado, aumentando o disminuyendo el rastro dependiendo del éxito que tuvo la hormiga que pasó por este arco.
- *Deamon Actions* (opcional): Esta es una etapa opcional, pero muy usada para mejorar los tiempos y la convergencia del algoritmo. El método mas usado en esta etapa es el de reforzar (añadir un extra de feromona) el rastro de la mejor ruta alcanzada hasta el momento.

Estas son las 4 etapas principales del método optimización por colonia de hormigas. La construcción de soluciones, la actualización del rastro de feromonas y las *Deamon Actions* están dentro del ciclo principal que se ejecuta hasta que se cumpla el criterio de detención definido por el usuario.

2.4. Ant Colony System

En la Heurística implementada hay dos tipos de actualizaciones sobre el rastro de feromona que hay en un arco, que une dos estados. La primera actualización corresponde a la depositación de feromona en un arco por parte de la hormiga cuando va de regreso al hormiguero después de haber encontrado la comida, que está dada por la siguiente ecuación.

$$\tau_{ij} \leftarrow (1 - \rho) \cdot \tau_{ij} + Ang^k(t_{\text{to the nest}}) \cdot \Delta\tau_{ij}^k \quad (1)$$

Esta actualización está compuesta por un factor de evaporación ρ sobre la feromona existente en el arco, un factor de agotamiento $Ang^k(t_{\text{to the nest}})$ de la hormiga y la cantidad de feromona que deposita en el arco $\Delta\tau_{ij}^k$. El factor de evaporación corresponde a un parámetro que evita la saturación del rastro de feromona en cada arco, haciendo que por cada depositación de feromona que haga una hormiga la cantidad de feromona que ya existía decrezca en una fracción ρ .

El factor de agotamiento $Ang^k(t_{\text{to the nest}})$, una vez que la hormiga encontró la comida, hace que la hormiga deposite menos feromona por arco a medida que se aleja del estado objetivo. Dado que todas las hormigas parten del hormiguero, mantener un rastro de feromona en las cercanías de ese estado puede llevar a estancamientos en óptimos locales. El factor de agotamiento intenta construir la solución desde el estado objetivo hacia el estado inicial. Por último está la depositación de feromona por cada hormiga. Dado un factor Q , que es la cantidad de feromona que una hormiga es capaz de depositar en un arco y el costo L_k de la ruta que la hormiga k hizo entre el hormiguero y la comida, la cantidad de feromona que la hormiga deposita en un arco entre los estados S_i y S_j es $\Delta\tau_{ij}^k$ y está dada por la siguiente ecuación.

$$\Delta\tau_{ij}^k = \begin{cases} Q/L_k & \text{si la hormiga } k \text{ usa el arco } (i, j) \text{ de regreso} \\ 0 & \text{si la hormiga } k \text{ no usa el arco } (i, j) \end{cases} \quad (2)$$

En el factor de depositación de feromona $\Delta\tau_{ij}^k$ es donde las mejores rutas son premiadas y las peores castigadas, disminuyendo el deposito de feromona a medida que el costo de la ruta L_k aumenta.

La segunda forma de actualización de feromona por arco tiene relación con el decaimiento de la intensidad de la huella. El decaimiento de la huella se debe a hormigas que estén en la búsqueda del estado objetivo. Ya que estas hormigas no depositan feromona en los arcos, el factor de decaimiento ayuda a impedir el estancamiento en óptimos locales y deshechar soluciones o rutas con costos muy altos. Esta actualización se expresa a través de la siguiente ecuación,

$$\tau_{ij} \leftarrow (1 - \varphi) \cdot \tau_{ij} + \varphi \cdot \tau_0 \quad (3)$$

En donde φ es el factor de decaimiento y τ_0 es el rastro de feromona inicial de cada arco en el espacio de búsqueda.

Un agente se mueve desde una posición S_i a una posición S_j guiado por una combinación entre las características propias de un problema particular (información heurística) y el rastro de feromona. Esta combinación se hace ponderando estos dos factores en cada arco adyacente al estado S_i . Finalmente se calcula la probabilidad de ir desde una posición S_i a una posición S_j en relación a todas las posiciones factibles con la siguiente ecuación

$$P_{ij}^k = \begin{cases} \frac{\tau_{ij}^\alpha \cdot \eta_{ij}^\beta}{\sum_{k \text{ factibles de } i} \tau_{ik}^\alpha \cdot \eta_{ik}^\beta} & \text{si } (i, j) \text{ es un arco factible} \\ 0 & \text{si } (i, j) \text{ no es un arco factible} \end{cases} \quad (4)$$

en donde τ_{ij}^α corresponde al rastro de feromona del arco que une el estado S_i y el estado S_j , y η_{ij}^β que se relaciona con la información heurística del problema (distancia entre el estado S_i y el estado S_j , factibilidad, etc). Los factores α y β es la importancia que se le da a cada factor en la ponderación, por lo general son $\alpha = 1$ y $\beta = 2$. En el caso del problema del camino mas corto, η_{ij} es 1 sobre la distancia entre los dos estados que en el caso mas simple puede ser la distancia euclidiana.

Por último, para mejorar la convergencia del algoritmo se introduce una variable proporcional pseudoaleatoria. Esta variable q_0 favorece las trayectorias con el mayor rastro de feromona. Sea $q_0 \in [0, 1]$ un parámetro, y q una variable aleatoria entre $[0, 1]$, entonces la elección del nodo j estando en el nodo i está dado por la siguiente distribución de probabilidad:

si $q \leq q_0$:

$$P_{ij}^k = \begin{cases} 1 & \text{si } j = \text{argmax}(\tau_{ij}^\alpha \cdot \eta_{ij}^\beta) \text{ y } j \text{ es un arco factible} \\ 0 & \text{en cualquier otro caso} \end{cases} \quad (5)$$

3. Implementación ACO-ALGES para el diseño de tuberías

Se ha diseñado una aplicación que implementa ACO para diseñar el trazado de tuberías. Los agentes de la simulación se utilizan para generar rutas dentro de un espacio que representa la topografía de cierto lugar modelada por una superficie. Estas rutas conectan un determinado punto de inicio y final, acercándose al óptimo, según las restricciones impuestas por la función de costos.

En específico, el algoritmo se implementa en Python, como un método iterativo cuya solución evoluciona en el tiempo hasta converger al óptimo buscado. Debido a las propiedades heurísticas del método se requiere aplicar técnicas avanzadas de visualización de datos para obtener los resultados y controlar su funcionamiento. Por este motivo, se ha integrado el programa con el *framework* Giulietta (desarrollo reciente del laboratorio ALGES), que provee un entorno de ejecución y obtención de datos permitiendo la correcta separación del problema de modelamiento y visualización como se observa en la figura 1. Esto permitirá, en el futuro, la ejecución distribuida del algoritmo y la visualización de resultados vía web.

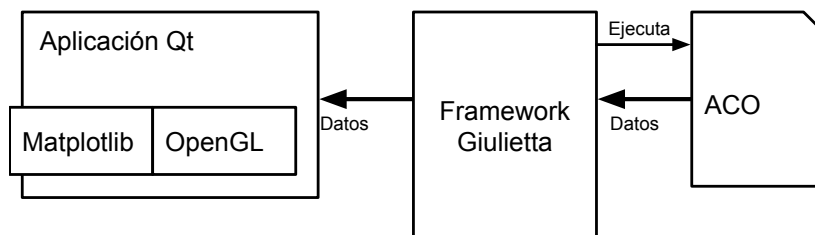


Figura 1: Implementación de la aplicación

La visualización de resultados se encapsula en la aplicación de *frontend* y su interfaz se programa utilizando Qt junto a Python y se divide en los siguientes componentes visuales:

1. Evolución del algoritmo: se utiliza OpenGL para obtener una visualización en tres dimensiones del comportamiento de la colonia simulada en tiempo real.
2. Mapa de feromonas: se utiliza una representación en dos dimensiones con la evolución del rastro feromonas en tiempo real. Mediante un mapa de calor, se representa la cantidad de feromona en cada nodo como el promedio del rastro disponible en todos los arcos que lo conectan.
3. Resultados: el costo de la mejor solución encontrada así como su evolución en el tiempo, se visualiza a través de un gráfico costo vs. tiempo que se actualiza a medida que se encuentran nuevas soluciones. El despliegue del diseño de la ruta para la tubería hace uso de OpenGL para visualizar la solución obtenida sobre la topografía objetivo.

El resultado final de la aplicación se muestra en la figura 2. La esquina superior izquierda muestra la evolución del algoritmo, la esquina superior derecha muestra el mapa de feromonas y en la parte inferior de la aplicación se muestra gráfico de la mejor ruta encontrada en función del número de iteraciones actuales.

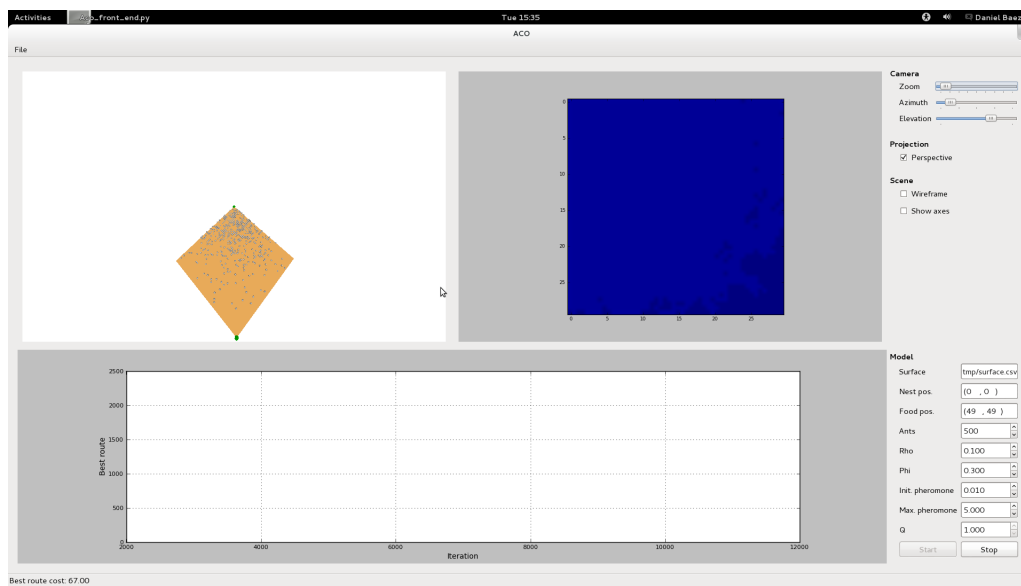


Figura 2: Aplicación PIPE-ACO

4. Resultados

Las figuras 3 y 4 muestran dos simulaciones hechas con PIPE-ACO, la aplicación hecha en ALGES para el diseño de tuberías usando Optimización por colonia de hormigas.

En la figura 3 se muestra el comportamiento de la aplicación frente a una superficie plana y con el nodo inicial u hormiguero ubicado en la esquina superior y el nodo objetivo ubicado en la superficie inferior. La función de costos es la norma euclidiana entre dos nodos adyacentes de la superficie. En este caso la ruta óptima está dada por la línea recta que une ambos nodos. Las tres visualizaciones muestran la dinámica de la simulación y a través de esto se puede observar la convergencia del algoritmo a esta ruta.

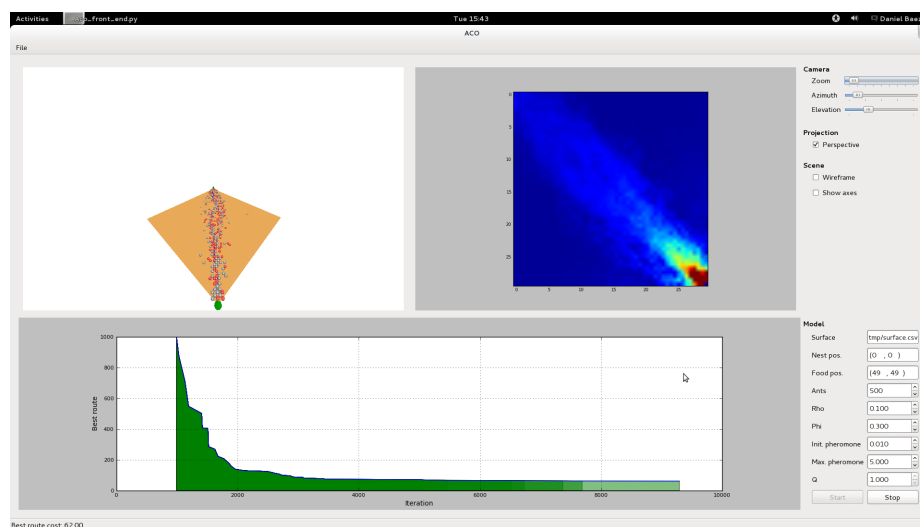


Figura 3: Aplicación PIPE-ACO

En la figura 4 se muestra el comportamiento de la aplicación en una superficie un poco mas compleja. La superficie en este caso posee una especie de laberinto con caminos sutilmente distintos. En este caso la colonia de hormigas también converge a la mejor ruta.

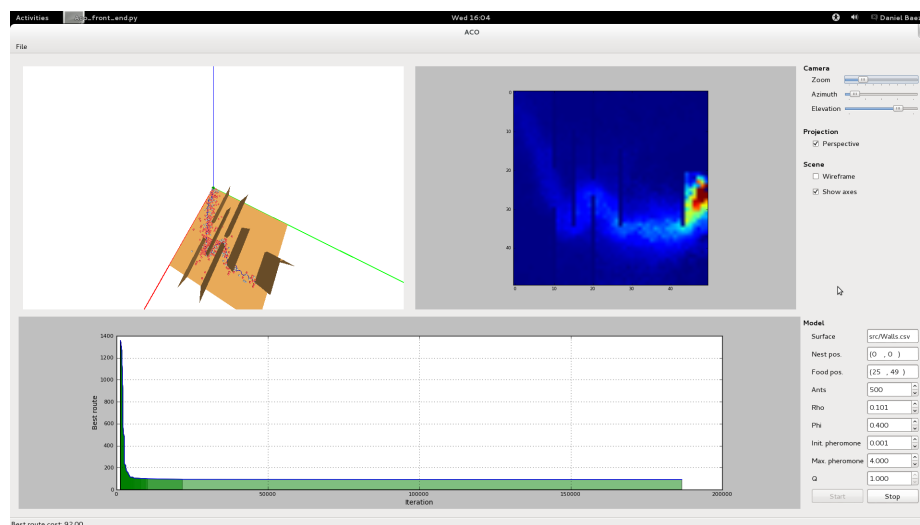


Figura 4: Aplicación PIPE-ACO

5. Conclusiones y trabajo futuro

- Los resultados iniciales de la implementación del algoritmo optimización por colonias de hormigas orientado al diseño de tuberías en topologías complejas son alentadores, convergiendo a las rutas esperadas en superficies ideales y laberintos.
- La aplicación PIPE-ACO hecha en ALGES es una herramienta muy útil para entender el comportamiento del algoritmo, dando una visualización de la colonia, el mapa de feromonas y la evolución de la mejor solución encontrada.
- Todas estas pruebas han sido hechas tomando una función de costos clásica, midiendo la distancia euclidiana entre dos nodos adyacentes. Es necesario probar una métrica hecha especialmente para el problema de diseño de tuberías, tomando en cuenta costos económicos, impacto ambiental, ángulos de elevación, etc.
- Como parte del trabajo futuro está implementando la métrica diseñada para medir los costos de las tuberías y complejizar cada vez más los escenarios de búsqueda, para llegar finalmente a casos reales.