



# Investigación de operaciones

## Optimización utilizando colonias de hormigas

Erik Regla, Daniela Paredes, Cristobal Henriquez  
eregla09@alumnos.utalca.cl, dparedes09@alumnos.utalca.cl,  
chenriquez09@alumnos.utalca.cl

28 de Junio del 2015

### 1. Resumen ejecutivo

Se presenta una introducción a los sistemas de optimización utilizando colonias de hormigas por medio del *El problema del estudiante viajero* como ejemplo práctico, a modo de presentar las ventajas, desventajas y orígenes de esta metaheurística junto con una explicación conceptual de esta técnica a modo de no condicionar al lector además de proporcionar una base comparativa frente a soluciones deterministas para el ejemplo a modo de motivación.

### 2. El problema del *estudiante viajero*<sup>1</sup>

Sea  $G(V, E)$  un grafo dirigido con pesos  $W(E_1, E_2)$ . Se desea saber un recorrido entre dos nodos  $t(u, v)$  y  $t(v, u)$  de modo que  $t(u, v) \cap t(v, u) = \emptyset$  en que  $w(t(u, v)) + w(t(v, u))$  sea mínimo.

#### 2.1. Acercamientos deterministas para obtener la solución óptima

Usualmente al resolver un problema computacionalmente, nos interesa obtener la solución óptima de manera determinista. A continuación, se presentan algunos acercamientos típicos bajo esta premisa.

---

<sup>1</sup>Un estudiante quiere salir a colación y probar cada completo disponible en el camino. Pero no quiere pasar dos veces por el mismo camino para que después sus compañeros no lo traten de glotón. El viaje debe de ser ida y vuelta porque tiene clases al bloque siguiente.

### 2.1.1. Acercamiento 1: Algoritmo de Dijkstra <sup>2</sup>

Una forma rápida de encontrar el camino más corto entre dos nodos de un grafo es utilizar el *Algoritmo de Dijkstra* sobre los vértices  $u$  y  $v$ , el cual ofrece una complejidad  $O(|E| + |V| \log(|V|))$  en el peor caso asumiendo que  $w(u, v) \geq 0, \forall w(u, v) \in E, \forall u, v \in V$ . Sin embargo el problema viene en que este algoritmo solo es capaz de computar el primer viaje  $T(u, v)$ , pero no necesariamente es capaz de garantizar la existencia de un viaje  $T(v, u)$  luego de ser ejecutado. La opción entonces sería utilizar *backtracking* para generar iterativamente todos los  $T(u, v)$  posibles y en base a esta solución generar circuitos  $T(v, u)$  prometedores para luego elegir el óptimo. Sin embargo, esta alternativa es  $\omega((|E| \times |V|)^2)$ , lo cual lo vuelve una opción inviable para este problema.

### 2.1.2. Acercamiento 2: Programación dinámica

Conceptualmente, es posible aplicar la búsqueda de un camino corto entre dos nodos para resolver el problema fácilmente si  $u$  y  $v$  son adyacentes calculando  $\min(w(u, v))$  y  $\min(w(v, u))$ . Si  $x \in V, \exists e(u, x), \exists e(x, v)$  y  $x' \in V, \exists e(x, u), \exists e(v, x)$  entonces es posible calcular  $T(u, v)$  mínimo utilizando  $T(u, x), T(v, x), T(v, u)$  y  $T(u, x)$  pero si y solo si  $x \neq x'$ . Este concepto es constructivamente extensible  $x$  y  $x'$  por definición, con lo cual ahora hemos cambiado el problema de generar todas las rutas de ida y vuelta posibles para ahora encontrar  $x$  y  $x'$  prometedores y con ellos luego entregar el la solución óptima.

## 3. Algoritmos aproximados y aleatorios

Está claro que si nos apegamos a una solución determinista para obtener la solución óptima al problema estamos obligados a generar y evaluar todas las posibilidades, lo cual presenta una alta complejidad de tiempo y espacio. En la literatura como también en la naturaleza se puede observar innumerables ejemplos en donde algoritmos que introducen aleatoriedad en los datos o el algoritmo mismo presentan soluciones rápidas con un error mínimo a problemas como por ejemplo el cálculo de  $\pi$  utilizando un algoritmo *MonteCarlo* o el como las hormigas encuentran comida y devoran rápidamente un cadaver en la selva.

### 3.1. Heurísticas

Una herurística es un conjunto de instrucciones que resuelve un problema. A diferencia de la definición de algoritmo, esta carece de la parte “*conjunto de pasos finitos*” o “*y determinista*” por la sencilla razón que tan solo resuelve el problema. No hay garantías que la solución sea óptima. Desde un punto de vista filosófico los animales son la aplicación misma de este concepto, a ellos no les interesa llevar una vida ostentosa y llena de lujos, tan solo les interesa vivir y no requieren una receta finita y determinista para lograrlo.

---

<sup>2</sup>Se asume que el lector posee un nivel mínimo de conocimiento sobre algoritmos para recorridos de grafos y crecimiento de funciones.

---

**Algoritmo 1** Algoritmo de un estudiante promedio

---

```
1: mientras está vivo hacer
2:   despertar()
3:   espabilar()
4:   ducharse()
5:   desayunar()
6:   viajar_a_la_u()
7:   para todo bloque i en bloques_del_día hacer
8:     si es tiempo libre entonces
9:       si tiene hambre entonces
10:        ir_a_buscar_algo_de_comida()
11:       si no
12:         si motivado entonces
13:           estudiar()
14:         si no
15:           sacar_la_vuelta()
16:       fin si
17:     fin si
18:   si no
19:     si la clase es aburrida entonces
20:       sacar_la_vuelta()
21:     si no
22:       prestar_atención()
23:     fin si
24:   fin si
25: fin para
26:   regresar_a_casa()
27:   dormir()
28: fin mientras
```

---

Por ejemplo, incluso si quisiéramos describir a grandes rasgos el algoritmo que describe el comportamiento de un estudiante universitario promedio (Algoritmo 1), necesitamos muchas instrucciones para poder llegar a una idea básica de este. Esto es gracias a que la dimensión intrínseca del problema es demasiado alta, hay demasiadas variables a controlar, por lo cual generar un modelo genérico para poder describirlo solo termina en un fracaso.

---

**Algoritmo 2** Algoritmo de un perro promedio

---

```
1: mientras está vivo hacer
2:   si tiene hambre entonces
3:     ir_a_buscar_algo_de_comida()
4:   fin si
5:   si está aburrido entonces
6:     perseguir_ruedas()
7:     espabilar()
8:   fin si
9:   si está cansado entonces
10:    dormir()
11:  fin si
12: fin mientras
```

---

Un caso diferente ocurre si nos dedicamos a estudiar el comportamiento de un perro promedio, es fácil llevar el modelo a cualquier tipo de perro, ya sea domesticado o salvaje por la simple razón de que todos se comportan igual. Factores como la edad, raza, origen y si su dueño los peinó o no poco interesan o afectan el resultado. Pero a final de cuentas, tanto el perro como el estudiante son seres vivos, si quisiéramos resolver el problema de darle vida a un planeta dejar una colonia de perros o una colonia de humanos es una solución, pero si quisiéramos vida inteligente entonces estamos obligados a ir por el camino difícil. Gracias a esto, podemos hacer una analogía entre heurística y la vida del perro, para simplificar los términos.

### 3.2. Caso de estudio: Comportamiento de hormigas en la naturaleza

En las colonias de hormigas muchas de estas tienen distintos roles, pero nuestro interés cae sobre las hormigas obreras que se dedican a labores de exploración y alimentación de la colonia. Tomemos por ejemplo una colonia cualquiera, en la cual salen miles de hormigas por la entrada de esta vagando aleatoriamente por un terreno. Se sabe de algunas hormigas que utilizan visión y memoria como método de orientación, pero también utilizan el apoyo de feromonas para dejar un *rastros de galletas* desde el hormiguero a su posición actual el cual también es utilizado por otras hormigas para influenciar su decisión de tomar un camino u otro.<sup>3</sup> Asumamos entonces, que todas las hormigas poseen feromonas diferentes para indicar si están en un viaje de exploración o ya están regresando de una exploración exitosa. Las feromonas se evaporan al cabo de un tiempo dado, si esta encuentra entonces podemos considerar tres casos:

---

<sup>3</sup> Algo similar ocurre con el turismo. Al momento de elegir un buen lugar para comer uno tiende a pensar que si el lugar es bien concurrido, entonces es bueno, lo cual influencia nuestra decisión.

1. La hormiga fue devorada por algún animal. En este caso, su rastro de feromonas al cabo de un tiempo se va a evaporar evitando que otra hormiga la encuentre y siga el mismo camino, por ende ser devorada.
2. La hormiga encontró comida y se está regresando. En este caso, la hormiga necesita volver por el camino donde volvió y para prevenir que las feromonas que la llevan a destino se evaporen, escribe un nuevo rastro. Dado que la hormiga tiene un pequeño rango de visión, aprovecha de acortar el camino cuando se da cuenta que puede acortar la distancia entre su posición actual al siguiente tramo de feromona más cercano. Un camino más corto implica que la feromona va a ser más fuerte al momento en que la hormiga se regrese a ese tramo en comparación a un camino largo.
3. La hormiga efectivamente encontró comida pero fue devorada. Si bien alcanzó a marcar el tramo final con feromona extra, el primer trazo en evaporarse es el el primero que hizo, por ende la conexión a la colonia se pierde y evita que futuras hormigas sean guiadas allá.

Si el caso que ocurrió efectivamente fue el dos (el cual provee solución al problema de la colonia de buscar alimento), la hormiga al regresar a la colonia y entrar por esta hará que su rastro de feromona sea más intenso por el mero hecho de ser más reciente, lo cual influencia a otras hormigas a seguir ese camino por ser atractivo para ellas y gracias a esto, este rastro se intensifica progresivamente a medida que aumenta el número de hormigas circulando por este tramo. De hecho, el que las feromonas se evaporen también es útil, ya que ayuda a prevenir el primer y último caso.

Un fenómeno interesante de observar es cuando las hormigas ya habiendo encontrado comida y un buen número de ellas está en el camino, dos hormigas chocan porque no pueden estar en el mismo espacio. Una de estas tiene que ceder lo cual crea dos rastros de feromonas diferentes, dividiendo el camino en dos. Cuando dos hormigas lleguen a esta bifurcación, elegirán con una alta probabilidad el camino que presente la mayor concentración de feromonas (el cual tiende a corresponder al camino más directo). Al repetirse este proceso una y otra vez, lo que antes era un ruta aleatoria continuamente se va transformando en una ruta directa entre el alimento y la colonia.

### **3.3. Optimización de recorridos en grafos utilizando algoritmos inspirados en colonias de hormigas**

Del caso de estudio anterior, podemos notar que las hormigas circulan en un espacio y dan solución a mejorar el recorrido entre el alimento y la colonia. El comportamiento de las hormigas resulta ser bastante efectivo cuando el problema viene dado por la información topológica de este.

El problema anterior del *estudiante viajero* es un problema que viene dado por la información topológica en forma de lugares por los cuales pasar y el costo que tiene. En el Algoritmo 3 es posible observar que se requiere de alguna condición de término. El problema de esta solución es que dado que no se sabe cual es la solución óptima, tampoco se sabe con exactitud cuando detener el progreso de este.

---

**Algoritmo 3** Algoritmo para comportamiento de una colonia

---

```
1: mientras condición de término no cumplida hacer  
2:   moverHormigas()  
3:   evaporarFerormonas()  
4: fin mientras
```

---

---

**Algoritmo 4** Algoritmo para comportamiento de una hormiga

---

```
1:  $w \leftarrow \text{evaluarSolucionesLocales}()$   
2:  $w' \leftarrow \text{elegirSolucionesPrometedoras}(w)$   
3: moverA( $w'$ )  
4: depositarFeromona( $w'$ )
```

---

### 3.3.1. Resolución del problema del estudiante viajero utilizando colonias de hormigas

Siguiendo la misma lógica desarrollada en el ejemplo anterior, extendemos el comportamiento descrito en el Algoritmo 4 para resolver el problema inicial. Esto no es necesario para el comportamiento de la colonia, dado que en este caso, no afecta a la solución.

En primera instancia la hormiga verifica si es que ya cumplió con la construcción del circuito, vale decir, si ya desarrolló una ruta  $T$  de ida y vuelta ente  $u$  y  $v$  sin pasar dos veces por el mismo vértice. Luego, en base a la visibilidad sobre la arista actual obtiene todas las aristas que no haya visitado y de estas evalúa una a una la intensidad de las feromonas en la arista. Las señales de feromonas son algo global a todas las hormigas, información compartida entre todas las hormigas que componen la colonia. Esta también tiene cierto grado de aleatoriedad, por tanto no necesariamente la que presente la señal más grande es la que será elegida como siguiente destino, tan solo quiere decir que tiene una probabilidad mas alta de ser elegida que el resto. Una vez calculada la ponderación final de cada arista se elige la más alta y se agrega a la ruta actual provocando que la hormiga avance.

Puede ocurrir en el Algoritmo 5 que no existan aristas disponibles para visitar en el momento, entonces, la hormiga se regresa y marca la arista actual con una feromona negativa, de modo de indicar que para la ruta actual, esa arista crea un camino cerrado. También puede ocurrir que si bien hay aristas disponibles, ninguna de estas le parezca una buena opción a la hormiga, en cuyo caso se regresa y deja una marca que para esa ruta esa arista no es prometedora. Solo si es prometedora la ruta deja una feromona atractiva junto con la información de la ruta.

### 3.4. Variantes y mejoras a algoritmos basados en colonias de hormigas para otros problemas

- *ASrank*: Agrega información de la ruta actual hecha por cada hormiga a cada rastro de feromona que deja esta a modo de poder clasificar las soluciones.
- *Lógica difusa*: En ciertos casos, permite acelerar la búsqueda.
- *Max-Min (MMAS)*: Solo la mejor iteración deposita feromonas. En caso de producirse un

---

**Algoritmo 5** Algoritmo para comportamiento de una hormiga para resolver el problema del estudiante viajero

---

**Precondición:**  $G(V, E)$  un grafo,  $W(u, v)$  son los pesos de  $G$ ,  $T$  es un arreglo global de circuitos,  $T'$  es la ruta actualmente computada,  $o$  es el vértice origen,  $a$  es el vértice actual,  $d$  es el vértice destino

**Postcondición:**  $T$  contiene las rutas calculadas

```
1: si  $T'$  es circuito entonces
2:   agregar  $T'$  a  $T$ 
3:    $T' \leftarrow \emptyset$ 
4:   agregar  $u$  a  $T'$ 
5: fin si
6:  $p \leftarrow \emptyset$ 
7: para todo  $v$  en  $V$  que construya  $E(a, v)$  hacer
8:   si  $v \notin T$  entonces
9:     agregar  $v$  a  $p$ 
10:  fin si
11: fin para
12: si  $p$  no contiene aristas entonces
13:   marcarFeromonaNegativamente( $a$ )
14:   regresarHormiga()
15: si no
16:    $p \leftarrow evaluarFeromonas(p)$ 
17:   si  $p$  no contiene aristas prometedoras entonces
18:      $p' \leftarrow extraerMnimo(p)$ 
19:     agregar  $p'$  a  $T$ 
20:   si no
21:     marcarFeromonaNegativamente( $a$ )
22:     regresarHormiga()
23:   fin si
24: fin si
```

---

estancamiento, todas las feromonas son reinicializadas con el valor máximo.

- *Sistema elitista*: La mejor solución global deposita feromonas automáticamente junto con las otras hormigas en cada iteración. De esta manera, se asegura que la mejor solución en el momento persista.

## 4. Conclusiones

### 4.1. Ventajas del uso de optimizaciones basadas en colonias de hormigas

- Dado que las feromonas se van evaporando, evitan la congestión en óptimos locales.
- Por lo general, cuando el problema viene dado por su topología, implementar colonias de hormigas es más sencillo que implementar una solución determinista.
- Dependiendo del problema, la complejidad espacial puede ser mucho más baja que la requerida por un algoritmo determinista.

### 4.2. Desventajas del uso de optimizaciones basadas en colonias de hormigas

- Al ser un algoritmo *Montecarlo*, su complejidad temporal no está definida.
- La solución que se obtiene con esta metaheurística no necesariamente es la óptima y podría no converger a ninguna solución.

En general, se puede afirmar que los sistemas de optimización basados en colonias de hormigas proveen de una solución aproximada fácil de implementar en caso de que el problema pueda ser transformado a alguno similar al de recorridos de grafos con una solución descrita directamente por la información topológica de este. Además como se pudo observar, a diferencia de una solución determinista, las metaheurísticas detrás del comportamiento de colonias de hormigas para ser comprendidas no requiere apegarse estrictamente a una definición formal y es fácil de comprender, lo cual es una ventaja al momento de resolver problemas complejos.