

# **Aplicación del Algoritmo de Colonia de Hormigas para la optimización del transporte público**

***Claudia Rodríguez Lemus***

Instituto Tecnológico de Roque  
*claulemus@itroque.edu.mx*

***Alejandro Guzmán Zazueta***

Instituto Tecnológico de Roque  
*claulemus@itroque.edu.mx*

***Jorge Alejandro Hernández del Razo***

Instituto Tecnológico de Roque  
*claulemus@itroque.edu.mx*

***Erick Jhonatan Sánchez Martínez***

Instituto Tecnológico de Roque  
*claulemus@itroque.edu.mx*

***Francisco Gutiérrez Vera***

Instituto Tecnológico de Celaya  
*francisco.gutierrez@itcelaya.edu.mx*

## **Resumen**

El uso de la minería de datos y cómputo evolutivo en el desarrollo de los sistemas actuales, permite resolver problemas complejos como se muestra en el presente trabajo, en donde se expone el desarrollo de un prototipo a través del uso del algoritmo de Colonia de Hormigas para la optimización del Transporte Público en la ciudad de León, Guanajuato.

Una de las principales preocupaciones para el gobierno municipal de la Ciudad de León Guanajuato, es el de mejorar su sistema de transporte público. Desde hace 15 años, este municipio ha puesto en marcha diferentes programas que han ayudado a contar

con mejores servicios de transporte para sus usuarios y a su vez, contribuir al desarrollo sustentable de la región (Moreno, 2009).

**Palabra(s) Clave(s):** Algoritmo de Colonia de Hormigas.

## 1. Introducción

El prototipo para la Optimización del Servicio de Transporte Público de la Ciudad de León Guanajuato, busca contribuir con el gobierno municipal aportando tecnología de vanguardia, interactuando con los usuarios del servicio, proponiéndoles alternativas de rutas y transportes para llegar a sus destinos, así como estimaciones de tiempos de llegada. El desarrollo del prototipo se basó en la minería de datos, que es el proceso de detectar la información procesable de los conjuntos grandes de datos. Utiliza el análisis matemático para deducir los patrones y tendencias que existen en los datos. Normalmente, estos patrones no se pueden detectar mediante la exploración tradicional de los datos porque las relaciones son demasiado complejas o porque hay demasiado datos (Microsoft, 2008).

La generación de un modelo de minería de datos forma parte de un proceso mayor que incluye desde la formulación de preguntas acerca de los datos y la creación de un modelo para responder dichas cuestiones, hasta la implementación del modelo en un entorno de trabajo (ver figura 1).

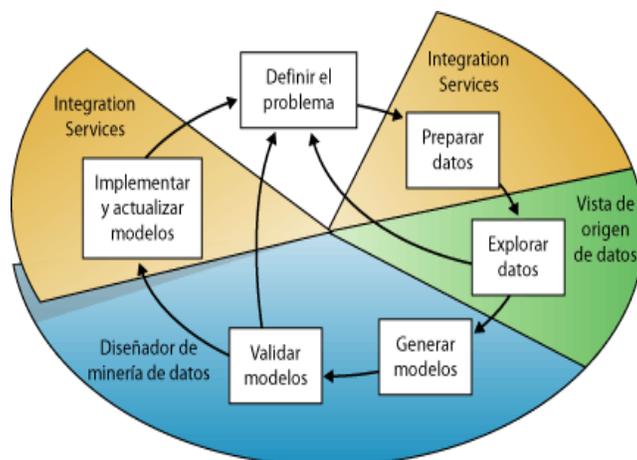


Figura no. 1 Descripción de las relaciones entre los procesos y las tecnologías.

## 2. Métodos

Para la implementación del prototipo, se requirió de una base de datos con información histórica acerca de las rutas y horas con mayor congestión de usuarios, por lo que se tomaron en cuenta investigaciones anteriores que apoyaran al proyecto y se encontró que recientemente, un número de algoritmos motivados en grupos sociales se han utilizado para solucionar problemas complejos de optimización. Se investigaron algoritmos motivados en Inteligencia Grupal, los cuáles han sido utilizados en diferentes aspectos sociales para solucionar problemas complejos de optimización.

- Algoritmos que incluyen el cúmulo de partículas (PSO),
- Algoritmos culturales (ACs), y
- Colonia de hormigas (ACO).

Estos tres algoritmos utilizan un modelo basado en una población como la parte medular del algoritmo y solucionan problemas compartiendo la información

Estos tres algoritmos utilizan un modelo basado en una población como la base del algoritmo y solucionan problemas compartiendo la información vía la interacción social entre agentes en la población (Ochoa, 2008).

El prototipo se basa en un algoritmo ACs (Algoritmo de Colonia de Hormigas. Por sus siglas en inglés: Ant Colony System), el cual estudia un sistema artificial de hormigas inspirado en la conducta colectiva de hormigas reales, utilizados para resolver problemas de optimización combinatoria (Barán, 2002).

El algoritmo de la colonia de hormigas para la búsqueda de la información importante (Gómez, 2009), tiene dos objetivos: Maximizar los recursos encontrados y minimizar los pasos dados. Este algoritmo guía las consultas hacia los nodos que tienen la mejor conectividad usando la métrica del Coeficiente del Grado de Dispersión (DDc por sus siglas en inglés). El DDC mide las diferencias entre el grado de un vértice y los grados de sus vecinos con el fin de reducir al mínimo los saltos. Por lo tanto, la consulta más frecuente a un recurso, será la mejor ruta seleccionada. El algoritmo de la Colonia de Hormigas consta de dos fases principales:

**Fase 1:** La evaluación de los resultados (Líneas 03-06. Tabla 1): Implementa la técnica de Buscar hacia Adelante. Esto es, la hormiga  $k$ , ubicada en el nodo  $s$ , solicita a los nodos vecinos los recursos solicitados. Si el recurso es encontrado, el resultado es recuperado. Si la evaluación falla, se efectúa la fase 2.

**Fase 2:** Estado de transición (Líneas 07-17. Tabla 1): Selecciona por  $q$ , un nodo vecino  $s$ . En el caso de que no haya un nodo hacia el cuál moverse (Es decir, el nodo es el fin ó todos los nodos vecinos han sido visitados), saltar hacia atrás sobre la ruta que se esté haciendo, de otra forma, la hormiga integra al nodo  $s$  a su ruta, y reduce un salto a  $TTL_k$ . El proceso de la consulta termina cuando el resultado esperado es satisfactorio o  $TTL_k$ , es igual a cero, entonces la hormiga se destruye indicando que es el final de la consulta.

Tabla 1 Algoritmo de Colonia de Hormigas.

01	para cada <i>consulta</i>
02	para cada <i>hormiga</i>
03	si $H$ es $<maxResults$ y $TTL > 0$
04	si el vecino de la orilla $s_k$ tiene resultado
05	abre $s_k$ , a $Path_k$ y $TTL_k = TTL_k - 1$
06	Pheromone globalUpdate
07	si no
08	$S_k =$ aplica la regla de transi. con la DDC
09	si la ruta no existe, o nodo visitado,
10	Quitar el último nodo de $Path_k$
11	si no
12	Abre $s_k$ a $Path_k$ y $TTL_k = TTL_k - 1$
13	Pheromone localUpdate
14	Fin si
15	Fin si
16	Si no
17	Destruye Hormiga
18	Fin si
19	Fin para
20	Fin para

El prototipo cuenta con una base de datos que contiene todas las rutas que se usan actualmente en la ciudad de León y está soportada en el manejador de base de datos MySQL (ver figura 2) y fue realizado en el lenguaje de programación java.

	clavedetalle	claveruta	colonia	calle	sentido	posicion
	1	R-1	LAS HILAMILLAS	CEFERINO ORTIZ	0	SW
	2	R-1	CIUDAD SATELITE	MARIANO ESCOBEDO	0	SW
	3	R-1	JOL-GUA-BER	21 DE MARZO	0	SW
	4	R-1	JOL-GUA-BER	NUEVA GALICIA	0	SW
	5	R-1	JOL-GUA-BER	SAN JUAN BOSCO	0	NW
	6	R-1	CHAPALITA	VALVERDE Y TELLEZ	0	NW
	7	R-1	OBREERA	VALVERDE Y TELLEZ	0	NW
	8	R-1	OBREGON	MELCHOR OCAMPO	0	NW
	9	R-1	EL COECILLO	MELCHOR OCAMPO	0	NW
	10	R-1	BUGAMBILIAS	LA LUZ	0	NE
	11	R-1	SAN JORGE	PASEO DE JEREZ	0	SE
	12	R-1	SAN JORGE	MADRE SELVA	0	SE
	13	R-1	SAN JORGE	OBELISCO	0	SE
	14	R-1	SAN JORGE	OLIMPICA	0	SE
	15	R-1	SAN JORGE	DE LOS INDUSTRIALES	0	SE
	16	R-1	SAN ISIDRO	PASEO DE JEREZ	0	SE
	17	R-1	SAN ISIDRO	TORRES LANDA	0	SE
	18	R-1	CERRITO DE JEREZ	CERRITO DE JEREZ	0	SE
	19	R-1	CERRITO DE JEREZ	DEFI TA	1	SE

Figura 2 Base de datos del prototipo.

El prototipo carga las colonias de la ruta donde se encuentra la colonia origen en un arreglo. Cada elemento del arreglo es un nodo que contiene las colonias por donde pasa el transporte público. De acuerdo al algoritmo de la tabla 1, el prototipo solicita la colonia y calle origen y la colonia y calle destino (ver figura 3). La hormiga en este momento, se dispone a recorrer el arreglo para localizar la colonia origen (En el ejemplo: colonia Casa Blanca). Después de encontrarla, deja la feromona y comienza el recorrido sobre el mismo arreglo, realizando la búsqueda de la colonia destino, tomando como premisa que esta se encuentra dentro de la misma ruta.

Dado que en el ejemplo la colonia destino se encuentra en diferente ruta, el algoritmo pasará a la fase dos donde buscará en otras rutas la colonia destino. (ver figura 4), en caso contrario, el algoritmo encontraría la ruta destino y terminaría su ejecución.

En el ejemplo, la hormiga comienza a hacer el trazado de la ruta, tomando en cuenta que el pasajero deberá realizar trasborde de transporte (ver figura 5)

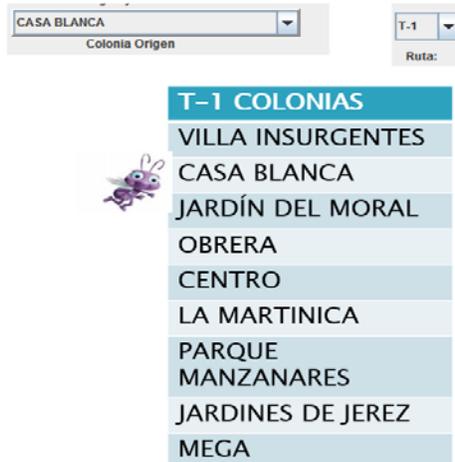


Figura 3 La hormiga apunta la colonia origen.



Figura 4 El sistema carga en el arreglo 2 las rutas que contienen la colonia destino.



Figura 5 La hormiga localiza la colonia para realizar el trasborde.

La hormiga deja la feromona (representada por la hormiga morada) y ahora hace recorridos en los dos arreglos para buscar coincidencias de colonias. Al encontrar la coincidencia (colonia Obrera), la hormiga deja las feromonas correspondientes (representada por las hormigas rojas), con lo que se determina la colonia donde se hará el trasborde.

Finalmente, la hormiga hace la búsqueda de la colonia destino, que en el prototipo por defecto, comienza a partir de la colonia de trasborde hacia arriba del arreglo y en caso de no encontrarla, continúa con la búsqueda hacia abajo del arreglo. En el ejemplo, la colonia destino (Hilamillas) se encontró haciendo el recorrido hacia arriba (ver figura 6).



Figura 6 Localización de la colonia destino.

Aquellas colonias (nodos) por donde exitosamente se encontró la ruta de la colonia origen a la colonia destino, son almacenados en un arreglo para optimizar la búsqueda, descartando aquellos nodos que no cumplan con las condiciones y tomando a aquellos nodos que si formen parte de la propuesta. Dado que estos nodos contienen el sentido de las calles, se forma la ruta indicando al pasajero las colonias por donde va a pasar, junto con las calles y además le indica la calle donde debe bajarse para hacer el trasborde, continuando con la descripción de las colonias y calles por donde va a pasar para llegar a la colonia destino (ver figura 7).

Consulta de Rutas de Transporte Público

Selección de Origen y Destino

Colonia Origen: CASA BLANCA | Calle Origen: BLVD. CAMPESTRE | Posicion: NW | Ruta: T-1

Colonia Destino: LAS HILAMILLAS | Calle Destino: CEFERINO ORTIZ | Posicion: SW | Ruta: R-1

Hora de Salida: 5:50

Generar | Mapa | Limpiar | Cerrar

**Resultado Sugerido de la Consulta**

Tomar ruta T-1 pasa colonia CASA BLANCA por calle BLVD. CAMPESTRE pasa colonia JARDINES DEL MORAL por calle BLVD. ADOLFO LOPEZ MATEOS pasa colonia OBRERA por calle BLVD. ADOLFO LOPEZ MATEOS BAJE en esa calle y busque la ruta R-1 pasa colonia OBRERA por calle VALVERDE Y TELLEZ pasa colonia CHAPALITA por calle VALVERDE Y TELLEZ pasa colonia JOL-GUA-BER por calle SAN JUAN BOSCO pasa colonia JOL-GUA-BER por calle NUEVA GALICIA pasa colonia JOL-GUA-BER por calle 21 DE MARZO pasa colonia CIUDAD SATELITE por calle MARIANO ESCOBEDO pasa colonia LAS HILAMILLAS por calle CEFERINO ORTIZ

Figura 7. Pantalla de Consulta de Rutas.

### 3. Resultados

La implementación del algoritmo de Colonia de Hormigas, para la Optimización del Transporte público, permitió formar rutas a través de la filosofía del algoritmo que marca la información necesaria contenida en la base de datos con feromonas para su localización, dando como resultado un prototipo que procesa las calles como nodos para la aplicación, y que ésta sigue de acuerdo al algoritmo, para formarrutas de transporte que se presentan al usuario como recomendaciones.

Además de lo anterior, el uso de los algoritmos evolutivos permitió en la programación de la aplicación, optimizar la búsqueda de la información, con lo cual se descarta aquella que no satisface la consulta inicial.

### 4. Discusión

El prototipo para la Optimización del Transporte Público, calcula la ruta óptima entre dos puntos. Estos puntos, deben encontrarse dentro del mapa de una ciudad, en este caso, de León, Guanajuato. Cuenta con una base de datos realizada con software libre, que contiene todas las calles y colonias de la ciudad, así como las rutas del transporte público que pasan por esas colonias. Estas rutas también contienen el sentido de la ruta para realizar los cálculos que regresarán la ruta óptima.

Los cálculos que se realizan en el presente prototipo, están basados en el Algoritmo de Colonia de Hormigas; en donde se lanza un apuntador, que hace la labor de la hormiga pionera y primeramente busca rutas de transporte que pasen por el origen de salida.

Cuando se encuentra la primer ruta origen, se lanza otro apuntador que busca si el destino se encuentra en la misma ruta. En caso de encontrarla, se construye la ruta a través del camino que haya seguido el segundo apuntador. En caso de que el destino no se encuentre dentro de la misma ruta, el prototipo lanza otras búsquedas a otras rutas para buscar aquella que llegue lo más cerca de la ruta destino.

El prototipo de Optimización del Transporte Público, sirve a los pasajeros de este transporte para tomar el que sea adecuado y llegar a su destino o cerca de éste, sobre todo cuando estos viven en ciudades muy grandes, con un gran número de rutas de transporte o, cuando son personas foráneas y que requieren este servicio. De lo contrario, deben invertir en otros servicios, arriesgándose a pagar más, o a que no lleguen a tiempo a su destino.

Además de lo anterior, el usuario administrador del prototipo, puede actualizar la información de la base de datos, en caso de que se requiera, por implementación de rutas nuevas o cambios en las rutas; ya que contiene una pantalla en donde puede realizar los cambios pertinentes al transporte público. Lo anterior, permite adecuar el prototipo a las necesidades cambiantes de la ciudad (ver figura 8).

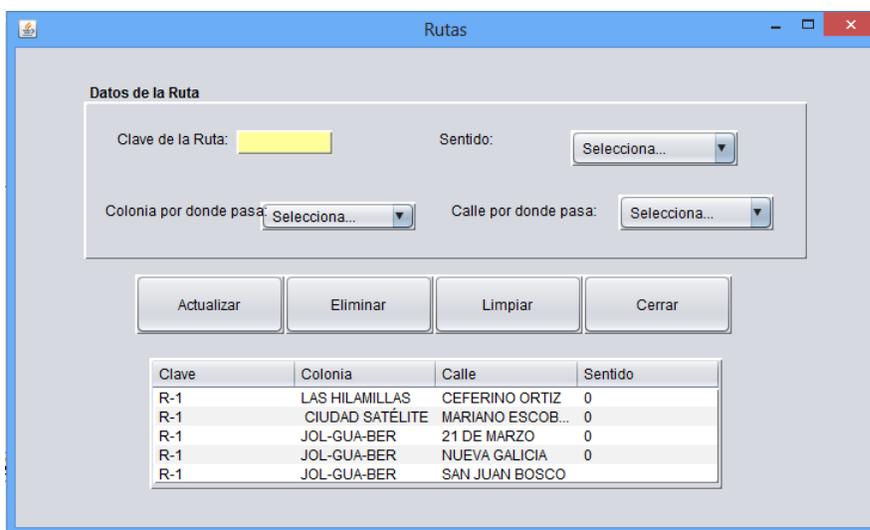


Figura 8 Pantalla para actualizar rutas.

## Bibliografía

- [1] Ochoa Ortiz Zezzatti, Alberto; “Algoritmos Culturales”; Ideas CONCYTEG, Año no. 3, 21 de enero del 2008.
- [2] Moreno Cortés, Luis Enrique (2009). “Reseña Histórica; Dirección General de Transporte Público”.  
<http://www.leon.gob.mx/portal/modules/icontent/index.php?page=229>
- [3] Barán, Benjamín; Almirón, Martha (2002). “Colonia de Hormigas en un Ambiente Paralelo Asíncrono”. Centro Nacional de computación, Universidad de Asunción. Barán, Benjamín;  
<http://www.cnc.una.py/cms/invest/download.php?id=104917,101,1>
- [4] Gómez Santillán, Claudia; Cruz Reyez, Laura; Meza Conde, Eustorgio; Amaro Martínez, Claudia; Aguirre Lam, Marco Antonio, Ochoa Ortiz Zezzatti Carlos Alberto; “Performance Analysis of the Neighboring-Ant Search Algorithm through Design of Experiment”.