

# Un algoritmo genético híbrido para encontrar las k rutas mas corta en un red de transporte

Jose Manuel Torres Lima  
Universidad Nacional de San Agustin  
Escuela de Ciencia de la Computación  
Arequipa, Perú  
manu.jose.torres@gmail.com

**Abstract**—El problema de encontrar la ruta mas corta es uno de los problemas clásicos en la teoría de grafos. Existen varios algoritmos que calculan la ruta más corta, pero la mayoría de ellos proporcionan solo una solución. Por lo tanto, este paper propone el uso de algoritmos bioinspirados para este problema. En este paper se describe cómo el uso de algoritmos genéticos junto con el algoritmo de la colonia de hormigas se puede aplicar para este problema. Además se propone un algoritmo genetico híbrido que agrega una operacion de busqueda local en la mutacion del algoritmo genetico clasico, para asi poder explorar mas posibles soluciones al problema, ademas de generar su siguiente poblacion a partir de la poblacion inicial del algoritmo de colonia de hormigas para iniciar la busqueda, reduciendo el numero de iteraciones del algoritmo genetico clasico. Como resultado se obtuvo que el algoritmo propuesto encuentra las distancias minimas en un menor numero de iteraciones del algoritmo genetico clasico. Se hizo una comparacion con el algoritmo de colonia de hormigas(ACS) que obtiene buenos resultados, obteniendo los mismos resultados.

**Keywords:** Algoritmos genéticos, algoritmo de la colonia de hormigas, rutas mas cortas.

## I. INTRODUCCIÓN

El problema de la encontrar ruta mas corta es un problema clásico en teoría de grafos. El problema de encontrar la ruta mas corta es aplicado en los campos de red informática, en la comunicación y en el transporte urbano. Hay muchos algoritmos clásicos que solucionan este problema como por ejemplo el algoritmo de Floyd[1], el algoritmo de Dijkstra[1], el algoritmo de búsqueda A\*[1].

Los conocidos y ampliamente utilizados son los algoritmos de Dijkstra y A\* funcionan bien, pero solo dan una solución. En una red de transporte multimodal (por ejemplo, transporte público) hay varias rutas que podrían mejorar el nivel de servicio y la experiencia del pasajero reduciendo costos de espera en el traslado del pasajero. Por este motivo, el uso de algoritmos genéticos y el algoritmo de colonia de hormigas podrían ser buenas alternativas.

Trabajos relacionados en el problema de encontrar la ruta mas corta también es aplicado a redes de computador en un proceso para encontrar la ruta mas corta entre el origen y el destino en una red. También aplicado en una red de conmutación de paquetes[2]. Por otro lado el uso

de Algoritmos genéticos empleando un método basado en computación en la nube para proporcionar mejoras en el aumento del rendimiento[3]. El uso de algoritmos genéticos junto a el algoritmo de la colonia de hormigas se pueden aplicar al sistema de transporte multimodal(p.e transporte publico) para encontrar la ruta mas corta[4].

Otro de los trabajos relacionados hace uso de algoritmos genéticos para resolver el problema de las k rutas más cortas basado en el ancho de banda más corto posible entre dos nodos en las redes de comunicación (suponiendo que cada borde en la red tiene el valor de ancho de banda)[5]. También se puede aplicar algoritmos genéticos a problemas BSP con dos funciones objetivo, es decir con dos objetivos, como por ejemplo minimizar el costo de transporte y el tiempo total de viaje. La solución consiste en encontrar rutas que sean eficientes con respecto a ambos criterios[6]. Y por ultimo el uso de un árbol de expansión orientado(OST) basado en algoritmos genéticos (GA) para el problema multi-criterio del camino más corto (MSPP), así como el problema de la ruta más corta restringida de criterios múltiples (MCSP)[7].

El resto de este documento esta organizado de la siguiente manera en la siguiente sección describimos los algoritmos genéticos, en la sección III describimos el algoritmo de la colonia de hormigas, en la sección IV mostramos la representación de las rutas de carreteras, en la sección V mostramos el esquema del método propuesto. los resultados experimentales obtenidos son mostrados en la sección VI. Finalmente en la sección VII concluimos este documento dando las conclusiones de la investigación.

## II. ALGORITMOS GENÉTICOS

Los algoritmos genéticos son algoritmos de búsqueda heurística adaptativos, que se basan en la evolución biológica, siguiendo los principios de supervivencia del más apto en la naturaleza. Generando una población inicial donde cada elemento de la población es un cromosoma. Un cromosoma es una solución y se evalúa usando una función de aptitud que nos indica el valor de este cromosoma. Este valor representa la calidad de la solución. En cada iteración nuevos descendientes son generados usando operadores genéticos de selección, cruzamiento y mutación entre los cromosomas. En

cada generación los mejores cromosomas son elegidos. Este proceso se repite hasta que los cromosomas tengan la mejor solución para el problema o hasta alcanzar un cierto número de iteraciones.

**Población:** La población inicial de Cromosomas que representan varias rutas, son una cadena de nodos de la forma:

Cromosoma 1: 0, 1, 2, 4, 5, 6, 9, 12

Cromosoma 2: 0, 2, 4, 12

Cromosoma 3: 0, 3, 5, 7, 12

Donde 0 es el nodo inicial, posición inicial de partida y 12 representa el nodo final posición a donde se quiere llegar a partir del nodo inicial.

**Función aptitud:** Para cada cromosoma, se calcula su aptitud que vendrá a ser la suma de las distancias de ir de un nodo a otro en la ruta. Por ejemplo en el individuo 1 será la suma de las distancias de ir del nodo 0 hacia el nodo 1, de este hacia el nodo 2, hacia el 4,5,6,9, hasta llegar hasta el nodo 12.

**Selección:** La selección de cromosomas se hará por ruleta a través de las probabilidades de cada cromosoma de ser elegido, esta probabilidad será calculada como la división de la inversa de la distancia de cada cromosoma hallado con la función aptitud entre la suma total (suma de las inversas de las distancias de todos los cromosomas de la población). Se eligió la selección por ruleta y no por torneo para no obviar posibles soluciones en la selección de candidatos como lo hace la selección por torneo. Los cromosomas con menor distancia tendrán mayor probabilidad de ser elegidos.

**Cruzamiento:** El cruzamiento se realiza sobre los cromosomas elegidos por la selección por ruleta y se combinarán sus genes para diversificar las soluciones obtenidas y así poder abarcar más cromosomas no considerados en la búsqueda, con el fin de encontrar la solución dentro de todas las combinaciones de los cromosomas elegidos. El operador cruzamiento generará dos nuevos cromosomas a partir de dos cromosomas elegidos por selección. La probabilidad de cruzamiento nos indica cuántas de las parejas de cromosomas elegidos se cruzan, no todas las parejas elegidas realizarán el cruzamiento, eso dependerá de la probabilidad de cruzamiento. Ejemplo de cruzamiento:

Se elige un nodo que este contenido en ambos cromosomas, en este ejemplo será el nodo 3. Los cromosomas resultantes son obtenidos copiando los nodos desde el nodo inicial (nodo 0) hasta el nodo elegido (nodo 3) y los nodos del otro cromosoma hasta el nodo final (nodo 12).

Cromosomas elegidos:

Cromosoma 1: 0, 1, 3, 6, 8, 12

Cromosoma 2: 0, 4, 3, 5, 12

Resultado:

Cromosoma 1: 0, 1, 3, 5, 12

Cromosoma 2: 0, 4, 3, 6, 8, 12

**Mutación:** El operador mutación se realiza en un cromosoma de la siguiente manera, se selecciona un nodo aleatorio dentro del cromosoma y se origina una nueva ruta desde ese nodo hasta el nodo final conservando la ruta desde el nodo inicial hasta dicho nodo seleccionado. La probabilidad de mutación nos indica cuántos de los cromosomas en la población mutan. Ejemplo:

Nodo de mutación elegido 5

Individuo elegido: 0, 1, 4, 5, 7, 8, 12

Individuo Mutado: 0, 1, 4, 5, 10, 12

**Evaluar Individuos:** Para esta parte se elegirá la mitad de los cromosomas de la población en cada generación y la mitad de los cromosomas de la población obtenida después del cruzamiento y la mutación de los cromosomas, estos conformarán la nueva población para las siguientes iteraciones. Esto para no descartar posibles soluciones dentro de los cromosomas de las poblaciones.

**Condición de parada:** Se especificará un número de iteraciones como condición de parada. La población final será la solución.

### III. ALGORITMO DE LA COLONIA DE HORMIGAS

El algoritmo de la colonia de hormigas es un algoritmo heurístico que consiste en el comportamiento que tienen las hormigas para encontrar la ruta más corta para transportar su alimento. Consta básicamente de las feromonas que van dejando a su paso por una ruta. Si el camino es corto la feromona será más tiempo percibida por las hormigas haciendo que estas elijan esa ruta, mientras que si el camino es largo la feromona se irá evaporando con el tiempo y será menos percibida por las hormigas.

Además del valor de la feromona se usará un valor de visibilidad que es un valor fijo igual a la inversa de la distancia entre los nodos que será representado por  $\eta_{ij}$ .

Dentro de los tipos de algoritmos de colonia de hormigas se eligió Ant Colony System (ACS), porque cuenta con operaciones de diversificación y intensificación que ayudan a que el algoritmo converja de manera más rápida a una posible solución.

**Selección de la ruta:** La elección de los nodos se hace a través de:

$$j = \begin{cases} \arg \max_{l \in \mathcal{N}_i^k} \{\tau_{il}[\eta_{il}]^\beta\} & , q \leq q_0 \\ J & , \text{en otro caso} \end{cases} \quad (1)$$

Donde  $q$  es una variable aleatoria, uniformemente distribuida entre el rango de  $[0, 1]$ ,  $q_0 \in [0, 1]$  es un parámetro que intensifica o diversifica la elección de los nodos en la construcción de las rutas. Si  $q \leq q_0$  se intensifica, caso contrario se diversifica la exploración de la ruta. La diversificación se hace a través de:

$$p_{ij}^k = \begin{cases} \frac{[\tau_{ij}]^\alpha [\eta_{ij}]^\beta}{\sum_{l \in \mathcal{N}_i^k} [\tau_{il}]^\alpha [\eta_{il}]^\beta} & , j \in \mathcal{N}_i^k \\ 0 & , \text{en otro caso} \end{cases} \quad (2)$$

Donde  $\tau_{ij}$  es la cantidad de feromona entre  $i$  y  $j$ ,  $\eta_{ij}$  es la inversa de la distancia entre  $i$  y  $j$ ,  $\alpha$  es un factor que indica la importancia del valor de la feromona,  $\beta$  es un factor que indica la importancia del valor de la visibilidad y  $\mathcal{N}$  es el conjunto de los nodos no visitados por la hormiga  $k$ .

#### Actualización de la feromona local:

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

Donde  $\varphi \in [0, 1]$  es el coeficiente de disminución de feromona, y  $\tau_0$  es el valor inicial de la feromona.

#### Actualización de la feromona global:

$$\tau_{ij} = \rho \cdot \tau_{ij} + (1 - \rho) \cdot \Delta \tau_{ij}^{bs} \quad (4)$$

Donde  $\rho$  es una constante que controla la evaporación de la feromona y donde  $\Delta \tau_{ij}^{bs}$  es:

$$\Delta \tau_{ij}^{bs} = \begin{cases} \frac{1}{L^{bs}} & , \text{si el arco}(i, j) \in S^{bs} \\ 0 & , \text{en caso contrario} \end{cases} \quad (5)$$

#### IV. RED DE TRANSPORTE

Para representar las rutas de una red de transporte, se usará una matriz de  $n \times n$  donde  $n$  es el número de nodos en la red que representan las ciudades, la distancia será el valor de la posición entre el nodo  $i$  y el nodo  $j$  dentro de la matriz que será la distancia de ir desde la ciudad  $i$  hasta la ciudad  $j$ . Ejemplo de una red representada por una matriz de  $13 \times 13$ .

	0	1	2	3	4	5	6	7	8	9	10	11	12
0	0	12	13	41	83	92	12	11	60	67	13	23	12
1	12	0	34	72	56	41	0	62	0	34	0	45	0
2	13	34	0	12	76	45	67	33	23	56	28	0	12
3	41	72	12	0	0	11	12	0	15	13	11	22	12
4	83	56	76	0	0	67	45	67	34	23	45	33	89
5	92	41	45	11	67	0	28	32	17	19	13	89	45
6	12	0	67	12	45	28	0	27	13	17	16	34	56
7	11	62	33	0	67	32	27	0	19	32	45	67	67
8	60	0	23	15	34	17	13	19	0	23	34	56	88
9	67	34	56	13	23	19	17	32	23	0	34	34	98
10	13	0	28	11	45	13	16	45	34	34	0	45	34
11	23	45	0	22	33	89	34	67	56	34	45	0	36
12	12	0	12	12	89	45	56	67	88	98	34	36	0

Fig. 1: Ejemplo de la representación de la red de transporte

Los ceros dentro de la matriz nos indica que no hay una ruta directa desde un nodo  $i$  hasta un nodo  $j$ .

#### V. MÉTODO PROPUESTO

Para el método propuesto la población inicial no será generada aleatoriamente, la población inicial será generada a partir de la población inicial del algoritmo de colonia de hormigas, se usarán solo dos iteraciones para el algoritmo de colonia de hormigas para generar la población inicial del algoritmo genético híbrido, una ventaja del algoritmo de la colonia de hormigas es que convergen a posibles soluciones de manera rápida, pero no se garantiza que cubran todo el conjunto de rutas (soluciones), pudiendo quedar atascadas en una solución local y no global en algunos casos, pero el algoritmo Ant Colony System logra obtener buenos resultados por los operadores de diversificación y intensificación en sus soluciones.

Se modificará la mutación del algoritmo genético agregando una búsqueda local en los cromosomas, esta búsqueda consiste en realizar más mutaciones dentro del operador de mutación buscando encontrar la mínima distancia a partir de las mutaciones del cromosoma original. Esto con la finalidad de obtener la mínima distancia generada a partir de este cromosoma. Para que el algoritmo híbrido propuesto converja de manera rápida a una solución.

Los parámetros iniciales se establecen al inicio del algoritmo genético como el número de iteraciones, el tamaño de la población, la probabilidad de cruzamiento y de mutación, el número de mutaciones que realizará la búsqueda local entre otros. Véase la Fig. 2.

## VI. RESULTADOS

Para las pruebas se eligió una matriz de 226x226 nodos descargada de <https://sites.google.com/site/awazyp/tsp>

Se evaluaron los parametros para el algoritmo genetico clasico en tamaños de poblacion de 10,15 y 20 cromosomas para así obtener los parametros que mejor se adapten a la solucion. Se consideraron los resultados obtenidos con una poblacion de tamaño 20. Los valores evaluados con la poblacion de 20 cromosomas desde un nodo inicial 100 hasta un nodo final 206, con un numero de iteraciones de 50 se muestran a continuacion:

Algoritmo Genetico			
Cruzamiento	Mutacion	Poblacion	Aptitud
0.9	0.1	20	856
0.5	0.1	20	1026
0.3	0.3	20	1013
0.5	0.3	20	426
0.9	0.3	20	704
0.7	0.3	20	620
0.5	0.5	20	1366

De la tabla se podria decir que a menor probabilidad de cruzamiento la solucion explorada no cambia , con lo que no se mejora la poblacion a traves de cada iteracion, y que a una mayor probabilidad de cruzamiento los resultados mejoran de cierta manera, tambien se observa que a mayor probabilidad de mutaciones la solucion encontrada cambia constantemente y los resultados obtenidos se pierden en las iteraciones, mientras que a una menor probabilidad de mutacion no se lleva a cabo una exploracion en los demas cromosomas(posibles soluciones) que no son parte de la poblacion actual que podrian ser la solucion a encontrar. Se encontro un equilibrio entre la probabilidad de cruzamiento y mutacion con los valores de 0.5 y 0.3 respectivamente y se obtuvo una menor distancia en las iteraciones.

Para el algoritmo de colonia de hormigas se evaluaron los parametros para una poblacion de 10,15 y 20 hormigas encontrando mejores resultados en una colonia de tamaño 20 desde un nodo inicial 100 hasta un nodo final igual a 206, los parametros evaluados se muestran en la siguiente tabla para un numero de 50 iteraciones:

Algoritmo de Colonia de hormigas(ACS)						
$\rho$	$q_0$	$\alpha$	$\beta$	$\varphi$	Poblacion	Aptitud
0.9	0.7	0.1	0.1	0.05	20	426
0.7	0.5	0.6	0.6	0.1	20	468
0.7	0.3	1.1	1.1	0.9	20	855
0.5	0.5	2.1	2.1	0.1	20	578
0.9	0.7	0.1	0.1	0.1	20	426

De la tabla se podria decir que a un mayor valor de  $\rho$  que representa la evaporacion de la feromona, se obtiene

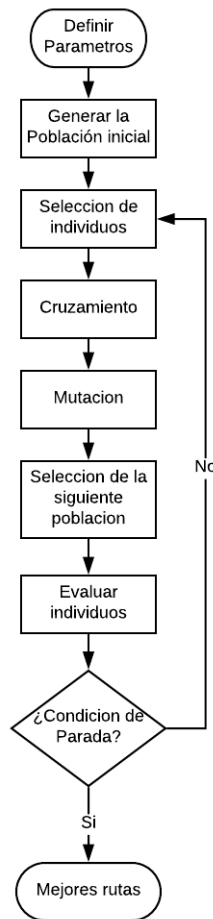


Fig. 2: Esquema del Algoritmo Genético

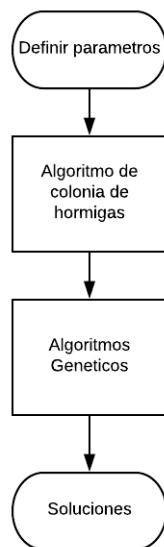


Fig. 3: Esquema General del algoritmo hibrido

un menor resultado en la distancia esto debido a que la evaporacion de la feromona disminuye lentamente con lo que se conserva la menor distancia encontrada, para un menor valor de  $\rho$  la feromona se evaporara de manera rapida reduciendose su valor en menos tiempo, no conservando las soluciones que no son visitadas nuevamente por las hormigas,  $q_0$  indica si se realizara una diversificacion(exploracion de las demas rutas) o si se intensifica las feromonas de las rutas ya encontradas. Para un mayor valor de  $q_0$  se intensifica las rutas encontradas reduciendose la diversificacion, para un valor menor se diversifica las rutas, explorando nuevas posibles soluciones, reduciendo la intensificacion de las rutas encontradas. De la tabla se obtuvieron buenos resultados para una valor de  $\rho$  igual a 0.9 y un valor de  $q_0$  igual a 0.7 con una  $\alpha$  y  $\beta$  de 0.1 y una valor de  $\varphi$  de 0.05 o 0.1.

Para el algoritmo propuesto se hizo una comparacion con el algoritmo de colonia de hormigas elegido(ACS) debido a que obtiene buenos resultados en comparacion con otros tipo de algoritmos de colonia de hormigas. Para el algoritmo genetico hibrido se usaron los parametros obtenidos en las pruebas anteriores con el algoritmo genetico clasico. Se usaron 2 iteraciones para la obtencion de la poblacion inicial apartir del algoritmo de colonia de hormigas usandose las iteraciones restantes para las demas operaciones del algoritmo genetico hibrido, para la busqueda local se uso un numero de mutaciones igual a 10 que significa que se haran 10 mutaciones a un cromosoma(ruta) escogiend el que genere la menor distancia. El tamaño de la poblacion para ambos algoritmos es de tamaño 30, con un numero de 30 iteraciones.

Se calculo la distancia de ir del nodo 100 hacia el 206. Con el algoritmo de colonia de hormigas(ACS) se obtuvo la siguiente grafica en la exploracion de la solucion, el tiempo tomado en el proceso de busqueda fue de 51.07 segundos.

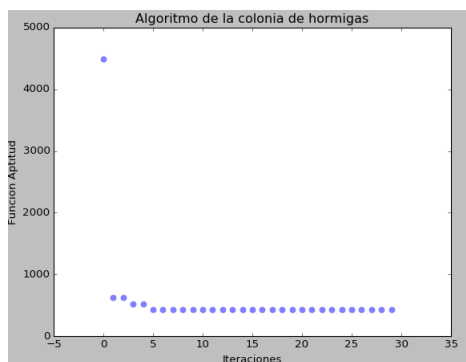


Fig. 4: Algoritmo colonia de Hormigas

La siguiente grafica muestra la exploracion del algoritmo propuesto, el tiempo que tomo fue de 26.34 segundos en la busqueda.

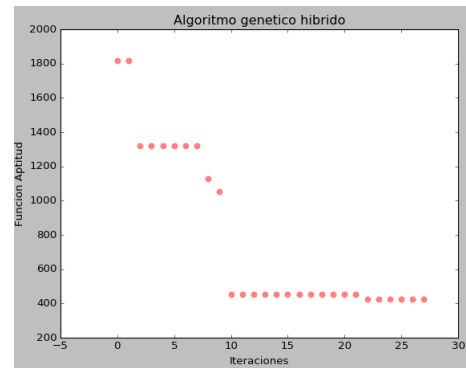


Fig. 5: Algoritmo propuesto

Las distancias encontradas por ambos algoritmos en las busquedas fueron de 452.0 de la siguiente ruta 100-79-175-201-1-206 y 426.0 desde la siguiente ruta 100-206 en este caso la ruta fue directa.

Para la siguiente prueba se considero la distancia de ir de 10 a 136. En el siguiente grafico se muestra la exploracion de soluciones con el algoritmo de colonia de hormigas(ACS) que obtiene un tiempo de 40.71 segundos en la busqueda.

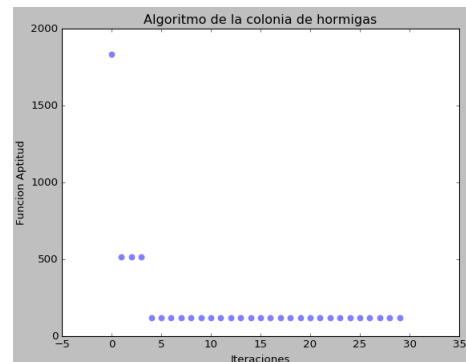


Fig. 6: Algoritmo colonia de Hormigas

El algoritmo propuesto requiere de un tiempo de 18.20 segundos en la busqueda.

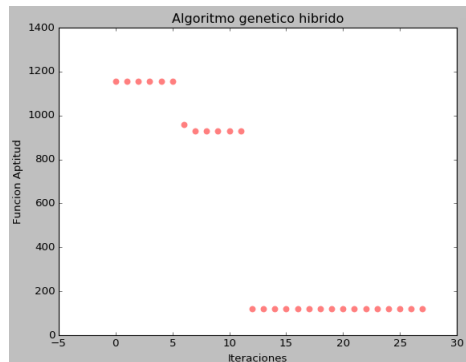


Fig. 7: Algoritmo propuesto

Solo se encontro una distancia por ambos algoritmos que fue de 119.0 con una ruta directa de 10-136

Para la siguiente prueba se eligio la ruta minima desde el nodo 35 hacia el 13. Para el grafico se muestra de igual manera la exploracion del algoritmo de colonia de hormigas(ACS). En este caso tomo un tiempo de 40.39 segundos en la busqueda. Encontrando una distancia de 574 con la ruta 35-152-13

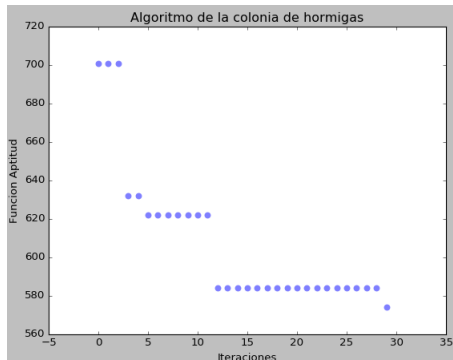


Fig. 8: Algoritmo colonia de Hormigas

El algoritmo propuesto requiere de menos tiempo igual a 28.62 segundos consiguio una menor ruta de 555 con la ruta 35-13 ademas de esta se obtuvo otra distancia minima de 564.

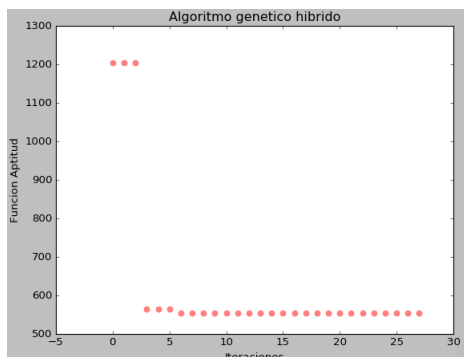


Fig. 9: Algoritmo propuesto

## VII. CONCLUSIONES Y TRABAJOS FUTUROS

En las pruebas nos dimos cuenta que a un mayor numero de iteraciones se llega a una solución, pero lo que se quiere es encontrar la solución en un numero de iteraciones no muy grande. El método propuesto muestra buenos resultados para un numero de población no muy grande, entre mayor sea el tamaño de la población mayor será el costo de tiempo en la búsqueda. Los algoritmos genéticos muestran soluciones factibles pero ello presenta un costo en el tiempo para un conjunto de datos extenso, esto debido a que las operaciones son realizadas de manera secuencial. Una solución sería paralelizar las operaciones independientes dentro de los algoritmos tanto en los algoritmos genéticos como en el algoritmo de colonia de hormigas(ACS) para reducir estos tiempos. En las pruebas nos dimos cuenta que para ciertos parámetros(tamaño de la

población, probabilidad de cruzamiento, probabilidad de mutación, numero de mutaciones) la búsqueda mejora y que para otros la solución se atasca en una solución local. Dentro de los trabajos futuros esta la idea de paralelizar las operaciones usando Cuda, Spark o Map Reduce para reducir los costos de tiempo. Otro de los trabajos futuros sería también evaluar los lugares con mayor tráfico y ver si estos se encuentran dentro de las rutas seleccionadas en cada iteración para poder extraerlas del conjunto de soluciones para que de este modo la ruta sea la más óptima abarcando el problema del tráfico en la ciudad.

## REFERENCES

- [1] Miss.Pooja R. Katre, Dr. Anuradha Thakare."A Survey On Shortest path Algorithm For Road Network In Emergency Services". 2017 2nd International Conference for Convergence in Technology (I2CT)
- [2] Maniu Rares. "Adaptive mutation in genetic algorithms for shortest path routing problem". ECAI 2015 - International Conference – 7th Edition Electronics, Computers and Artificial Intelligence 25 June -27 June, 2015, Bucharest, ROMÂNIA
- [3] Chu-Hsing Lin, Jung-Chun Liu, Ming-Hong Liou, Wen-Chen Wu."Shortest Driving Time Computation Based on Cloud Technologies and Genetic Algorithm".2014 Fifth International Conference on Intelligent Systems, Modelling and Simulation
- [4] KATONA, Géza, PhD. JUHÁSZ, János, LÉNÁRT, Balázs."Compare Ant-colony and Genetic algorithm for shortest path problem and introduce their parallel implementations".2015 Models and Technologies for Intelligent Transportation Systems (MT-ITS) 3-5. June 2015. Budapest, Hungary
- [5] Ahmed Younes Hamed. "A genetic algorithm for finding the k shortest paths in a network". Egyptian Informatics Journal (2010) 11, 75–79
- [6] Cheikh Mohamed, Jarboui Bassem, Loukil Taicir. "A genetic algorithms to solve the bicriteria shortest path problem". Electronic Notes in Discrete Mathematics 36 (2010) 851–858
- [7] Linzhong Liu , Haibo Mu, Xinfeng Yang, Ruichun He, Yinzheng Li. "An oriented spanning tree based genetic algorithm for multi-criteria shortest path problems". Applied Soft Computing 12 (2012) 506–515
- [8] Tarak Nath Paul, Abhoy Chand Mandai."Set of Shorter Paths Using Genetic Algorithm and Evaluating Network Delay".2015 International Symposium on Advanced Computing and Communication (ISACC)
- [9] Omar DIB, Marie-Ange MANIER, Alexandre CAMINADA."A Hybrid Metaheuristic for Routing in Road Networks".2015 IEEE 18th International Conference on Intelligent Transportation Systems