

Una metaheurística híbrida aplicada a un problema de planificación de rutas

A hybrid metaheuristic applied to a route planning problem

Daniel Soto, Ing.¹; Wilson Soto, Esp.², Yoan Pinzón, PhD.²

1. Universidad San Martín – Sede Bogotá. 2. Grupo de Algoritmia y Combinatoria ALGOS-UN, Universidad Nacional de Colombia – Sede Bogotá
dsoto.7@gmail.com, {wesotof,ypinzon}@unal.edu.co

Recibido para revisión: 20 de Septiembre de 2008, Aceptado: 28 de Noviembre de 2008, Versión final: 19 de Diciembre de 2008

Resumen—El problema de planeación de rutas en el ámbito del enrutamiento de transporte ha sido un tema de gran interés debido a su alta complejidad computacional. Algunas de las variaciones a este problema son: el problema del agente viajero – *Travel Salesman Problem* (TSP) –, el problema de enrutamiento de vehículos – *Vehicle Routing Problem* (VRP) – y el múltiple problema del agente viajero – *Multiple Travel Salesman Problem* (mTSP) –. Dentro del VRP se encuentra el problema de recolección, el cual además de designar varias rutas para prestar un servicio de recolección a varios usuarios, tiene en cuenta la capacidad de la flota y la cantidad o demanda de capacidad por usuario. En este artículo se propone un algoritmo híbrido entre un algoritmo genético y un algoritmo de colonia de hormigas para tratar el problema de recolección básico con una flota de capacidad homogénea, múltiples depósitos y un periodo de m días. Finalmente, este trabajo muestra experimentalmente, el comportamiento del algoritmo híbrido en encontrar una solución óptima para el problema particular de recolección.

Palabras Clave—Inteligencia artificial, Optimización combinatorial, Metaheurísticas.

Abstract—The problem of route planning in the transport routing research field has been a topic of the most interest due to its high computational complexity. Some variants of this problem are: the Travel Salesman Problem (TSP), the Vehicle Routing Problem (VRP) and the Multiple Travel Salesman Problem (mTSP). Within the VRP we can find the pickup problem, which mainly fix several routes for rendering a pickup service to several users taking into account the fleet capacity and the capacity on demand's user. In this paper to propose a hybrid algorithm between a genetic algorithm and ant colony algorithm to deal with the pickup basic problem with homogeneous fleet capacity, multiple depots and a period of m days. Finally, this work shows experimentally, the performance of hybrid algorithm for finding an optimal solution for the particular pickup problem.

Keywords—Artificial intelligence, Combinatorial optimization, Metaheuristics.

I. INTRODUCCIÓN

El problema de planeación de rutas es uno de los principales problemas en la optimización de operaciones logísticas en sistemas de transporte que buscan primordialmente la reducción de costos. Este problema en el enrutamiento de transporte ha sido catalogado como NP-Completo debido a la gran cantidad de consumo en recursos computacionales para encontrar una solución óptima que crece de forma exponencial con respecto al tamaño del problema, razón por la cual es tratado con algoritmos que no tienen la necesidad de explorar todo el espacio de búsqueda asociado para dar una respuesta aproximada. Los algoritmos metaheurísticos son una familia de algoritmos cuya meta es precisamente dar soluciones aproximadas a problemas generales de tipo NP, sin necesidad de recorrer todo el espacio de búsqueda [15].

Las metaheurísticas son una derivación de las heurísticas clásicas. Las heurísticas clásicas realizan una exploración limitada sobre el espacio de búsqueda y normalmente las soluciones producidas son buenas en poco tiempo. Su implementación es sencilla y son fácilmente adaptadas a problemas del mundo real. Existen dos tipos de heurísticas: constructivas y de mejoramiento o de búsqueda local. Las metaheurísticas, por el contrario de las heurísticas, son un conjunto de conceptos diseñados para guiar a otras heurísticas en regiones específicas del espacio de búsqueda. Su ventaja es la exploración más profunda en el espacio de búsqueda pero a costa de mayor tiempo de procesamiento [1].

Principalmente, las metaheurísticas a menudo usan la experiencia ganada en búsquedas previas (memoria) para guiar nuevas búsquedas con el fin de encontrar soluciones cercanas a las óptimas y tienen la opción de incorporar mecanismos para evitar caer en áreas limitadas del espacio de búsqueda. Para evitar caer en estas áreas, las metaheurísticas balancean dinámicamente la exploración de la experiencia previamente ganada, llamada intensificación y la exploración del espacio de búsqueda, llamada diversificación. Este balance es necesario, por un lado, para rápidamente identificar la región en el espacio de búsqueda donde las soluciones son buenas, por otra parte, para no perder mucho tiempo buscando dentro de las regiones que ya han sido exploradas o que al parecer no tienen buenas soluciones [11].

En las últimas décadas el desarrollo de métodos no exactos para la planeación de rutas se ha focalizado en los algoritmos metaheurísticos: Recocido Simulado, Búsqueda Tabú, Algoritmos Genéticos y Colonia de Hormigas.

Este trabajo se centra en los algoritmos genéticos y en el algoritmo de colonia de hormigas y solo por citar algunos de los trabajos relacionados con estos algoritmos metaheurísticos aplicados al problema de planificación de rutas tenemos [1][6][11][14][16].

Uno de los problemas relacionados a la planificación de rutas es el problema de recolección. Este problema consiste en planificar las rutas para prestar un servicio de recolección a varios usuarios, teniendo en cuenta la capacidad de la flota y la demanda de cada usuario. En particular para este trabajo se adicionaron las restricciones de una flota de vehículos con capacidad uniforme, múltiples depósitos y un periodo de m días para el prestar el servicio.

Las contribuciones originales de este trabajo son: la implementación del algoritmo híbrido propuesto como resultado de un algoritmo genético con un algoritmo de colonia de hormigas para la búsqueda de una solución óptima al problema de recolección particular, desarrollo de una interfaz grafica que sirve para mostrar las rutas obtenidas basadas en las restricciones parametrizadas y con los resultados experimentales mostrar el comportamiento del algoritmo.

Este artículo tiene la siguiente estructura: la sección II explica los antecedentes sobre la planeación de rutas, sus variantes y algunos de los trabajos realizados en esta área. Además incluye la descripción de los conceptos de algoritmos genéticos y algoritmos de colonia de hormigas. La sección III explica en detalle el algoritmo propuesto. Las últimas secciones detallan la metodología experimental, los resultados obtenidos y las conclusiones.

II. ANTECEDENTES

El problema de la planeación de rutas es uno de los problemas más comunes en la optimización de operaciones logísticas y

por ende uno de los más estudiados. El planteamiento inicial del problema consiste en buscar la solución más óptima con diversos parámetros proporcionados por el usuario como el número de vehículos, la capacidad de los vehículos, lugares a visitar (clientes) y su demanda. El desarrollo de soluciones al problema de planeación de rutas en el área de operaciones logísticas ha sido de gran ayuda a la hora de hacer planeaciones de tiempo, distancias e inversión.

Los problemas más conocidos en el ámbito de la planeación de rutas son: el problema del agente viajero – *Travel Salesman Problem* (TSP) – y el problema de planeación de rutas vehiculares – *Vehicle Routing Problem* (VRP) –, para los cuales se han desarrollado diversos algoritmos que buscan una solución óptima con el menor costo computacional, e.g. [2][5][12].

El problema del agente viajero (TSP) consiste en encontrar el camino más corto posible entre un número variable de villas o sitios que se deben visitar y por los cuales solamente se debe pasar una sola vez, con la condición que al final del recorrido se debe retornar al punto de origen. Se ha demostrado que la complejidad del problema es $(n-1)!$, e.g., si se tienen 6 villas se tienen 120 posibles soluciones [9].

Una de las variantes del TSP es el problema de los múltiples agentes viajeros – *Multiple Traveling Salesman Problem* (mTSP) –, el cual consiste en construir m rutas para los diferentes agentes viajeros que interactúan con el mismo mapa de puntos de villas [13]. Otros problemas relacionados con TSP son: OLTSP – *On Line TSP* –, DTSP – *Deadline TSP* – y KTSP – *K Delivery TSP* .

El problema de planeación de rutas vehiculares (VRP) tiene por objetivo encontrar las rutas que recorran cada uno de los vehículos (ubicados en un depósito) de manera que se satisfagan los requerimientos de los clientes, las restricciones operativas y se minimice el costo total de transporte. Las siguientes son algunas de las variantes más importantes del problema VRP.

El problema de planeación de rutas vehiculares con capacidades – *Capacitated Vehicle Routing Problem* (CVRP) – trata de determinar los recorridos de una flota de vehículos con capacidad homogénea que parten de un origen común y deben pasar por un conjunto de lugares de interés (clientes) para recoger o distribuir mercancías según la demanda y volver de nuevo al origen de manera que la distancia recorrida (el coste o el tiempo empleado) por la flota de vehículos sea mínima.

El problema de planeación de rutas vehiculares con ventanas de tiempo – *Vehicle Routing Problem with Time Window* (VRPTW) – es un VRP que implica un intervalo de tiempo o lapso para cada cliente en el cual debe ser atendido.

El problema de planeación de rutas vehiculares con capacidades y ventanas de tiempo – *Capacitated VRP with time Windows* (CVRPTW) – busca diseñar un mínimo costo para las rutas en una flota de vehículos con capacidad homogénea

que sirven a diversos clientes con un límite de tiempo en la atención.

El problema de planeación de rutas vehiculares con flota heterogénea – *Fleet Size and Mix Vehicle Routing Problem* (FSMVRP) – es un VRP donde los costos y la capacidad de la flota de vehículos varían.

El problema de planeación de rutas vehiculares con recolección y entrega – *VRP with Pick-up and Deliveries* (VRPPD) – es un VRP con la posibilidad que los cliente pueden devolver determinados bienes, por tanto, se debe tener presente la capacidad del vehículo. Esta restricción hace más difícil el problema de planificación lo que puede causar una mala utilización de las capacidades de los vehículos, aumento en distancias recorridas o un numero mayor de vehículos. A este problema se le puede adicionar ventanas de tiempo asociadas a la recolección y entrega, problema conocido como VRPPDTW – *VRP with Pick-up and Deliveries and Time Windows* –.

El problema de planeación de rutas vehiculares con múltiples depósitos – *Multiple Depot VRP* (MDVRP) – es un VRP de depósito múltiple (cada uno con una flota de vehículos independiente) que debe servir a todos los clientes. Una de las variantes a este problema surge al tener en cuenta ventanas de tiempo asociadas a cada cliente llamado MDVRPTW – *Multiple Depot VRP with Time Windows* –.

El problema de planeación de rutas vehiculares con periodicidad – *Periodic VRP (PVRP)* – contempla en su espacio de operación un rango de m días, periodo durante el cual cada cliente debe ser visitado una vez. El mismo planteamiento del problema PVRP puede ser usado con ventanas de tiempo PVRPTW – *Periodic VRP with Time Windows* –.

El problema de planeación de rutas vehiculares estocástico – *Stochastic VRP (SVRP)* – es un VRP con uno o varios componentes aleatorios; clientes, demandas y tiempos estocásticos son las principales inclusiones en este tipo de problemas.

El problema de la planeación de rutas con entrega dividida – *Split Delivery VRP (SDVRP)* – es un VRP donde se permite que un cliente pueda ser atendido por varios vehículos si el costo total se reduce, lo cual es importante si el tamaño de los pedidos excede la capacidad de un vehículo. Al problema del SDVRP también se le puede contemplar ventanas de tiempo – *Split Delivery VRP with Time Windows* –.

El problema de recolección básico se puede relacionar con el CVRP. Así, se puede definir el problema de la recolección como en designar varias rutas para prestar un servicio a varios usuarios teniendo en cuenta la capacidad de la flota y la cantidad o demanda de capacidad por cada usuario. En la Figura 1 se puede observar la representación de un problema básico de recolección con múltiples depósitos.

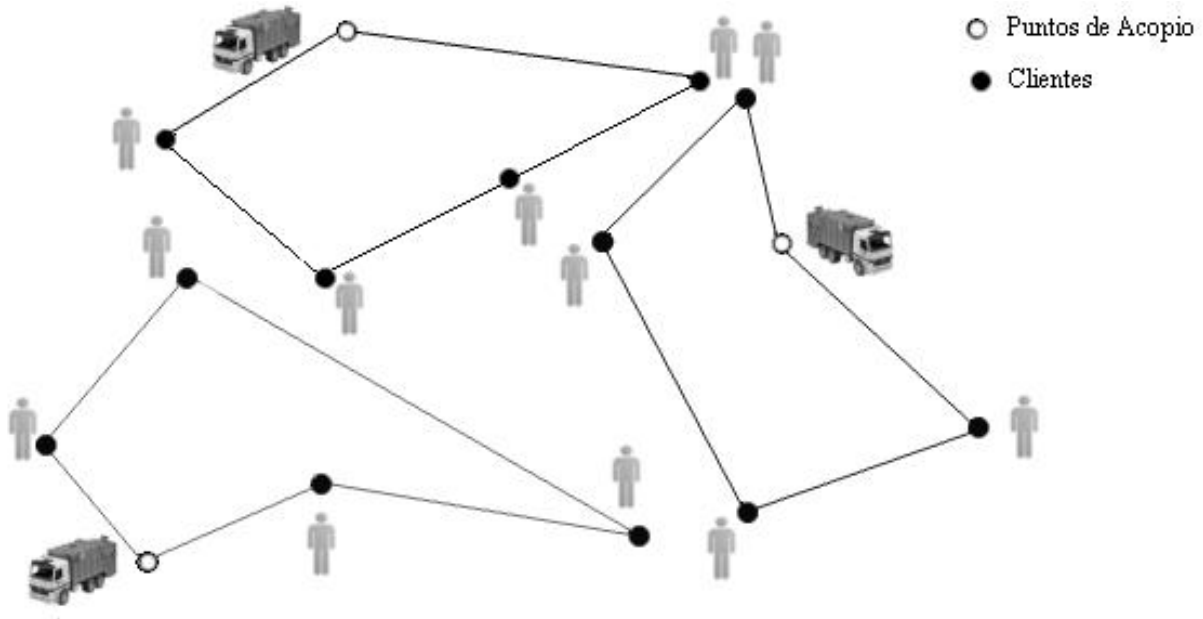


Figura 1. Representación del Problema de Recolección.

Las investigaciones más recientes para los problemas de planeación de rutas se centran especialmente con las metaheurísticas de algoritmos genéticos y de algoritmos de colonia de hormigas.

A. Algoritmos Genéticos

Los algoritmos genéticos están creados a partir de las teorías de Charles Darwin sobre la evolución por selección natural y sobre los resultados de la investigación de cruce genético de Gregory Mendel. Al fusionar la teoría de la evolución por selección natural, las leyes de herencia de Mendel y la hipótesis germinal de Weismann surge el concepto llamado neo-darwinismo [18].

En la actualidad los algoritmos genéticos están compuestos de tres etapas fundamentales:

- Selección: se escogen a los individuos más aptos para resolver el problema tratado.
- Cruce: tiene por objetivo mezclar la información genética de los padres escogidos para dar nacimiento a nuevos individuos más aptos para resolver el problema tratado.
- Mutación: se busca cambiar aleatoriamente la información genética de un individuo para tratar de darle más habilidad en resolver el problema tratado.

En los algoritmos genéticos hay funciones importantes a tener en cuenta para obtener una respuesta óptima, como lo son, la función de evaluación o *fitness* y la función de diversidad.

- Función de evaluación o *fitness*: a partir de esta función el algoritmo genético da el indicador de adaptabilidad de cada individuo, es decir, aquí cada individuo demostrará su capacidad para resolver el problema tratado.
- Función de diversidad: esta no es indispensable, pero puede ser desarrollada para mejorar la exploración del espacio de búsqueda del problema particular. Esta función tiene por objetivo la identificación de cada individuo para no permitirle estar dentro de la población ni en la misma generación. Es decir, con la función de diversidad se va a garantizar la unicidad de todos los individuos presentes dentro de la población en una generación determinada.

B. Algoritmo de Colonia de Hormigas

Los algoritmos de colonias de hormigas fueron creados por Marco Dorigo y están basados en el comportamiento de las hormigas naturales. Los biólogos y los entomólogos han descubierto la facilidad de las hormigas para encontrar siempre el camino más corto entre el hormiguero y la fuente de alimento, este comportamiento ha sido estudiado y se ha encontrado que las hormigas mantienen una comunicación indirecta con una sustancia volátil llamada feromona. Con la feromona las hormigas son capaces de crear una ruta y a través del tiempo optimizar sus recorridos obteniendo así un camino corto sin una visión global del terreno [3][4].

Los componentes de un algoritmo de colonia de hormigas son:

- Hormigas: tienen por objetivo realizar un recorrido (Ciclo Completo Cerrado) dentro del problema dado por el usuario.
- Memoria de feromonas: es utilizada para tener una historia de los recorridos efectuados por las hormigas y poder cambiar la intensidad de cada posible ruta a cada nueva iteración.
- Probabilidad de selección de un camino: toma de decisión de cada hormiga al momento de escoger un camino para pasar de una villa a otra. La fórmula que define la probabilidad es llamada regla de transición de estados aleatorios o regla de Bonabeau [4]:

$$p_{ij}^k(t) = \begin{cases} \frac{\tau_{ij}(t)^\alpha \cdot \eta_{ij}(t)^\beta}{\sum_{l \in J_i^k} \tau_{il}(t)^\alpha \cdot \eta_{il}(t)^\beta} & \text{si } j \notin J_i^k \\ 0 & \text{si } j \in J_i^k \end{cases} \quad (1)$$

La regla de Bonabeau (1) está basada en los conceptos de visibilidad (la visibilidad es el número inverso de la distancia entre dos villas i y j , $\eta_{ij} = \frac{1}{d_{ij}}$) y de la traza de feromona dejada sobre un arco del camino, es decir, la traza de feromona dejada entre una villa i y una villa j en una iteración t es $\tau_{ij}(t)^\alpha$, y b son dos parámetros que tiene el control sobre la importancia relativa de la intensidad de la feromona y la visibilidad, así se puede decir que con $a=0$ solamente la visibilidad es evaluada, por el contrario si $b=0$ solamente la intensidad de las feromonas es evaluada.

Como resultado de la evaluación de la regla se tiene una probabilidad para cada posible camino, $p_{ij}^k(t)$ es la probabilidad que la hormiga k tome la ruta hacia j desde i en la iteración t , la probabilidad solamente es calculada para las villas j que no han sido visitadas aún por la hormiga k .

Después de haber completado los recorridos de todas las hormigas escogidas es necesario dejar una cantidad de feromona sobre cada arco de cada recorrido en función de la calidad de la solución encontrada para cada recorrido, esta función es llamada impregnación de las feromonas, para ello se usa la regla de intensificación [4]:

$$\Delta \tau_{ij}^k(t) = \begin{cases} \frac{Q}{L^k(t)} & \text{si } (i, j) \in T^k \\ 0 & \text{si } (i, j) \notin T^k \end{cases} \quad (2)$$

En (2), $L^k(t)$ es la longitud del recorrido de la hormiga k en la iteración t y Q es una constante, la regla es aplicada a todos los arcos que pertenecen al recorrido realizado por la hormiga k en la iteración t .

Otra función importante es la función de atenuación de las feromonas, proceso que evapora las trazas de feromonas garantizando que la memoria global del sistema olvida las malas soluciones encontradas con la regla de evaporación enunciada en [4]:

$$\tau_{jj}(t+1) = (1-\rho) \cdot \tau_{jj}(t) + \Delta \tau_{jj}(t) \quad (3)$$

En (3) r es el índice de evaporación de la feromona, es decir, el índice con el cual la traza de feromonas se va a evaporar en relación con las iteraciones y $\Delta \tau_{ij}(t)$ es la función de impregnación.

III. ALGORITMO HÍBRIDO PROPUESTO

Los algoritmos híbridos tienen por meta explotar las mejores características de cada algoritmo para un determinado pedazo del problema tratado y dar una respuesta más óptima de forma global. Con los algoritmos híbridos se trata de establecer una colaboración entre dos o más algoritmos independientes, teniendo en cuenta las ventajas y desventajas de cada uno de ellos.

Algunas de las investigaciones que usan algoritmos híbridos en la planeación de rutas en TSP es [19] y en VRP se pueden enumerar [7][8][10][17].

En este trabajo se propone un algoritmo híbrido con el objetivo de minimizar las distancias recorridas en las rutas de recolección desde múltiples centros de acopio o depósitos a los lugares a visitar o clientes, teniendo en cuenta la capacidad de carga de la flota, la demanda de cada uno de los usuarios y el periodo de tiempo asignado. El algoritmo híbrido desarrollado está compuesto de un algoritmo genético como clasificador y un algoritmo de colonia de hormigas como optimizador local de cada subruta.

A. Algoritmo Genético

1) Representación del cromosoma

La representación del cromosoma es la base del desarrollo del algoritmo genético. Entre las formas más conocidas son la codificación binaria, la codificación real y la codificación Gray. En el desarrollo del algoritmo híbrido propuesto se usa la codificación real, porque los datos son tratados como una representación real del problema, es decir, los parámetros y el problema no son transformados.

2) Función de Fitness

El objetivo principal de la función de *fitness* es disminuir la longitud del recorrido total para la suma de todos los caminos posibles. Las distancias son evaluadas como distancias euclidianas. La representación del *fitnesses* es la suma de todos los recorridos descritos dentro del cromosoma evaluado.

3) Selección Elitista

A partir de la evaluación del recorrido se organizan los individuos de la población en orden ascendente, es decir que el individuo que ha realizado el menor recorrido será colocado

en la primera posición dentro del vector de individuos, después la primera mitad de la población va a pasar a la siguiente generación y la segunda mitad será eliminada.

4) Cruce

Una vez que los individuos son organizados y toda la población ha sido cortada, es necesario completarla para continuar con la ejecución del algoritmo, para eso, los individuos son cruzados unos con otros para dar nacimiento a nuevos individuos que deben pasar a la mutación. El cruce se hace entre la cabeza y la cola de la población, para dar nacimiento a individuos más diversos y no proporcionar toda la ventaja a los primeros individuos que son más capaces.

5) Mutación

La mutación se hace como factor aleatorio de cada nuevo individuo, esto favorece la exploración en el espacio de búsqueda y la diversidad de los individuos dentro de la totalidad de la población, además permite alcanzar un espacio más grande evitando una convergencia muy rápida hacia una solución local. El algoritmo propuesto usa la mutación por reemplazo, es decir, que se va a intercambiar dos valores dentro del mismo cromosoma.

6) Creación de grupos

Una vez el cromosoma ha pasado por el proceso de selección, cruce y mutación se crean los grupos que representan las posibles rutas respetando la dificultad de la demanda y la capacidad de carga. Los grupos son marcados al final con un menos uno (-1) y son creados sumando la demanda de cada cliente hasta llegar a un valor menor o igual a la capacidad de carga, si el valor de demanda es mayor a la capacidad de carga se deja el grupo con un único elemento al cual se debe ir varias veces (ver Figura 2).

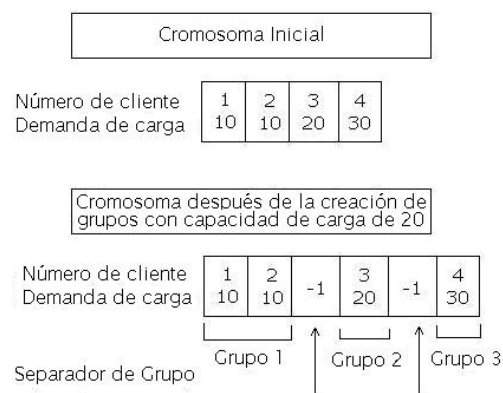


Figura 2. Ejemplo de creación dinámica de grupos

7) Función de Diversidad

El objetivo de la función de diversidad es garantizar la posible individualidad de cada individuo dentro de cada generación, es decir, no permitir individuos iguales dentro

de una misma generación. Una de las ventajas del algoritmo propuesto es ahorrar operaciones al momento de la comparación entre individuos, para ello cada individuo tiene un número asociado que se calcula con la ecuación (4). Este número se genera posteriormente a la creación de los grupos dinámicos, así la diversidad va a ser posible comparando solamente estos números.

donde $v = c[j]$ y $N = |c|$ (4)

$$\sum_{i=0}^{i < N} \sqrt{\sum_{i \neq 1} \sqrt{v}}$$

Para (4) c es el cromosoma, N es el tamaño del cromosoma y v es el valor de la posición i ésima dentro del cromosoma c .

B. Algoritmo de Colonia de Hormigas

1) Recepción de los datos

Los cromosomas creados deben pasar por el algoritmo de colonia de hormigas para ser optimizados localmente y tratar de obtener la mejor respuesta de la distribución obtenida con el algoritmo genético. Cada cromosoma de la población es enviado uno por uno como un vector al que se le aplica el algoritmo de colonia de hormigas.

2) Multi-Hilos

Para mejorar el rendimiento del algoritmo se implementó una variación multi-hilos capaz de poner varias hormigas a recorrer varias veces y por diferentes rutas del grafo asociado al vector enviado desde el algoritmo genético.

3) Flujo de los datos

Una vez el cromosoma ha sido optimizado localmente es necesario enviarlo de regreso hacia la población del algoritmo genético, para eso se copian los datos del vector obtenido y el valor de la distancia asociada que hacen parte de la estructura de la población.

Una representación implícita del algoritmo híbrido se puede observar en la Figura 3.

El algoritmo híbrido se puede observar en detalle a continuación (Algoritmo 1):

Algoritmo 1 Metaheurística híbrida

Creación de la población.

Evaluación de *fitness*.

Para $tg = 1, \dots, tgmax$

Selección de la población, selecciona los $(tampoblacion/2)$ mejores individuos.

Para tamaño de la población

Cruza los padres y obtiene 2 hijos por cada pareja de padres hasta completar el tamaño de la población (Cruce Uniforme).

Fin Para

Para número de individuos generados

Mutación de cada individuo por intercambio.

Fin Para

Creación de grupos dinámicos, según las restricciones de carga y asignación de punto de acopio para cada grupo creado.

Evaluación de diversidad según ecuación (4).

Envío de cada subgrupo al algoritmo de colonias de hormigas para optimización local.

Para $t = 1, \dots, tmax$

Para cada hormiga $k = 1, \dots, m$

Escoger una villa inicial aleatoriamente.

Para cada villa no visitada i

Escoger una villa j , dentro de la lista de villas restantes, según la ecuación (1).

Fin Para

Poner una pista de feromona sobre el trayecto conforme a la ecuación (2).

Fin Para

Evaporar las pistas según la ecuación (3).

Fin Para

Evaluación de *fitness*.

Fin Para

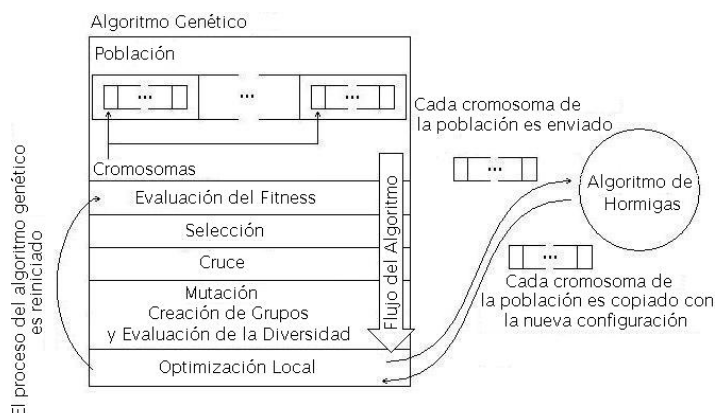


Figura 3. Representación implícita del Algoritmo

IV. MARCO EXPERIMENTAL

Se desarrolló una aplicación empleando el lenguaje C++ sobre el IDE Borland Turbo C++ Explorer 2006 en la que se implementa la simulación del algoritmo híbrido. La experimentación fue realizada sobre una máquina con procesador Intel Core2 Duo de 2Ghz y 3Gb de memoria RAM.

Los datos de entrada del algoritmo se definen de acuerdo a un formato en texto plano separado por comas (ver Figura 4), donde, el primer valor representa la etiqueta del punto o el nombre que identifica a cada punto, tanto en la presentación grafica como en la representación en modo texto de la ruta a seguir. El segundo y tercer valor son las coordenadas X e Y en

un plano cartesiano. El cuarto valor es el indicador de centro de acopio (1 ó 0), 1 para centro de acopio de lo contrario es un cliente al cual se le debe prestar el servicio. El quinto valor representa la demanda de carga en las mismas unidades en la cuales se especifica la capacidad de carga de los camiones. El quinto valor es la frecuencia de recolección que se puede especificar de dos formas, con un número que indica la frecuencia semanal o con un vector binario que indica los días exactos en los cuales se debe realizar la recolección. El número de registros en los datos de entrada identifican los lugares a visitar que representan un cliente o un centro de acopio.

Indicador de Centro de Acopio (1 es Centro, 0 No es Centro)		Demanda de Carga		Frecuencia de Recolección	
Etiqueta del Punto	Coordenada X	Coordenada Y			
P1	100	100	0	12	7
P2	200	100	0	10	6
P3	140	200	0	22	7
P4	200	300	0	10	7
P5	300	300	0	2	7
P6	220	23	0	10	[0011111]
P7	300	100	0	34	7
P8	300	210	0	20	7
P9	820	20	0	12	[0110111]
P10	330	540	0	31	7
P11	260	95	0	25	7
O1	150	50	1	0	0
O2	120	230	1	0	0

Figura 4. Datos de entrada para el algoritmo híbrido.

El problema de recolección para este trabajo además de ser un VRP básico contempla una flota de vehículos con capacidad homogénea, múltiples depósitos y un periodo de 1 a 7 días, esto significa que este problema particular tiene una mezcla de los siguientes tipos de problemas de ruteo de vehículos: CVRP, MDVRP y PVRP.

Debido a la mezcla de diferentes tipos de VRP en el problema presentado en este trabajo, para la experimentación se usaron conjuntos de datos específicos y con características particulares resumidas en la Tabla 1.

La aplicación en la que se implementó el algoritmo permite además seleccionar la opción de ejecución entre secuencial y multi-hilos, lo que permite aprovechar computadores en cuya arquitectura se pueden obtener mejores resultados de tiempos de respuesta con un método multi-hilos.

Tabla 1. Conjuntos de experimentación.

N = clientes, t = numero de depósitos, d = rango de demanda de los clientes, Q = capacidad homogénea de los vehículos de la flota y el Costo es la mejor solución conocida

Tipo VRP	Conjunto de datos	N	t	d	Q	Costo
CVRP ¹	A-n32-k5	32	1	1-24	100	784
MDVRP ²	P07	100	4	1-50	100	885.8
PVRP ²	P07	100	4	1-50	100	885.8

La aplicación implementada muestra los resultados del algoritmo con los itinerarios óptimos por cada día (ver Figura 5), además se puede observar la evolución y diversidad para cada día detallada por iteración (ver Figura 6).

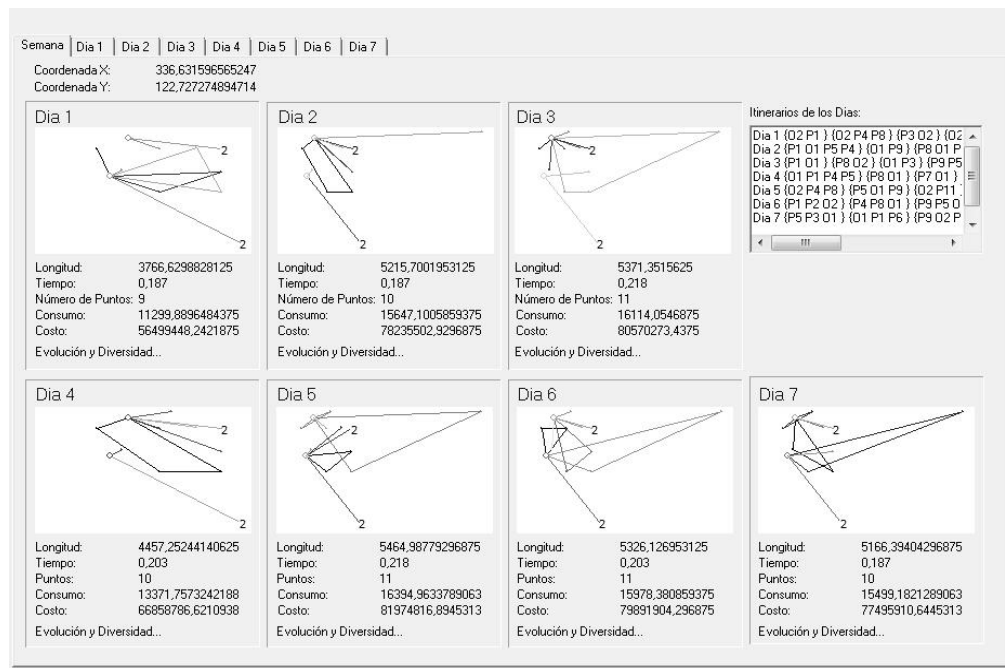


Figura 5. Representación de la solución más óptima encontrada por día.

V. RESULTADOS

- 1) El resultado del comportamiento del algoritmo híbrido para el CVRP con el conjunto A-n32-k5 (ver Tabla 1) con la siguiente configuración de parámetros se puede observar en la Figura 7.

Algoritmo genético:

Número de Iteraciones = 2000

Número de Individuos = 50

Algoritmo de Colonia de Hormigas:

Número de Iteraciones = 10

Número de hormigas= 4

$a = 1$; $b = 5$

CVRP:

Número de Puntos Acopio = 1

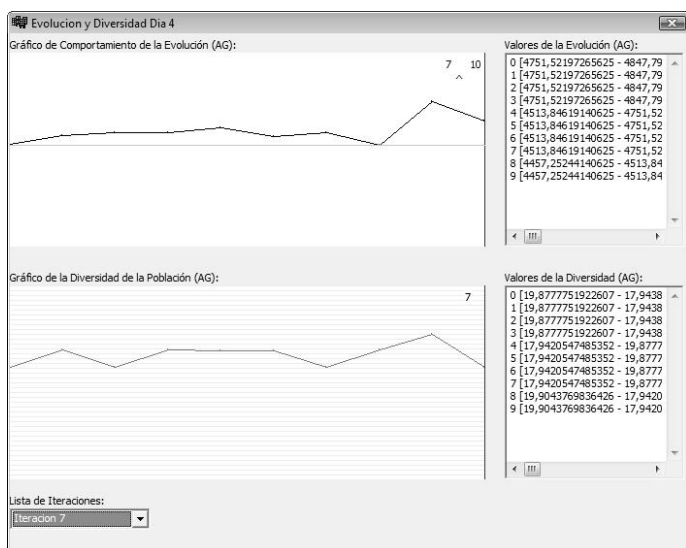


Figura 6. Representación de la evolución y diversidad de una iteración.

La Tabla 2. muestra el resumen de los resultados del algoritmo híbrido después de 10 iteraciones. La Figura 8. y la Figura 10. presentan el tiempo empleado a medida que se encuentran mejores soluciones para cada caso en particular. Este tiempo crece debido al aumento en el número de generaciones.

Tabla 2. Resumen de resultados de algoritmo híbrido CVRP.

	Mejor Solución	Promedio
Costo	803,6	822,18
Tiempo(seg.)	2603	2734,83

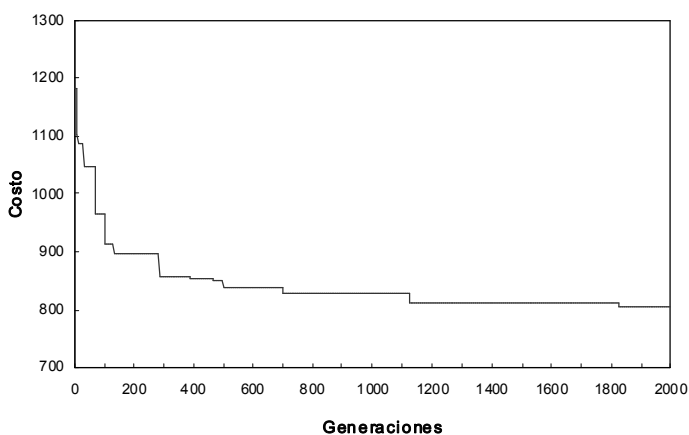


Figura 7. Costo vs. Generaciones para CVRP.

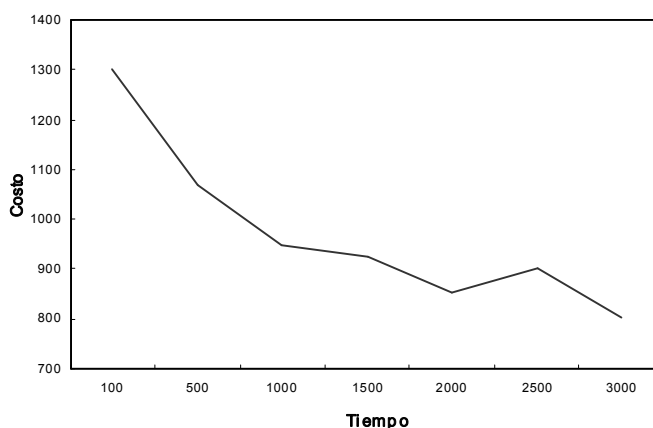


Figura 8. Costo vs. Tiempo para CVRP.

- 2) El resultado del comportamiento del algoritmo híbrido para el MDVRP y PVRP con el conjunto P07 (ver Tabla 1) con la siguiente configuración de parámetros se puede observar en la Figura 9.

Algoritmo genético:

Número de Iteraciones = 2000

Número de Individuos = 50

Algoritmo de Colonia de Hormigas:

Número de Iteraciones = 10

Número de hormigas = 4

$a = 1$; $b = 5$

MDVRP y PVRP:

Días = 4

Número de Puntos Acopio = 4

Tabla 3. Resumen de resultados de algoritmo híbrido MDVRP y PVRP.

	Mejor Solución	Promedio
Costo	923,7	984,6
Tiempo(seg.)	3571	3597,7

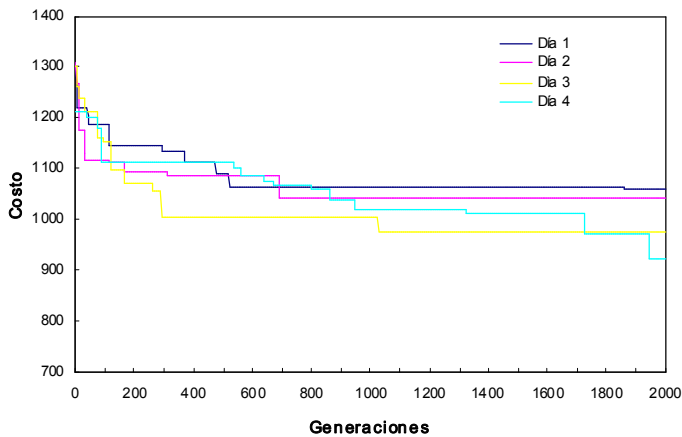


Figura 9. Costo vs. Generaciones para MDVRP y PVRP.

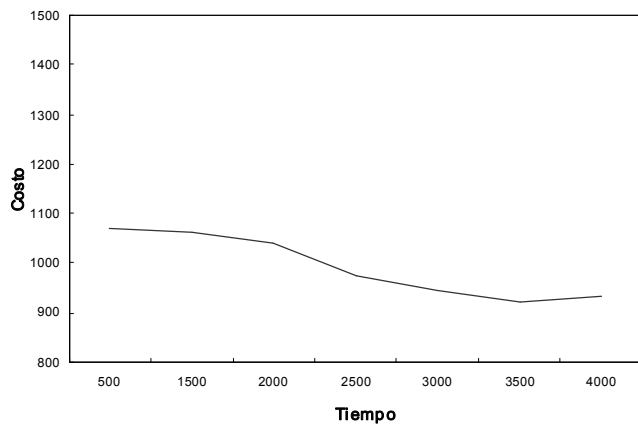


Figura 10. Costo vs. Tiempo para MDVRP y PVRP.

Basado en los anteriores experimentos se puede analizar que el algoritmo propuesto es capaz de lograr buenas soluciones para el problema de enrutamiento de vehículos. Para el primer caso experimental con datos asociados a CVRP, la mejor solución encontrada tiene un costo de 803,6 mientras que la solución óptima tiene un costo de 784. El segundo caso experimental con datos asociados a la combinación de MDVRP y PVRP se encontró que la mejor solución es 923,7 mientras que la óptima tiene un costo de 885,8.

VI. CONCLUSIONES

El desarrollo del algoritmo propuesto en un problema de recolección busca planificar las rutas más óptimas entre todos los lugares a visitar teniendo en cuenta su demanda y considerando una flota de vehículos con capacidad homogénea, múltiples centros de acopio o depósitos y un periodo de m días, esto significó una buena aproximación en la aplicación de metaheurísticas híbridas para encontrar soluciones óptimas. Es importante realizar estudios más profundos para añadir más restricciones que se puedan presentar en sistemas de recolección.

La interfaz gráfica realizada es una herramienta importante para visualizar las rutas obtenidas, seguir el comportamiento y la diversidad del algoritmo que se generan a lo largo del número de iteraciones que fueron ingresadas por el usuario, suministrar información de los resultados, configurar manualmente los parámetros del algoritmo y servir como apoyo estadístico en la toma de decisiones para empresas con sistemas de recolección.

En este artículo se presentaron resultados preliminares que demuestran experimentalmente que el comportamiento del algoritmo híbrido a través de las generaciones mejora el costo. Se puede observar que las mejores soluciones encontradas del algoritmo híbrido son cercanas a las soluciones óptimas conocidas para cada uno de los tipos de VRP usados en la experimentación.

REFERENCIAS

- [1] Balseiro, S., 2007. Logística y distribución: Algoritmos para problemas de ruteo de vehículos con restricciones de capacidad y ventanas de tiempo. Tesis de Maestría. Universidad de Buenos Aires, Argentina.
- [2] Berbeglia, G., Cordeau, J. F., Gribkovskaia, I. y Laporte, G., 2007. Static Pickup and Delivery Problems: A Classification Scheme and Survey. *Journal TOP*, Vol. 15(1), pp. 45 – 47.
- [3] Dréo, J., 2003. Adaptation de la méthode des colonies de fourmis pour l'optimisation en variables continues. Application en génie biomédical. Tesis de Ph D. Universidad de Paris XII, Francia.
- [4] Dréo, J., Pétrowsky, A., Siarry, P. y Taillard, É., 2003. Métaheuristiques pour l'optimisation difficile. Paris: Eyrolles Editions, Francia.
- [5] González, G. y González, F., 2006. Metaheurísticas aplicadas al ruteo de vehículos. Un caso de estudio. Parte 1: formulación del problema. *Ingeniería e Investigación*, Vol. 26(3), pp. 149 – 156.
- [6] Hanshar, F., y Ombuki-Berman, B., 2007. Dynamic vehicle routing using genetics algorithms. *Applied Intelligence*, Vol. 27(1), pp. 89 – 99.
- [7] Ho, W., Ho, G., Ji, P. y Lau, H., 2008. A hybrid genetic algorithm for the multi-depot vehicle routing problem. *Engineering Applications of Artificial Intelligence*, Vol. 21, pp. 548 – 557.
- [8] Kammarti, R., Hammadi, S., Borne, P. y Ksouri, M., 2004. A new hybrid evolutionary approach for the pickup and delivery problem with time windows. En: IEEE International Conference on Systems, Man and Cybernetics, Vol. 2, pp. 1498 – 1503.
- [9] Larson, R. y Odoni, A., 1981. Urban Operation Research. New Jersey: Prentice-Hall, USA.
- [10] Le Bouthillier, A., Crainic, T. y Kropf, P., 2005. A Guided Cooperative Search for the Vehicle Routing Problem with Time Windows. *Journal IEEE Intelligence Systems*, Vol. 20(4), pp. 36 – 42.
- [11] Manfrin, M., 2004. Ant Colony Optimization for the Vehicle Routing Problem. Tesis de Maestría. Universidad Libre de Bruselas, Bélgica.
- [12] Martello, S., Osman, I. H., Voss, S. y Roucairol, C., 1999. Meta-Heuristics: Advances and Trends in Local Search Paradigms for Optimization. Boston: Kluwer Academic Publishers, USA.
- [13] Miller, C. y Tucker, R., 1960. Integer programming formulation of traveling salesman problems. *Journal de ACM* Vol. 7(4), pp. 326 – 329.
- [14] Potter, T. y Bossomaier, T., 1995. Solving vehicle routing problems with genetic algorithms. En: IEEE International Conference on Evolutionary Computation, Vol. 2(29), pp. 788 – 793.
- [15] Rebreyend, P., 1999. Algorithmes génétiques hybrides en optimisation combinatoire. Tesis de Ph D, Escuela Normal Superior de Lyon, Francia.
- [16] Rizzoli, A., Montemanni, R., Lucibello, E. y Gambardella, L. M., 2007. Ant colony optimization for real-world vehicle routing problems. *Swarm Intelligence*, Vol. 1(2), pp. 135 – 151.
- [17] Tan, K. C., Chew, Y. H. y Lee, L. H., 2006. A hybrid multi-objective

evolutionary algorithm for solving truck and trailer vehicle routing problems. *European Journal of Operational Research*, Vol. 172, pp. 855 – 885.

- [18] Toscano, G., 2001. Optimización Multiobjetivo usando un Micro Algoritmo Genético. Tesis de Maestría. Universidad Veracruzana, México.
- [19] Wang, Y. P., Li, Y. H. y Dang, C. Y., 2004. A novel globally convergent hybrid evolutionary algorithm for travelling salesman problem. En: *International Conference on Machine Learning and Cybernetics*. Vol. 4, pp. 2485 – 2489.

Daniel Soto. Obtuvo su título de Ingeniero de Sistemas en la Universidad San Martín. Actualmente trabaja en proyectos de robótica para la empresa privada. Sus áreas de interés son los algoritmos y las metaheurísticas.

Wilson Soto. Ingeniero de Sistemas y Especialista en Ingeniería de Software. Actualmente esta desarrollando su tesis de Maestría en Ingeniería de Sistemas y Computación en la Universidad Nacional de Colombia. Hace parte del grupo Algoritmia y Combinatoria ALGOS-UN de la Universidad Nacional de Colombia. Sus principales áreas de interés son la algoritmia, bioinformática e ingeniería del software.

Yoan Pinzón Ingeniero de Sistemas, Especialista en Ingeniería de Software, Master en Ingeniería de Sistemas y Computación Avanzada de la Universidad de Londres, Reino Unido y Doctor de Filosofía de la Universidad de Londres, Reino Unido. Director del grupo Algoritmia y Combinatoria ALGOS-UN de la Universidad Nacional de Colombia. Actualmente se desempeña como profesor asociado en el Departamento de Ingeniería de Sistemas en la Universidad Nacional de Colombia. Sus principales áreas de interés son la algoritmia, bioinformática y criptografía.