

# Solución al Problema de Ruteo de Vehículos con Capacidad Limitada (CVRP) usando una técnica metaheurística

## Solution to the Capacitated Vehicle Routing Problem (CVRP) using a metaheuristic technique

Juan Pablo Orrego Cardozo, Daniela Ospina Toro, Eliana Mirledy Toro Ocampo  
*Ingeniería Industrial, Universidad Tecnológica de Pereira, Pereira, Colombia*  
 jporrego@utp.edu.co, [dospina@utp.edu.co](mailto:dospina@utp.edu.co), [elianam@utp.edu.co](mailto:elianam@utp.edu.co)

**Resumen**— El problema de ruteo de vehículos (VRP) es uno de los temas más importantes en las redes de distribución; con diversas aplicaciones en transporte, logística y telecomunicaciones. Debido a la dificultad que implica resolver instancias de gran tamaño con métodos exactos, la mayoría de los investigadores se han centrado en resolver problemas de la vida real a través de técnicas metaheurísticas. En este trabajo, se generan soluciones aproximadas para el CVRP tradicional usando la heurística de barrido de dos fases y resolviendo cada problema del agente viajero con el algoritmo genético modificado de Chu-Beasley. Se valida la metodología con casos de prueba de la literatura especializada.

**Palabras clave**— algoritmo genético, agente viajero, heurísticas, logística, metaheurísticas, problema de ruteo de vehículos.

**Abstract**— The Vehicle Routing Problem (VRP) is one the pivotal topics in distribution networks. VRP has many applications on transport, logistics and telecommunications. Due the difficulty in solving large instances with exact methods, for solving real-life problems most researchers have been focused on metaheuristics. In this work, approximate solutions to the traditional CVRP are generated using a two phase heuristic, sweep, and solving each traveling salesman problem with the modified genetic algorithm of Chu-Beasley. The methodology is validated using benchmark instances from specialized literature.

**Key Word**— genetic algorithm, heuristics, logistic, metaheuristics, traveling salesman problem, vehicle routing problem.

## I. INTRODUCCIÓN

Debido al gran interés científico en los problemas de ruteo de vehículos, estos problemas han sido objeto de investigación intensiva por más de 50 años. Esto ocurre debido a que son NP-completos y porque algunas instancias de tamaño medio, no han podido ser resueltas de forma exacta [1]. Por otra parte,

también es de gran interés por la variedad de aplicaciones que tiene en diversas áreas como: transporte, logística, comunicaciones, manufactura, militar, entre otros.

El problema tradicional CVRP (por sus siglas en inglés *Capacitated Vehicle Routing Problem* ó *Problema de Ruteo de Vehículos con Capacidad Limitada*) puede ser descrito en su forma más sencilla como una flota de vehículos con capacidades uniformes que tiene que satisfacer la demanda de un grupo de clientes a través de un conjunto de rutas que empiezan y terminan en un almacén común y que representan el menor costo posible, así como la identificación del orden de visita a los mismos [2].

En la Tabla 1, se presenta una revisión del estado del arte con base en artículos de la literatura especializada. Dicho recorrido se hace desde el año de 1969, donde se aborda el tema por primera vez como el *Problema de Despacho de Vehículos*, hasta el 2011. A través de los años, se generaliza como *VRP*. Además, se observa que las técnicas de solución para resolver este problema van desde las exactas exactas Branch and Bound, ramificación y corte, Branch and Cut, y el Branch and Cut and Price; hasta las aproximadas, como son las metodologías heurísticas y metaheurísticas, entre las que se encuentran: las de agrupar primero y rutear después, ahorros, algoritmos genéticos, búsqueda tabú, recocido simulado, entre otras.

Año	Bibliografía	Técnicas de Solución
1969	[3]	Se consideran tres métodos de solución, siendo el último el de mejores resultados: (1) Aproximación de <i>Ramificación y Acotamiento</i> . (2) Aproximación de los ahorros. (3) El método de los 3 tours óptimos.
1993	[4]	Rutear primero y asignar después,

		utilizando un tour gigante generado por la técnica de Christofides.
1993	[5]	Relajación del CVRP a través del <i>Ruteo de Vehículos Gráfico (GVRP)</i> y además extendiendo los resultados poliédricos conocidos para el TSP.
1994	[6]	Se modela el VRP como un problema para encontrar el costo mínimo de un Árbol-K, con dos extremos K incidentes en un depósito, e imponiendo restricciones paralelas de la capacidad del vehículo y que cada cliente debe ser visitado una vez. Estas restricciones paralelas son dualizadas para obtener un problema Lagrangiano que provee los límites inferiores para ejecutar un algoritmo de <i>Ramificación y Acotamiento</i> .
1998	[7]	Implementación del algoritmo <i>Branch-and-Cut</i> , basado en la descripción parcial poliédrica del polígono correspondiente. Además, se centra en el diseño de procedimientos de separación para varias clases de desigualdades válidas.
2001	[8]	Metodología de separación basada en descomposición para las restricciones de capacidad. Implementaciones a través de estructuras paralelas <i>Branch and Cut, and Price SYMPHONY</i>
2002	[9]	Heurística de rutear primero y asignar después en grafos mixtos.
2002	[10]	Discute las metaheurísticas que han sido aplicadas para resolver el CVRP en el tiempo: recocido simulado, recocido determinístico, búsqueda tabú, algoritmos genéticos, colonias de hormigas y redes neuronales.
2003	[11]	Algoritmo genético híbrido, evolucionando dos poblaciones de soluciones para minimizar la distancia total recorrida y utilizando diversos operadores genéticos.
2004	[1]	Algoritmo genético híbrido sin delimitadores de rutas y potencializado con operadores de búsqueda local, utilizando el operador OX para la recombinación.
2006	[12]	Algoritmo híbrido entre el uso de la

		optimización discreta de la partícula swarm, utilizada para la búsqueda óptima de los vecindarios globales y locales, y el recocido simulado con ciertas probabilidades para evitar quedar atrapado en óptimos locales.
2008	[13]	Algoritmo de búsqueda local para el VRP, basado en la exploración de vecindarios de orden exponencial resolviendo un problema de programación lineal y una heurística de refinamiento propuesta por Franceschi et al.
2009	[14]	Rutear primero y asignar después, utilizando un tour gigante generado por heurísticas de arranque y mejoradas a través de la búsqueda Tabú. Posteriormente la planificación de la flota de vehículos fue realizada mediante la técnica de <i>scheduling</i> .
2011	[15]	Asignar primero y rutear después, utilizando las heurísticas de Clark and Wright y barrido para formar los clusters, y posteriormente resolver cada TSP usando el algoritmo de cúmulo de partículas.

**Tabla 1:** Revisión del estado del arte. [Fuente Propia]

Por medio de la tabla 1, se plantea la necesidad de presentar una metodología de solución basada en la heurística constructiva de barrido de dos fases; donde inicialmente se agrupen un conjunto de clientes que serán satisfechos por un camión dado y luego se generen las rutas óptimas a través del Algoritmo Genético de Chu-Beasley (AGCB) para cada vehículo. Esto con el fin de minimizar los costos de recorrido para todas las rutas mínimas necesarias.

## II. FORMULACIÓN MATEMÁTICA DEL PROBLEMA

El desarrollo del CVRP, siendo el caso más general, depende en gran medida de la fundamentación y desarrollo de dos problemas de optimización del tipo NP-Completo; el Bin Packing Problem (BPP) y el problema del agente viajero (TSP). Las principales formulaciones matemáticas utilizadas para representar el VRP han sido propuestas en la literatura según [16].

El primero asociado específicamente con el CVRP determina el número mínimo de vehículos. Cada vehículo tiene capacidad  $C$ , y son requeridos para satisfacer los  $n$  clientes, cada uno con una demanda no negativa. Esto está dado por la

relación de la sumatoria total de las demandas ( $d(S)$ ), sobre la capacidad de cada camión disponible ( $C$ ).

$$\left\lceil \frac{d(S)}{C} \right\rceil$$

El segundo caso es el de un agente viajero que debe realizar un recorrido partiendo de una ciudad de origen, pasando por cada una de las ciudades una sola vez, y regresando a la ciudad de origen. Todo esto minimizando los costos totales de recorrido del viajero. Así, el modelo matemático asume la siguiente forma [17]:

$$(1) \quad \text{Min } v = \sum_{(i,j) \in A} c_{ij} x_{ij}$$

Sujeto a:

$$(2) \quad \sum_{\{i:(i,j) \in A\}} x_{ij} = 1 \quad \forall j \in V$$

$$(3) \quad \sum_{\{j:(i,j) \in A\}} x_{ij} = 1 \quad \forall i \in V$$

$$(4) \quad \sum_{\{(i,j) \in A: i \in U, j \in (V-U)\}} x_{ij} \geq 1; \quad 2 \leq |U| \leq |V| - 2$$

Donde  $V$  es el conjunto de ciudades (vértices),  $A$  el conjunto de caminos (arcos) y  $U \subset V$ . El conjunto de restricciones (2) indica que se puede llegar a una ciudad dada desde una sola ciudad anterior y el conjunto (3) de restricciones indica que a partir de una ciudad se puede pasar a una única ciudad. El conjunto (4) de restricciones evita el surgimiento de subtours [17].

Posterior a la conceptualización teórica y matemática de ambos problemas base para el desarrollo del CVRP, es posible presentar la principal formulación matemática usada para modelar dicho problema, llamado modelo de flujo de vehículos (*Vehicle Flow Model*).

Los modelos de este tipo usan variables enteras, asociadas a cada arco o extremo del grafo, el cual cuenta el número de veces que el arco o extremo es atravesado por un vehículo. Siendo este el modelo más usado para las formulaciones básicas del VRP, generalmente es utilizado para los casos en los cuales el costo de una solución puede ser expresada como la suma de los costos asociados a los arcos, y cuando las restricciones más relevantes conciernen la transición directa entre los clientes y la ruta, por lo tanto, este puede ser modelado efectivamente a través de una definición apropiada del conjunto de arcos y sus costos. Por otra parte, existen casos muy prácticos en los cuales estos modelos no deberían ser utilizados, como: cuando el costo de una solución depende de la secuencia total de los vértices, o del tipo de vehículo asignado a la ruta.

Este modelo de programación lineal del tipo de formulación de flujo de vehículos de dos índices (*two-index vehicle flow formulation*) utiliza  $O(n^2)$  variables binarias de  $x_{ij}$  para indicar con un valor tomado de 1 si el arco  $(i, j) \in A$  pertenece a la solución óptima y toma un valor de 0 en otro caso; es decir, toma su valor si el vehículo atraviesa o no un arco en la solución óptima. Para simplificar las notaciones, en las siguientes formulaciones se asume que el grafo  $G(V, A)$  (ó  $G(V, E)$ ) es completo [16].

$$(5) \quad \text{min } \sum_{i \in V} \sum_{j \in V} c_{ij} x_{ij}$$

Sujeto a:

$$(6) \quad \sum_{i \in V} x_{ij} = 1 \quad \forall j \in V \setminus \{0\}$$

$$(7) \quad \sum_{j \in V} x_{ij} = 1 \quad \forall i \in V \setminus \{0\}$$

$$(8) \quad \sum_{i \in V} x_{i0} = K,$$

$$(9) \quad \sum_{j \in V} x_{0j} = K,$$

$$(10) \quad \sum_{i \in S} \sum_{j \in S} x_{ij} \geq r(S) \quad \forall S \subset V \setminus \{0\}, S \neq \emptyset,$$

$$(11) \quad x_{ij} \in \{0, 1\} \quad \forall i, j \in V.$$

Donde:

- $V = \{v_0, v_1, \dots, v_n\}$  es el conjunto de vértices del gráfico, donde  $v_0$  corresponde al almacén.
- $C$  es la matriz de distancias o costos  $c_{ij}$  entre los clientes  $v_i$  y  $v_j$ .
- $K$  es el número de vehículos de capacidad  $C$ , necesarios para cargar toda la demanda.
- $S$  es un subconjunto de vértices del grafo donde  $S \subset V \setminus \{0\}$ .
- $r(S)$  es el número mínimo de vehículos necesarios para servir a todos los clientes en  $S$ .

Las restricciones de grado de entrada y de salida (6) y (7), imponen que exactamente un arco entra y otro sale de cada vértice asociado a un cliente, para satisfacer que todos los clientes sean visitados una vez. Además, las restricciones (8) y (9) imponen los requerimientos para el vértice depósito; es decir, los mismos vehículos que salen del depósito, regresan al depósito. Finalmente, la restricción de *corte de capacidad* (10) satisface tanto la conectividad de la solución y las no violaciones de la capacidad  $C$  de cada vehículo. Dicha restricción resta aún válida si el  $r(S)$  es reemplazado por el valor solución trivial del BPP mencionado anteriormente.

### III. TÉCNICAS DE SOLUCIÓN

Entre las técnicas de solución aproximadas para el CVRP, las más frecuentemente utilizadas son las heurísticas y metaheurísticas, con las cuales se pueden obtener buenas soluciones en tiempos de cómputo razonables. Por otra parte, aun cuando la característica de este problema es NP-completo, diversas técnicas exactas han sido exploradas satisfactoriamente a través del tiempo, logrando avances y mejoras en cuanto a tiempos de cómputo para obtener las soluciones óptimas a los problemas tratados. A continuación, se relacionan diversas metodologías de solución.

#### A. Técnicas exactas (PLE)

Los algoritmos exactos son aquellos que siempre producen una solución óptima. Dichas técnicas no son adecuadas en aplicaciones que requieren soluciones rápidas o que tratan de resolver instancias de problemas muy grandes. Debido a la naturaleza NP del problema VRP, la búsqueda exhaustiva de estas técnicas no resulta eficiente computacionalmente [18]. Entre las que se encuentran: Branch & Bound, Branch & Cut, Branch & Cut & Price y la programación dinámica. [6], [7], [19].

#### B. Técnicas heurísticas

Las técnicas denominadas como heurísticas, realizan una exploración relativamente limitada en el espacio de búsqueda y comúnmente producen soluciones de buena calidad dentro de tiempos de cómputo razonables. En un macro-nivel, las heurísticas para el VRP están divididas entre 3 tipos.

##### 1. Heurísticas constructivas

Las ideas detrás de la mayor parte de este tipo de heurísticas se encuentran muy bien documentadas en [18], [19], [20]. Entre estas se encuentran frecuentemente el método del ahorro o Clark & Wright, el vecino más cercano (aleatorio o no), inserción menos costosa, inserción del más lejano, algoritmo goloso, entre otras.

##### 2. Heurísticas de dos fases

Estas técnicas, parten de un problema y una solución “vacía” para que a partir de ella se pueda construir una solución factible, pero que casi nunca resulta óptima. Para la obtención de la solución, las heurísticas de dos fases dividen al problema de VRP en dos etapas: una de asignación de clientes a vehículos y otra para la determinación del orden de visita a dichos clientes. Existen luego dos grandes grupos de solución, y a los cuales pertenecen diversos métodos específicos; dichos grupos son los de agrupar primero y

rutear después, o los de rutear primero y asignar después. [18], [19].

##### 3. Heurísticas de mejora iterativa

Una vez que se tiene una solución para un problema VRP, se puede intentar mejorar mediante algún procedimiento de búsqueda local. Para cada solución  $s$  se define un conjunto de soluciones vecinas  $N(s)$ . Un procedimiento de búsqueda local parte de una solución  $s$  la reemplaza por una solución  $s' \in N(s)$  de menor costo y repite el procedimiento hasta que la solución no pueda ser mejorada. Al terminar, se obtiene una solución localmente óptima respecto a la definición de la vecindad. [18], [21]. Entre ellas se encuentran:  $\lambda$  – intercambio, el operador or-opt, operadores string exchange y string relocation, GENI y GENIUS, transferencias cíclicas, intercambios  $r-opt$ , mecanismos unstringing y stringing (US) [2].

#### C. Técnicas metaheurísticas

El énfasis en las técnicas metaheurísticas está en realizar una exploración profunda de las regiones más prometedoras del espacio solución. La calidad de las soluciones producidas por estos métodos es mucho mejor que la obtenida por las heurísticas clásicas, aunque su tiempo de cómputo es por lo general mucho mayor. Existen dos tipos principales de familias de técnicas heurísticas avanzadas (*metaheurísticas*), que son, las basadas en métodos de búsqueda local, y las que explotan las poblaciones de soluciones

##### I. Métodos de búsqueda local

Su principio de funcionamiento de base parte del hecho que hay una solución inicial realizable y a esta se le aplican una serie de modificaciones locales a la solución. Entre tanto, dichas modificaciones mejoren el objetivo del problema en estudio. Entre los principales métodos de búsqueda local se encuentran: búsqueda local simple, búsqueda local múltiple, recocido simulado, Búsqueda de Vecindario Variable (VNS), Búsqueda de Vecindario Variable Descendente (VND), métodos de búsqueda con tabúes (simples y probabilísticos) [22].

##### II. Métodos de exploración poblacional

Entre los principales métodos basados en manejo de poblaciones se encuentran: algoritmos genéticos (variación Chu-Beasley), algoritmos meméticos, optimización mediante cúmulo de partículas (Swarm), optimización por colonia de hormigas [22].

#### IV. PLANTEAMIENTO DEL PROBLEMA

La metodología de solución implementada para el desarrollo del trabajo actual es la de dos fases, mencionada anteriormente. Es decir, está guiada inicialmente por la aplicación de la heurística de barrido o *sweep* [18], [19] generando los clusters de afectación por vehículo, y luego la solución a cada uno como un problema de agente viajero (TSP) a través del AGCB. A continuación, se hace referencia a cada una de las fases y posterior análisis detallado de cada una de ellas.

**Fase 1:** Creación de los clusters de afectación para cada uno de los vehículos, dependiendo de su capacidad y ubicación de los clientes respecto al depósito.

**Fase 2:** Se da solución a cada cluster establecido en la etapa anterior, implementando el AGCB adaptado a un problema de optimización del agente viajero simétrico.

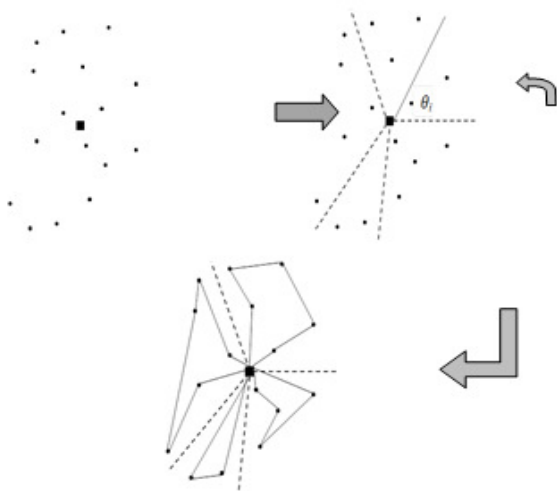


Figura 1. Implementación fase 1. [Fuente Propia].

##### A. Implementación de la fase 1

En la figura 1, se puede observar de forma gráfica la metodología de agrupamiento de los clientes, que se hace por medio de sus ángulos respecto a una recta arbitraria y la suma de sus demandas hasta que se llena el camión, evaluando de forma iterativa las posibilidades de incluir clientes posteriores al que viola la restricción de capacidad inmediatamente. Cuando un cliente de los posteriores evaluados es agregado sin violar la restricción de capacidad, aquel que es ignorado, vuelve a quedar de primero en la lista y así se comienza nuevamente el proceso de agrupamiento, hasta utilizar todos los camiones. Durante el transcurso del procedimiento, se generan diversos cambios de la recta arbitraria de arranque para explorar aún más el espacio de soluciones y generar diferentes clusters en cada iteración para a mayor diversidad.

##### B. Implementación de la fase 2

En esta segunda, se optimiza la ruta de cada camión por cluster de forma separada. Esto se hace resolviendo un problema del agente viajero TSP, implementando un AGCB [23] y potencializándolo con la calibración de parámetros para éste tipo de códigos de manejos poblacionales propuesto por [1] y con algunas claves parciales aleatorias para algoritmos genéticos “*Biased Random-Key Genetic Algorithms*” propuestas por [24].

##### 1. Codificación

Cada vector representa la solución óptima del recorrido del vehículo para el cluster  $i = 1, \dots, n$  y finalmente cada elemento “gen” del vector representará el orden secuencial en que serán visitados cada uno de los clientes de dicho agrupamiento. El tamaño de cada solución  $n_i$  depende del número de clientes pertenecientes a cada grupo, limitado por la demanda de los mismos y la capacidad del vehículo.



Figura 2. Esquema de codificación.

##### 2. Población inicial

Un total de 30 individuos en la población inicial. 7 generados a través de heurísticas constructivas (Clark & Wright, vecino más cercano aleatorio y algoritmo goloso). Y los 13 restantes generados de forma aleatoria.

##### 3. Selección y recombinación

En la figura 3 se esquematiza el proceso de selección para éste problema, donde la población es dividida en dos grupos, un pequeño grupo de individuos élite  $p_e$ , que son aquellos 7 que tienen los mejores valores de función objetivo, y otro conjunto de individuos no-élite, siendo estos los faltantes de la relación entre  $p - p_e$ . Ahora bien, el elemento padre es seleccionado de la partición de soluciones élite de la generación actual, y el otro de la partición no-élite, ambas selecciones de forma aleatoria. Estos son los únicos dos elementos que compartirán material genético en la siguiente etapa para dar paso a un nuevo elemento en la generación  $k+1$ .

Posterior a la selección, se inicia otra etapa del proceso denominada recombinación, es aquí donde se genera el intercambio de material genético entre los dos padres seleccionados, el de la población élite y no-élite para este caso.

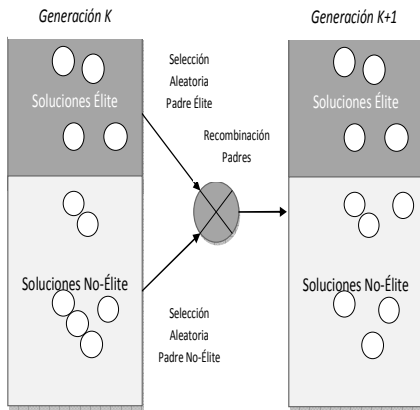


Figura 3. Esquema de selección. Adaptado de [24].

En esta metodología implementada, *Recombinación Parametrizada Uniforme* [25], se establece  $p_e = 0.7$  como la probabilidad que tiene un hijo de adquirir el material genético proveniente de las componentes del vector de su padre élite. Por tanto, se genera un vector aleatorio, que indicará el gen nuevo del hijo que información tendrá y de qué padre. Es decir, si un gen del vector aleatorio posee un valor inferior a  $p_e$ , el hijo adquiere el material genético del padre 1 en la posición evaluada, ó del padre 2 en caso contrario.

En caso de haber repetición de los elementos que ya están incluidos en la solución en curso, se dejan dichos genes vacíos, y al final se completan de forma trivial con el cliente no visitado más cercano al cliente inmediatamente anterior. En la figura 4, se presenta un ejemplo sencillo de dicho procedimiento.

P1 =	3	1	5	4	2
P2 =	1	2	4	5	3
Aleatorios =	0.5	0.2	0.9	0.8	0.1
Hijo =	3	1	4	5	2

Figura 4. Proceso de recombinación.

#### 4. Mutación

En cada iteración, se realiza después de la recombinación un proceso de mutación simple a un elemento aleatorio de toda la población de la solución en curso. Esta mutación es un intercambio de posición o *swap* de dos elementos aleatorios de un cromosoma. Esto utilizado como un mecanismo de perturbación, con el fin de explorar mejor el espacio de soluciones y escapar de óptimos locales.

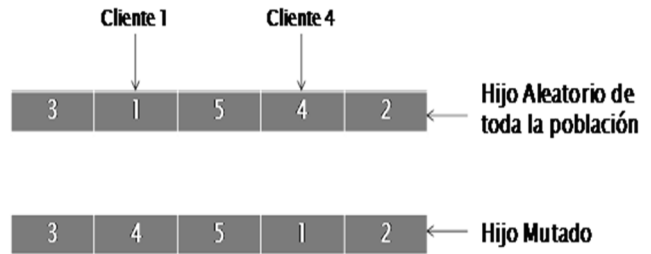


Figura 5. Proceso de mutación.

#### 5. Modificación de la población

Para éste código, sólo se cambia un individuo de la población en cada ciclo generacional. Además, no se permiten infactibilidades en las soluciones. Finalmente, el descendiente debe sustituir al elemento de la población de peor calidad siempre y cuando éste sea de mejor calidad que el peor de la población actual y completamente diferente.

### C. RESULTADOS

La codificación y ejecución del algoritmo propuesto se realizó en el lenguaje de programación Matlab. Se cuantificó para 4 tipos de instancias de diferentes autores [3],[6],[26],[27] se calcularon elementos de comparación como el GAP promedio y el GAP mínimo -donde ambos son la diferencia porcentual con el promedio de las 10 corridas calculadas a través de la metodología propuesta- y el otro respecto al mejor valor encontrado en las 10 corridas en cuanto a la evaluación de la función objetivo de las soluciones finales.

Las codificaciones de las instancias se presentan de la siguiente manera;  $X-NYY-KZ$ . Donde,  $X$  es igual a una inicial que indica el autor de la instancia.  $YY$  es el número de nodos del sistema; es decir, el número de clientes + 1. Finalmente,  $Z$  es el número de vehículos a utilizar. Salvo para las instancias de Christofides, Mingozi y Toth (CMT) [26] cuya codificación es independiente a las características del problema.

El valor BKS.FO indica el valor de la función objetivo de la mejor solución encontrada hasta el momento. En tanto que las instancias que presenten un \* en la casilla OPT (\*), esto indica que ese BKS es el óptimo global de la instancia encontrado a través de alguna técnica exacta de solución.

Según las tablas 2-5, se reporta la ventaja que tiene la metodología propuesta para dar solución aproximada a las instancias seleccionadas de Christofides y Eilon. Estas instancias tienen GAPs para las instancias pequeñas entre 4% - 10%, y para las grandes entre 5% - 12% ambas en el mejor de los casos. Además, se obtuvieron buenos resultados al trabajar con las instancias de CMT, vrnc 1 y vrnc 4, ambas con un GAP menor al 10%.

Por otra parte, para el *benchmark* realizado a los casos de prueba propuestos por Augerat et al., se observa la necesidad de realizar movimientos inter-rutas e intra-rutas para poder escapar de óptimos locales y poder explorar aún más el espacio de soluciones. Para instancias de tamaños similares, se tienen GAPs muy distantes, como es el caso de las instancias A-N48-K7 (4% GAP) y B-N44-K7 (18% GAP).

Caso similar ocurre con las tres instancias propuestas por Fisher, donde para los tres casos, la diferencia promedio con las BKS está por encima del 20%.

Nombre	OPT (*)	BKS. FO	# Iteraciones	Tipo Medida	Promedio	Min	GAP. Min	GAP. Promedio
A-N48-K7	*	1073	1500	F.O	1119.07	1119.07	4%	4%
A-N48-K7	*	1073	1500	Tiempo	42.42	42.21		
A-N46-K7	*	914	1500	F.O	999.01	999.74	9%	9%
A-N46-K7	*	914	1500	Tiempo	48.44	48.23		
B-N44-K7	*	909	1500	F.O	1074.93	1074.30	18%	18%
B-N44-K7	*	909	1500	Tiempo	34.12	33.96		
A-N62-K8	*	1288	2000	F.O	1536.83	1536.70	19%	19%
A-N62-K8	*	1288	2000	Tiempo	69.34	68.97		
B-N57-K9	*	1598	2000	F.O	1764.20	1764.20	10%	10%
B-N57-K9	*	1598	2000	Tiempo	71.60	64.20		
A-N80-K10	*	1763	3000	F.O	2106.54	2105.80	19%	19%
A-N80-K10	*	1763	3000	Tiempo	165.56	161.24		
B-N78-K10	*	1221	3000	F.O	1331.52	1327.50	9%	9%
B-N78-K10	*	1221	3000	Tiempo	161.96	161.12		

Tabla 2. Casos de prueba Augerat et al.[27]

Nombre	OPT (*)	BKS. FO	# Iteraciones	Tipo Medida	Promedio	Min	GAP. Min	GAP. Promedio
vrpc1	*	524.61	2000	F.O	571.64	569.08	8%	9%
vrpc1	*	524.61	2000	Tiempo	55.41	55.11		
vrpc4		1028.4	3000	F.O	1136.28	1120.27	9%	10%
vrpc4		1028.4	3000	Tiempo	349.52	346.48		
vrpc11		1042.1	3000	F.O	1295.57	1285.12	23%	24%
vrpc11		1042.1	3000	Tiempo	180.67	179.78		
vrpc12		819.56	3000	F.O	1014.23	1011.52	23%	24%
vrpc12		819.56	3000	Tiempo	174.13	173.61		

Tabla 3. Resultados casos de prueba de Fisher [6]

Tabla 4. Resultados casos de prueba CMT [26]

Nombre	OPT (*)	BKS. FO	# Iteraciones	Tipo Medida	Promedio	Min	GAP. Min	GAP. Promedio
F-N45-K4	*	724	3000	F.O	900.65	889.94	23%	24%
F-N45-K4	*	724	3000	Tiempo	55.58	54.01		
F-N72-K4	*	237	5000	F.O	325.51	315.35	33%	37%
F-N72-K4	*	237	5000	Tiempo	120.77	119.75		
F-N135-K7	*	1162	5000	F.O	1489.44	1454.94	25%	28%
F-N135-K7	*	1162	5000	Tiempo	3530.37	350.25		

Tabla 5. Resultados de Christofides y Eilon[3]

Nombre	OPT (*)	BKS. FO	# Iteraciones	Tipo Medida	Promedio	Min	GAP. Min	GAP. Promedio
E-N101-K14	*	1071	3000	F.O	1204.34	1203.01	12.00%	12.00%
E-N101-K14	*	1071	3000	Tiempo	311.46	303.45		
E-N101-K8	*	815	3000	F.O	888.56	880.56	8.00%	9.00%
E-N101-K8	*	815	3000	Tiempo	205.43	203.06		
E-N76-K10	*	830	2000	F.O	871.06	870.15	5.00%	5.00%
E-N76-K10	*	830	2000	Tiempo	112.65	109.40		
E-N51-K5	*	521	2000	F.O	571.41	566.36	9.00%	10.00%
E-N51-K5	*	521	2000	Tiempo	56.78	55.17		
E-N33-K4	*	835	2000	F.O	875.59	872.16	4.00%	5.00%
E-N33-K4	*	835	2000	Tiempo	28.14	25.43		
E-N23-K3	*	569	3000	F.O	647.55	623.70	10.00%	14.00%
E-N23-K3	*	569	3000	Tiempo	18.80	15.75		

### III. CONCLUSIONES

- Se desarrolló una metodología de solución de dos fases para el CVRP. Basada en la implementación de la heurística constructiva de barrido para la generación de los clusters, y el AGCB para dar solución a la ruta óptima de cada camión dentro del mismo.
- El AGCB potencializado desarrollado en el trabajo actual presenta resultados satisfactorios en tiempos de cómputo razonables para algunas instancias trabajadas.
- Se evidenció que no solamente el tamaño de las instancias y la cantidad de vehículos, son las que determinan la

complejidad del problema, sino la topología misma de la red.

- Las utilizaciones de heurísticas constructivas para generar soluciones básicas factibles de inicio se muestran como una ventaja tanto computacional como en resultados, ya que se garantiza tener individuos de buena calidad desde el inicio del algoritmo.

#### RECOMENDACIONES

- Incluir estrategias de búsqueda local y de vecindario, para realizar movimientos intra-ruta e inter-ruta. Debido a que la propuesta actual no permite intercambio de información entre los clusters, este aspecto origina respuestas atrapadas en óptimos locales.
- Implementar la generación de clusters aleatorios para explorar aún mejor el espacio de soluciones del problema.
- Desarrollar la metodología de solución en el lenguaje de programación C++. Esto con el fin de lograr mayor eficiencia en los tiempos de ejecución.

#### REFERENCIAS

- [1]. C. Prins. "A simple and effective evolutionary algorithm for the vehicle routing problem". *Computers & Operations Research*, volume 31, issue 12, pages 1985-2002. 2004.
- [2]. M. Gendreau, T. Vidal, T. G. Crainic, C. Prins. "Heuristics for multi-attribute vehicle routing problems: A survey and synthesis". *CIRRELT*. 2012.
- [3]. N. Christofides, S. Eilon. "An Algorithm for the Vehicle Dispatching Problem". *Operational Research Quarterly*. Volume 20, pages 309-318. 1969.
- [4]. J. Klaus. "Bounds for the general capacitated routing problem". *Networks*, volume 23, issue 3, pages 165-173. 1993.
- [5]. G. Cornuejols, F. Harche. "Polyhedral study of the capacitated vehicle routing problem". *Mathematical Programming*, volume 60, issue 1-3, pages 21-52. 1993.
- [6]. M. Fisher. "Optimal Solution of Vehicle Routing Problems Using Minimum k-Trees". *Operations Research*. Volume 42, no. 4, pages 626-642. 1994.
- [7]. Augerat, P., Belenguer, J.M., Benavent, E., Corberan, A., Naddef, D., Rinaldi, G., 1998. *Computational Results with a Branch and Cut Code for the Capacitated Vehicle Routing Problem. Research Report 949-M*, Université Joseph Fourier, Grenoble, France
- [8]. T. K. Ralphs, L. Kopman, W. R. Pulleyblank, Jr. L. E. Trotter. "On the Capacitated Vehicle Routing Problem". *Mathematical Programming Series*. B 94, 343. 2001.
- [9]. J. C. Angel, D. Soler, A. Hervas. "The capacitated general routing problem on mixed graphs". *Revista investigación operacional*, volume 23, No. 1. 2002.
- [10]. M. Gendreau, G. Laporte, J-Y. Potvin. "Metaheuristics for the capacitated VRP". *The vehicle routing problem*, volume 9 of SIAM monographs on discrete mathematics and applications, chapter 6. 2002.
- [11]. J. Berger, M. Barkaoui. "A hybrid genetic algorithm for the capacitated vehicle routing problem". *Genetic and evolutionary computation – GECCO*, volume 1, pages 646-656. 2003.
- [12]. A-L. Chen, G-K. Yang, Z-M. Wu. "Hybrid discrete particle swarm optimization algorithm for capacitated vehicle routing problem". *Journal of Zhejiang University SCIENCE A*, volume 7, issue 4, pages 607-614. 2006.
- [13]. P. Toth, A. Tramontani. "An integer linear programming local search for capacitated vehicle routing problems". *The vehicle routing problem: Latest advances and new challenges*, volume 2, pages 275-295. 2008.
- [14]. J. M. Daza, J. R. Montoya, F. Narducci. "Resolución del problema de enrutamiento de vehículos con limitaciones de capacidad utilizando un procedimiento metaheurístico de dos fases". *Revista EIA*, No. 12, páginas 23-38. 2009.
- [15]. S. R. Venkatesan, D. Logendran, D. Chandramohan. "Optimization of capacitated vehicle routing problem using PSO". *International Journal of Engineering Science and Technology*, volume 3, number 10, pages 7469-7477. 2011.
- [16]. P. Toth, D. Vigo. "An Overview of Vehicle Routing Problems, Monographs on Discrete



Mathematics and Applications”. In: *The Vehicle Routing Problem*. SIAM, 2000, pp. 1-26. 2000.

- [17]. R. Gallego, A. Escobar, E. Toro. “Técnicas metaheurísticas de optimización”. *Universidad Tecnológica de Pereira. Textos Universitarios*, ISBN: 978-958-722-007-0, Ed. 2. 2008.
- [18]. J. Corona. “Hiperheurísticas a través de programación genética para la resolución de problemas de ruteo de vehículos”. *Tesis MSc. Ciencias en sistemas inteligentes*, Instituto Tecnológico y de Estudios Superiores de Monterrey, Monterrey, México. 2005.
- [19]. B. E. Gillet, L. R. Miller. “A Heuristic Algorithm for the Vehicle Dispatch Problem”. *Operations Research*. Volume 22, pages 340-349. 1974.
- [20]. A. Martel. “Le pilotage des flux”. *CENTOR, Université Laval*. DF-3.4.1. 2003.
- [21]. M. Gendreau, A. Langevin, J. M. Frayret. “Notes du cours” « Réseaux Logistiques », séance 6 : Planification du transport par camions. *École Polytechnique de Montréal*. 2013.
- [22]. A. Duarte. “Metaheurísticas”. *Librería-Editorial Dykinson*. Volumen 22 de Ciencias Experimentales y Tecnología. 2007.
- [23]. P. C. Chu, J. E. Beasley. “A Genetic Algorithm for the Generalized Assignment Problem”. *Computers & Operations Research*. Volume 24, issue 1, pages 17-23. 1997.
- [24]. M. Resende. “BiasedRandom-Key Genetic Algorithms With Applications In Telecommunications”. *AT&T Labs Research Technical Report*. Pages 1-6. 2011.
- [25]. W. M. Spears, K. A. De Jong. “On The Virtues of Parameterized Uniform Crossover”. *Proceedings of the Fourth International Conference on Genetic Algorithms*, eds. R. Belew and L. Booker, San Mateo, CA: Morgan Kaufmann, 230-236. 1991.
- [26]. N. Christofides, A. Mingozzi, P. Toth. “The Vehicle Routing Problem”. In: *Combinatorial Optimization* (N. Christofides, A. Mingozzi, P. Toth and C. Sandi, editors). Chichester: John Wiley & Sons Ltd. Pages 315-338. 1979.