

# Inteligencia Artificial

## Estado del arte y aproximación FC+CBJ para el Green Vehicle Routing Problem

Ignacio Cisternas Núñez

17 de marzo de 2020

### Evaluación

Resumen (5 %):	_____
Introducción (5 %):	_____
Definición del Problema (10 %):	_____
Estado del Arte (35 %):	_____
Modelo Matemático (20 %):	_____
Conclusiones (20 %):	_____
Bibliografía (5 %):	_____
<b>Nota Final (100 %):</b>	_____

### Resumen

The Vehicle Routing Problem tiene como objetivo minimizar la distancia total que diversos vehículos deberían recorrer para cubrir una cierta cantidad  $n$  de nodos definidos y luego retornar al punto de inicio. Green Vehicle Routing Problem (GRVP) es una variante a este problema que considera una cantidad máxima de combustible por vehículo y puntos de abastecimiento para recarga. El objetivo es dilucidar tanto la cantidad de vehículos a utilizar, como la ruta seleccionada y los nodos que deben cubrir, mediante la ayuda de un grafo como representación de la situación, todo esto con el fin de minimizar la distancia total recorrida por los vehículos. Este documento busca exponer los avances y distintas aproximaciones a la resolución de GRVP.

## 1. Introducción

El presente informe tiene como objetivo abordar el Green Vehicle Routing Problem (Desde ahora GVRP), una variante del Vehicle Routing Problem (Desde ahora VRP). VRP modela un problema de logística de reparto, en el que vehículos salen de un deposito, pasan por ciertos nodos (clientes y/o de abastecimiento) y vuelven a su origen minimizando la distancia total recorrida. El método GVRP aborda la situación planteada como un grafo que busca las rutas más cortas considerando límites de combustible y posibilidad de recarga en estaciones.

La principal motivación de analizar este problema es la distribución comercial que el mercado en general debe emplear para la distribución de sus productos, donde actualmente es predominante el uso de combustibles fósiles dañinos para el ambiente, además del costo per se que implica movilizar estos vehículos en tiempo y dinero. Este documento busca dar un contexto de los avances realizados en este método, contextualizando propuestas realizadas para su solución y métodos alternativos con base similar que sirvan de punto de comparación.

El presente documento cuenta con 4 partes, además de lo ya introducido. Se define primero el problema, partiendo desde planteamientos básicos como Traveling Salesman Problem (presentado a continuación) y VRP para exponer GRVP en sí. Luego, se expone el estado del arte actual, mencionando las aproximaciones existentes que ofrecen una solución a este problema. Se procede con un modelo matemático que representa la situación planteada y finalmente, se concluye de acuerdo a todo lo analizado.

## 2. Definición del Problema

Para comprender GVRP (Green Vehicle Routing Problem) es necesario entender dos problemas que son base. TSP (Travelling Salesman Problem) y VRP (Vehicle Routing Problem).

### 2.1. TSP : Traveling Salesman Problem

Conocido también como el problema del vendedor viajero. Modela la ruta por la que debería circular un supuesto viajero que debe cruzar óptimamente una cantidad determinada de ciudades para finalmente volver al punto de retorno. Debe considerarse, además, que cada ciudad solo debe cruzarse una vez.

TSP es un problema NP-Hard, es decir, puede verse como un problema NP (problemas que pueden ser resueltos en tiempo polinómico por una máquina de Turing no determinista) Transformado polinomialmente. Además, se tiene conocimiento de que un algoritmo trivial que compruebe las posibles soluciones tiene un costo en tiempo de  $O(n!)$ , siendo  $n$  la cantidad de “ciudades”(nodos) a visitar. Esto implica, por lo tanto, que tanto VRP como GVRP también son NP-Hard de un costo al menos  $O(n!)$ . [J., 2003].

### 2.2. VRP : Vehicle Routing Problem

VRP (Vehicle Routing Problem) tiene como objetivo simular un problema de reparto desde un nodo origen (puede pensarse como un depósito o bodega) hacia varios nodos receptores (pueden pensarse como clientes) e identificar la cantidad de vehículos y rutas óptimas que deberían seleccionarse para abarcar todos los nodos minimizando los costos involucrados.

En pocas palabras, VRP consiste en modelar mediante grafos la logística de despacho a varios clientes por una ruta óptima. Al igual que TSP debe comenzar y terminar el recorrido en el mismo punto, sin embargo, se consideran distintos “vendedores”(vehículos en este caso) disponibles para cumplir la demanda. Una manera analógica de entender este método es con una variante del ya explicado TSP, conocido como Multiple - Travelling Salesman Problem que el lector puede profundizar en [El-Sherbeny., 2010].

Variantes que pueden ser asociadas a VRP:

- (CVRP) Capacitated and Distance-Constrained Vehicle Routing Problem: Incluye una

restricción de capacidad para llevar productos en los vehículos que, al igual que la capacidad de combustible, no deben superar.

- (VRPTW) VRP with Time window : Cada vehículo posee un rango de horario (por nodo) en el cual debe haber un vehículo con carga.
- (VRPB) VRP with Backhauls: Variación en la que cada cliente cuenta con 2 conjuntos asociados, uno con los clientes “anteriores.”<sup>a</sup> el según logística de entrega y otro con los “posteriores.”<sup>a</sup> la ruta designada como óptima.
- (VRPPD) VRP with Pickup and Delivery: Modelo que considera, además de lo ya planteado por VRP, la posibilidad de cada cliente de enviar productos a la distribuidora

Todas estas pueden ser consultadas en [P. and Vigo, 2002].

### 2.3. GVRP : Green Vehicle Routing Problem

A diferencia del ya mencionado VRP, GVRP modela situaciones en la que los vehículos involucrados utilizan alternativas a los típicos combustibles fósiles, como motores eléctricos por ejemplo, los cuales involucran pocos puntos de recarga (en comparación con gasolina tradicional).

El objetivo, al igual que cualquier variante de VRP, es minimizar la distancia total recorrida (y por tanto, tiempo en ruta y uso de combustible) de los vehículos designados. Hay que destacar que ya no se requiere visitar todos los nodos implicados, solo aquellos que correspondan a clientes, los nodos de recarga pueden o no ser una buena elección. Las variables involucradas en este problema son principalmente:

- rutas escogidas por vehículo.
- cantidad de vehículos a enviar.

Las cuales deben restringirse a las siguientes condiciones:

- No se debe asignar una mayor cantidad de vehículos que un máximo definido.
- Cada vehículo debe comenzar y terminar en el mismo lugar.
- Cada nodo cliente debe ser visitado.
- Los vehículos no pueden exceder su capacidad de combustible, pero pueden recargar en puntos de abastecimiento.
- Deben visitarse los nodos en un tiempo menor al asignado como jornada de trabajo..

Un mejor contexto de este modelo puede encontrarse en [Erdogan and Miller-Hooks, 2010].

Algunas variantes a GVRP son:

- (GVRP-Q) Green Vehicle Routing Problem with queue: Variante de GVRP que considera colas en las estaciones de recarga, las cuales son planteadas como M/M/1. Los costos asociados a la espera de los vehículos se pueden profundizar en [Poonthalir, 2019]
- (GVRSP) Green Vehicle Routing Problem with Time-Varying Traffic Congestion: Variante de GVRP en la que se incluye el tiempo promedio que hay entre cada tramo y se penaliza cada vez que un vehículo tarda más o menos de lo pensado, reajustando los tiempos. El modelo lineal que representa este tipo de problemas se observan en [Xiao and Konak, 2015]

- (GVRPTW-P) Green vehicle routing problem with path flexibility: GVRP que considera diversos posibles caminos para llegar de un nodo a otro. Este analiza luces de semáforos o posibles situaciones en el camino que impliquen mayor tiempo en la llegada. Se plantea en [X. Liu and Cheng, 2017]
- (GVRP-MTPR) Green Vehicle Routing Problem with Multiple Technologies and Partial Recharges: Además de considerar lo que GVRP implica, se incluye la posibilidad de recargas parciales de combustible, pensando que la carga en el depósito es favorable al tiempo y en costos, y la posibilidad de considerar variantes (en tiempo y costos) en las estaciones de recarga. explicado en [Ángel and Tirado, 2014]

### 3. Estado del Arte

Dos importantes matemáticos, Sir William Rowan Hamilton (Escosés) y Thomas Penyngton Kirkman (Británico), plantearon en los años 1800 los primeros modelos del problema del vendedor viajero (TSP), el cual es la base principal de VRP (y GVRP). En 1857, Hamilton crea “The Icosian game” (El juego Icosiano) Un tablero de clavijas con 20 agujeros que deben ser cruzados solo una vez y volver al origen. El fin de este juego es dar a conocer lo que se conoce como camino Hamiltoniano, esencia del problema estudiado en este documento [A., 2012].

Las primeras aproximaciones a problemas de reparto se abordan en [Clarke G., 1964], donde se plantea el algoritmo de guardado de Clarke and wright, sin embargo, las primeras investigaciones sobre VPR como tal se exponen específicamente en [Laporte G., 1985], donde se detallan modelos con capacidad de combustible y restricciones de distancia, pero que no consideran acciones en ruta que puedan incrementar estas capacidades.

Respecto a GVRP, este fue planteando por primera vez en [Erdogan and Miller-Hooks, 2010].

A continuación, con el fin de contextualizar el avance en este problema, se presentan posibles soluciones a GVRP propuestas en documentos referenciados:

#### 3.1. Integer Linear programming algorithm of Laporte / Nobert / Desrochers

En el documento propuesto por estos 3 autores [Laporte G., 1985], como su nombre lo indica, se presenta una posible solución del problema mediante programación lineal de enteros, utilizando relajo de restricciones y eliminación de subcaminos para agilizar el cálculo. Esta es una de las primeras soluciones planteadas para el problema de reparto. Se detalla la construcción del algoritmo en pasos con ciertas condiciones que deben cumplirse.

- Se modela el problema mediante un grafo, donde cada nodo representa un punto que cruzar y cada camino tiene un costo asociado.
- Se asumen vehículos de igual capacidad.
- Cada ciudad debe ser “marcada” (cruzada) solo una vez.
- Todo vehículo comienza en el mismo depósito.
- La suma de caminos no puede exceder el camino posible a recorrer con la capacidad máxima de combustible.

El hecho de que no se permita recarga de combustible en algunos nodos, o el que se considere todos los vehículos con la misma capacidad máxima de combustible hace de esta propuesta una aproximación un tanto obsoleta (considerando también que fue planteada en 1984). Además, más que resolver GVRP, este documento apunta a abordar problemas de reparto comunes (VRP).

A continuación, se presentan varios enfoques considerados en [Park, 2014].

### 3.2. Branch and Bound / Dynamic programming

Branch and Bound representa todas las soluciones parciales mediante un árbol, donde cada ramificación va representando posibles soluciones, éstas se van evaluando paso a paso y a medida que los datos no son las deseadas, se van descartando todas las opciones bajo el punto de quiebre, lo que ahorra una importante cantidad de cálculos. Esta opción nos entrega la solución exacta, sin embargo, es eficiente solo en la medida que haya pocos nodos involucrados, después de cierto número implica un tiempo computacional para nada práctico.

Se puede observar en [Andelmin and Bartolini, 2017] que se obtienen resultados razonables utilizando un algoritmo Branch and cut (un tipo de algoritmo branch and bound explicado en [Bard, 1998]) planteado para 100 clientes y 21 estaciones de recarga (tiempo de ejecución máximo de 5.208 h), pero cuando la cantidad de clientes incrementa a 111, aunque se consideran 28 estaciones, se obtiene un incremento severo de tiempo evidentemente poco efectivo (tiempo de ejecución máximo de 11.265). Esto nos da un indicio del exponencial incremento que se presentaría al trabajar con magnitudes de orden mayor.

Existen softwares como CPLEX, LINGO y Xpress que se basan en esta técnica y ofrecen soluciones exactas al entregarles representaciones matemáticas de las situaciones (se presenta más adelante un ejemplo).

Dynamic programming, por su parte, divide la problemática en dos. Primero, busca las rutas óptimas entre cada nodo. Segundo, con los datos entre cada nodo obtenidos, se buscan las rutas óptimas de los vehículos. Al igual que antes, es eficiente en la medida que la cantidad de nodos no supera cierto valor, luego de esto se hace insostenible. En [Qian, 2013] se presenta un ejemplo de programación dinámica. Aunque trabaja sobre un caso de VRP con variación de velocidades, es una representación de como funciona este tipo de aproximaciones. En esta se evidencia que la reducción en el uso de combustibles alcanza hasta un 33 por ciento en casos extremos respecto a la programación habitual.

Para profundizar en Branch and bound, se invita al lector a observar el procedimiento expuesto en [Bektaş et al., 2011]. Un ejemplo de Dynamic programming puede consultarse en [Qian et al., 2014].

Pareciera ser que para un problema de reparto con muchas variables involucradas requiere inevitablemente de una capacidad de cómputo relativamente alta. Sin embargo, podemos abordar GVRP a través de metaheurísticas. La gran mayoría de la investigación correspondiente a GVRP utiliza metaheurística para generar soluciones. pero ¿Qué son?. En palabras simples, es un enfoque que acepta soluciones de peor calidad en cada posible solución candidata (en cada iteración), con el fin de escapar de óptimos locales para una mayor exploración del espacio de búsqueda.

### 3.3. Tabu Search

Metodología de dos pasos, primero encontrando las posibles rutas, luego calculando en reversa (desde el nodo más lejano hacia el depósito de origen) el combustible requerido para completar las rutas óptimas. Una característica es que posee una cola que registra las últimas rutas consideradas con el fin de evitar sugerir ciclos. Existen algunas variantes de este algoritmo como tabu-insertion, tabu-swap y tabu-hybrid. [Park, 2014] para más información.

Una Esencial característica de Tabu search es el admitir soluciones de peor calidad que la mejor ya encontrada, esto con el fin de escapar de óptimo locales y así diversificar la exploración en el campo de soluciones. La ya mencionada lista Tabu es fundamental para no caer en ciclos y dependiendo del largo asignado para esta se obtienen comportamientos distintos. Fomentando la exploración con una lista mayor y motivando la intensificación de en caso de asignar una lista más corta.

En [Gendreau M and G, 1994] se plantea un algoritmo denominado TABUROUTE y se compara este con los algoritmos utilizados en esa época (1994) (generalized savings algorithm, MBSA algorithm, two-phase algorithm, especificados en el documento citado), superando el rendimiento de cada uno.

### 3.4. Simulated annealing

Modelo que utiliza 3 pasos para desarrollarse; primero buscando las posibles rutas, segundo obteniendo los óptimos tiempos en cada ruta, y finalmente el gasto en combustible. Principalmente, este método utiliza una analogía de temperatura para representar la intensificación y diversificación del espacio de búsqueda, mostrándose más "frío." más caliente respectivamente (con colores azul y rojo por ejm). Para detalles, ver también [Park, 2014].

La principal característica de este algoritmo es la mayor disposición a explorar el dominio de soluciones en un inicio al tener un aceptación de peores soluciones proporcional a la "temperatura" del sistema. A medida que va iterando, es más propenso a buscar óptimos locales (se va enfriando).

Diferentes evaluaciones de algoritmos basados en Simulated Annealing pueden encontrarse en [Xiao, 2012].

### 3.5. Genetic Algorithm

Los Genetic algorithm (por su traducción: Algoritmos genéticos) se basan en la evolución darwiniana como tal, es decir, tienen como objetivo cruzar.º mejor dicho, combinar, dos diferentes soluciones con el fin de obtener lo mejor de ambas, algo que no necesariamente debe ocurrir, pero que si ha presentado frecuentemente mejoras en resultado respecto a las soluciones originales.

Un ejemplo del uso de estos algoritmos se puede apreciar en [Thangiah, 1999]. Donde, si bien se analiza el Vehicle Routing Problems with Time Windows (VRPTW) y no GVRP, es posible comparar el rendimiento de este método respecto a los ya mencionados Tabu Search y Simulated Annealing. Utilizando una heurística híbrida que utiliza algoritmos genéticos, se encontraron, en 60 casos distintos, 9 soluciones de igual rendimiento que la mejor alternativa encontrada por los otros 2 métodos y 30 soluciones que la superan, además de considerar siempre una menor cantidad de vehículos para realizar los despachos. Cabe destacar que el uso de algoritmos genéticos puede combinarse con Tabu Search / Simulated Annealing para mejorar soluciones locales y conseguir mayor eficiencia.

La población de soluciones tiene diferentes representaciones y dependiendo de estas se crean algoritmos genéticos que permitan el cruzamiento y las mutaciones, en [Thangiah, 1999], por ejemplo, se observa una representación en bits. Lo que permite, como se muestra en el documento, el cruzamiento de las soluciones en un punto y la mutación de cada gen con una probabilidad asociada que determina si el valor que posee debe cambiar en cada iteración.

Como ya se ha mencionado, el uso de metaheurísticas es la actual tendencia para resolver problemas de esta naturaleza, sin embargo, aquellas aproximaciones que involucran algoritmos genéticos prometen ser las mejores aproximaciones a considerar, por su naturaleza simple, pero efectiva, dando origen a soluciones que difícilmente podríamos obtener de manera directa. Más detalles del funcionamiento de este método pueden encontrarse en [Thangiah, 1999].

## 4. Modelo Matemático

En [Erdogan and Miller-Hooks, 2010]. Se muestra una representación matemática de GVRP explyada a continuación.

### 4.1. Parámetros

- $I_0$ : Conjunto Vértice Clientes.
- $F_0$ : Conjunto de Vértices Estaciones de recarga.
- $P_i$ : Tiempo en Vértice  $i$  (Si  $i$  es un vértice cliente, refiere a tiempo empleado en servicio de entrega, si  $i$  es un vértice estación de recarga, refiere a tiempo empleado en recarga de vehículo que tiende a ser constante.)
- $r$ : Tasa de consumo de combustible de los vehículos.
- $Q$ : Capacidad máxima de combustible de vehículos.

### 4.2. Variables

$$X_{ij} = \begin{cases} \text{Si el vehículo recorre el camino desde } i \text{ hacia } j & 1 \\ \text{si no} & 0 \end{cases} \quad (1)$$

- $Y_j$ : Combustible restante al llegar a vértice  $j$  (Toma valor  $Q$  en estaciones de recarga y al volver al depósito.)
- $\tau_j$ : Tiempo transcurrido hasta vértice  $j$  desde que vehículo abandona el depósito.
- $m$ : Número de vehículos operando.

### 4.3. Función objetivo

$$\text{Min} \sum_{\substack{i,j \in V' \\ j \neq i}} d_{ij} x_{ij} \quad (2)$$

#### 4.4. Restricciones

$$\sum_{\substack{j \in V' \\ j \neq i}} x_{ij} = 1, \quad \forall i \in I, \quad (3)$$

$$\sum_{\substack{j \in V' \\ j \neq i}} x_{ij} \leq 1, \quad \forall i \in F_0 \quad (4)$$

$$\sum_{\substack{i \in V' \\ j \neq i}} x_{ij} - \sum_{\substack{i \in V' \\ j \neq i}} x_{ij} = 0, \quad \forall j \in V', \quad (5)$$

$$\sum_{\substack{j \in V' / 0 \\ j \neq i}} x_{0j} \leq m, \quad (6)$$

$$\sum_{\substack{j \in V' / 0 \\ j \neq i}} x_{j0} \leq m, \quad (7)$$

$$\tau_j \geq \tau_i + (t_{ij} - p_j)x_{ij} - T_{max}(1 - x_{ij}), i \in V', \quad \forall j \in V' / \{0\}, \text{ and } i \neq j \quad (8)$$

$$0 \leq \tau_0 \leq T_{max}, \quad (9)$$

$$t_{0j} \leq \tau_j \leq T_{max} - (t_{j0} - p_j), \quad \forall j \in V' / \{0\}, \quad (10)$$

$$y_j \leq y_i - r(d_{ij}x_{ij}) + Q(1 - x_{ij}), \quad \forall j \in I \text{ and } V' / \{0\}, \quad (11)$$

$$y_j \geq \text{Min}(r(d_{j0}, r((d_{jl} + d_{l0})))), \quad \forall j \in I, \forall l \in F', \quad (12)$$

$$x_{ij} \in 0, 1 \quad \forall i, j \quad (13)$$

El objetivo del modelo es minimizar (2) la distancia total recorrida por los vehículos, la función objetivo. La ecuación (3) (para clientes) y (4) (para puntos de recarga) asocia cada vértice visitado a un sucesor, en otras palabras, a cerrar todo posible circuito. (5) Mantiene el flujo al obligar el mismo número de vehículos que entran en cada vértice con el número de vehículos que salen. Se debe asegurar que a lo más "m" vehículos dejen el depósito (6) y a lo más "m" vehículos retornen al depósito (7). Además, estos deben volver antes de  $T_{max}$  que es el tiempo máximo, esto se indica en (8), (9) y (10). Por su lado, (9) mantiene los tiempos de llegada a cada vértice para evitar crear subrutas con vértices ya visitados. (10) fija límites máximos de llegadas al depósito, con el objetivo de cumplir toda la entrega en  $T_{max}$ . (11) reduce los niveles de combustible "q" a medida que el vehículo recorre los vértices, (12) nos asegura que los vehículos no se quedarán sin combustible en medio de la ruta y, finalmente, (13) nos garantiza integridad binaria.



## 5. Representación

De acuerdo a las restricciones, las variables y los parámetros ya mencionados, se explica la representación utilizada.

La ubicación de las instalaciones se representa por coordenadas geométricas  $(x, y)$ , es decir,  $x, y \in R^2$  con valores entre  $(-90, 90)$ . El tiempo y el combustible se determinan como variables positivas de una sola dimensión, por lo que  $t, q \in R^+$ . El estado de cada cliente (visitado o no visitado) es modelado como un arreglo binario.  $X_i \in (0, 1)$  con  $X_i = 0$  si cliente  $i$  no ha sido visitado y  $X_i = 1$  en caso contrario. Las distancias entre nodos se representan en una matriz. Un espacio  $M_{n \times n}(R^+)$  donde  $n$  es la cantidad total de nodos, es decir, la suma de clientes y estaciones de recarga ( $n = I_0 + F_0$ )

Las soluciones en sí son modeladas como un string donde se indica el identificador del vehículo ( $A$ ) con una flecha que indica el número del nodo a visitar( $n$ ). separados por una barra ( $|$ ) que separa cada paso ( $|A- > n|$ ). Tanto el identificador, como los nodos son numeros naturales  $A, n \in N$

## 6. Descripción del algoritmo

La idea central del algoritmo es encontrar con un vehículo una posible ruta hasta que no tenga más opciones de movimiento, para entonces mandar otro vehículo desde el origen y así sucesivamente. Cada movimiento se realiza llamando una funcion recursiva que se instancia a si misma cada vez que hay desplazamiento o se necesite enviar a otro vehículo. Cuando una solucion candidata es encontrada, se retorna información para volver a la instancia anterior (paso anterior) y que esta siga iterando en busqueda de otras soluciones. Bajo estas premisas se desarrolla el software de busqueda completa.

Las técnicas asociadas en este algoritmo son Forward checking (FC) y Conflict-Directed Jumping (CBJ). Mediante el uso de un arreglo binario que nos indica los clientes que han sido ya visitados se evita el chequeo de soluciones que lleguen más de una vez al mismo cliente, además, la matriz que asocia las distancias entre cada nodo almacena un valor invalido (-1) para aquellos puntos que se encuentran demasiado lejor entre sí (Respecto a capacidad máxima de combustible de los vehículos). Ambas estructuras de datos nos facilitan la implementación del ya mencionado FC.

Para el caso de CBJ, se implementa un uso más bien implicito, pues al chequear previamente con FC los candidatos infactibles, no es posible llegar directamente a instanciaciones que generen conflictos. Es por esto que se considera como conflicto la idea de no poder llegar a otros nodos no visitados ya sea por combustible o por tiempo, en cuyos casos la instancia de la función recursiva que representa el último paso decidido termina retornando el arreglo binario de nodos visitados, permitiendo las siguientes iteraciones del paso anterior y así sucesivamente hasta recorrer todas las soluciones candidatas.

La data de posiciones de los nodos indica en primer lugar la posición de las estaciones de recarga. Es importante desarrollar la matriz de distancia de tal forma que los clientes estén dispuestos antes de las estaciones para que sean prioridad en el chequeo, evitando así soluciones no prácticas en las que se asignen vehículos que caigan en ciclos solo entre las estaciones de recarga.

## 7. Experimentos

Para comprobar los resultados del algoritmo, se prueba con variados archivos que siguen el siguiente formato de orden.

Archivo	clientes	estaciones	Max tiempo	Max distancia	velocidad	t.clientes	t.recarga
ID nodo	Tipo nodo		longitud	latitud			
ID nodo	Tipo nodo		longitud	latitud			
.....							
ID nodo	Tipo nodo		longitud	latitud			

El primer punto mencionado en cada archivo es el deposito, desde el que parten los vehículos, seguido por las estaciones de recarga, que como se menciona anteriormente, son almacenadas en la matriz de distancias despues de los nodos clientes, los cuales son representados por las últimas filas de cada archivo

Para efectos prácticos se realizan 10000 iteraciones del problema, ya que la implementación tarda bastante tiempo en arrojar resultados que puedan ser analizados. El equipo utilizado para las evaluaciones posee las siguientes características 4GB de RAM , tarjeta de vídeo Intel HD Graphics 3000 y un procesador Intel Core i5 de 2,1Ghz

## 8. Resultados

Se muestran a continuación los valores obtenidos en la evaluación de 7 casos diferentes:

Archivo	N.autos	Distancia	N.clientes	Tiempo
AB101	8	2937.64	50	4.995
AB102	11	3595.05	50	5.813
AB103	9	3093.55	50	7.655
AB104	5	1634.67	50	4.736
AB105	9	2327.93	75	8.046
AB106	13	4531.56	75	8.032
AB107	16	5043.2	75	8.421

Cuadro 1: Resultados obtenidos con algortimo sobre varios casos distintos modelados en archivos. Se indica nombre de archivo, cantidad de vehiculos enviados, Distancia total recorrida (km) y tiempo (s) de ejecucion de programa

## 9. Conclusiones

Como era de esperarse, una aproximación de busqueda completa implica una cantidad de recursos de computación importantes, considerando que podrían considerarse dimensiones en el espacio de busqueda mucho mayores. El uso de FC + CBJ muestran la gran ventaja de ahorrar una importante cantidad de cálculos que optimizan la busqueda de la mejor solución, sin embargo, no son suficientes para resolver en un tiempo prudente los modelos implicados en la experimentación mostrada previamente.

Existen diversas formas de abordar este tipo de situación. Una postura de Greedy, por ejemplo, puede ser una buena primera aproximación, pero existen mejores formas de optimizar situaciones de mayor envergadura y apuntar a resultados que hagan la diferencia. Algoritmos, como los mencionados Tabu Search o de programación dinámica tienen una aproximación de óptimo local, que como se vió, ofrece un mejor resultado en tiempos de cómputo respecto a las búsquedas exactas y entregan soluciones que pueden considerarse aceptables. Trabajar con este tipo de algoritmos es eficiente en la medida que se pueden manejar los parámetros de manera adecuada. Fomentando la exploración y la diversificación en la medida correcta.

La apuesta más prometedora son los algoritmos genéticos, que cuentan con la ventaja de creación de posibilidades jamás imaginadas por un ser humano. Han probado ser un aproximamiento muy prometedor respecto a resultados como lo destacado en [Thangiah, 1999]. Donde supero en 30 casos al estado del arte de varios problemas tipo VRP. Este último enfoque debería considerarse con suma importancia, ya que con capacidades de cómputo que cada vez son mayores y con investigación matemática que incrementa la eficiencia de algoritmos, es posible que se encuentren soluciones que sorprendan nuestras expectativas.

La industria debería enfocar la resolución de este tipo de problemas en la creación de nuevas heurísticas que aborden el problema, o en su defecto, una combinación de estas a lo largo de la ejecución. Como se plantea en este documento, la aproximación de métodos de búsqueda completos requiere de una capacidad de cómputo y tiempos importantes a considerar. Esto, teniendo en cuenta que el dominio del problema no era significativamente grande, (Como podría ser un problema a escala multinacional o global) incentiva el uso de métodos que, aunque no exploren la totalidad de las posibilidades, puedan ofrecer soluciones que mejoren significativamente el rendimiento.

## 10. Bibliografía

### Referencias

- [A., 2012] A., R. S. (2012). *The Traveling Salesman problem*. Indiana State University,.
- [Andelmin and Bartolini, 2017] Andelmin, J. and Bartolini, E. (2017). *An Exact Algorithm for the Green Vehicle Routing Problem*.
- [Bard, 1998] Bard, J. F., H. L. D. M. . J. P. (1998). *A branch and cut algorithm for the VRP with satellite facilities*.
- [Bektaş T, 2011] Bektaş T, L. G. (2011). *The pollution-routing problem*. Transport Res B-Meth. 2011:44.
- [Clarke G., 1964] Clarke G., Wright, J. (1964). *Scheduling of vehicle from central depot to a number of delivery points*. Elsevier.
- [El-Sherbeny., 2010] El-Sherbeny., N. A. (2010). *Vehicle routing with time windows: An overview of exact, heuristic and metaheuristic methods*. Journal of King Saud University - Science, 22(3),.
- [Erdogan and Miller-Hooks, 2010] Erdogan, S. and Miller-Hooks, E. (2010). *A Green Vehicle Routing Problem*. Elsevier.
- [Gendreau M and G, 1994] Gendreau M, H. A. and G, L. (1994). *A tabu search heuristic for the vehicle routing problem*. Mngt Sci 40.

- [J., 2003] J., W. G. (2003). *Exact Algorithms for NP-Hard Problems: A Survey. Lecture Notes in Computer Science*. Elsevier.
- [Laporte G., 1985] Laporte G., Nobert, Y. D. M. (1985). *Optimal routing under capacity and distance restrictions*. INFORMS.
- [P. and Vigo, 2002] P., T. and Vigo, D. (2002). *An overview of vehicle routing problems*. SIAM monographs.
- [Park, 2014] Park, Y., . C. J. (2014). *A review of the solution approaches used in recent G-VRP (Green Vehicle Routing Problem)*. International Journal of Advanced Logistics.
- [Poonthallir, 2019] Poonthallir, G., . N. R. (2019). *Green vehicle routing problem with queues*. Expert Systems With Applications.
- [Qian, 2013] Qian, J., . E. R. (2013). *Finding least fuel emission paths in a network with time-varying speeds*. Department of Management Science, Lancaster University Management School, Lancaster, LA1 4YX, United Kingdom.
- [Qian J, 2014] Qian J, E. R. (2014). *Finding least fuel emission paths in a network with time-varying speeds*. Networks 2014:63.
- [Thangiah, 1999] Thangiah, S. (1999). *A hybrid genetic algorithms, simulated annealing and tabu search heuristic for vehicle routing problems with time windows*. CRC Press.
- [X. Liu and Cheng, 2017] X. Liu, M. Q. and Cheng, C. (2017). *Green vehicle routing problem with path flexibility*.
- [Xiao and Konak, 2015] Xiao, Y. and Konak, A. (2015). *Green Vehicle Routing Problem with Time-Varying Traffic Congestion*.
- [Xiao, 2012] Xiao, Y., Z. Q. K. I. . X. Y. (2012). *Development of a fuel consumption optimization model for the capacitated vehicle routing problem*. Computers Operations Research 39(7).
- [Ángel and Tirado, 2014] Ángel, F., O. T. R. G. and Tirado, G. (2014). *A heuristic approach for the green vehicle routing problem with multiple technologies and partial recharges*. Elsevier.