UNIVERSIDAD DE LA REPÚBLICA

Facultad de Ingeniería

Instituto de Computación

“MDVRP”

Introducción Estado del Arte de Proyecto de Grado

Javier de Prado

Alejandro García

Francisco Güella

Tutores

Sandro Moscatelli

Omar Viera

Contenido

1 Introducción

Este documento trata sobre el estado del arte de MDVRP del proyecto de grado de la carrera Ingeniería en Computación de los estudiantes Francisco Güella, Alejandro García y Javier de Prado.

La gestión logística es un elemento clave en la estrategia empresarial, siendo una de sus funciones principales la distribución, y dentro de ella la capacidad para optimizar las rutas de transporte. En este contexto, las empresas deben analizar los factores más relevantes en el diseño de sus rutas vehiculares así como las metodologías más adecuadas para tal optimización. La optimización de una ruta engloba todas las acciones que contribuyen a la mejora de la función de distribución en términos de nivel de servicio, calidad y costos a través de decisiones de carácter estratégico, táctico y operativo. [1]

El estudio de los problemas de optimización combinatoria se remonta a 1784 cuando G. Monge busca la forma óptima de transportar tierra desde un terreno a otro. En su estudio, busca la forma de transportar tierra de forma tal que la distancia total de transporte sea la menor posible. [2].

El Problema a estudiar en este proyecto de grado, el cual está relacionado a la gestión logística y a la optimización combinatoria, es el de Ruteo de Vehículos con Múltiples Depósitos (MDVRP, Multi Depot Vehicle Routing Problem). El escenario planteado presenta a un conjunto de clientes con una determinada demanda a los cuales hay que distribuirles mercadería. Quienes distribuyen la mercadería cuentan con varios depósitos que tienen capacidad de almacenamiento y una flota de vehículos. La mercadería se traslada a través de la flota de vehículos. El problema planteado es el de optimizar la elección de las rutas que deben realizar los vehículos para satisfacer la demanda de los clientes teniendo en cuenta que los vehículos tienen una capacidad limitada para el transporte de la mercadería. Típicamente se plantea que los vehículos comiencen y terminen su ruta en el mismo depósito y además el cliente recibe una única visita de un vehículo de la flota. El mencionado es la versión básica del problema. Bodin et al en [3] formula el problema de Ruteo de Vehículos con múltiples Depósitos. Otras definiciones del problema MDVRP se pueden encontrar en [4] [5] [6]. Se han propuesto distintas variantes las cuales se comentarán más adelante.

El problema de optimizar las rutas para los vehículos se puede entender como el problema de encontrar un valor mínimo para algún criterio como puede ser distancia, tiempo, consumo de combustible, etc., relacionado a la ruta del vehículo. En general a este criterio se lo presenta como costo. En la definición del problema recién expuesta, se planteó que un vehículo distribuye mercadería a un cliente. Un problema equivalente sería el de recoger mercadería de los clientes y llevarlos a los depósitos. Por ejemplo, cuando un camión levanta la leche de los tambos.

El problema MDVRP es una generalización del problema VRP (Vehicle Routing Problem) [4]. El problema VRP consta de optimizar las rutas en el mismo escenario, con la diferencia que se cuenta con un único depósito. Fue formulado en 1959 por Dantzig y Ramser [7] en donde se presenta "The Truck dispatching Problem" en el cual un camión debe distribuir combustible a un conjunto de estaciones de servicio (clientes). Las estaciones tienen una demanda y los camiones capacidades de carga de combustible. De esta forma se daba comienzo al estudio de lo que luego se conocería como VRP [8].

A su vez, el problema VRP es una generalización del problema TSP. (Travelling Salesman Problem). Así presentaron Dantzig y Ramser "The Truck dispatching Problem" [7], como una generalización de TSP. Travelling Salesman Problem, en castellano "Problema del Agente Viajero" es el siguiente: Dada una lista de ciudades y las distancias entre cada una de ellas, ¿cuál es la ruta más corta posible que visita cada ciudad exactamente una vez y regresa a la ciudad origen? [2] [9]. Tomando a las ciudades como clientes, y tomando a un único vehículo que no lleva ninguna carga y que solamente debe visitar a los clientes, es que se puede ver al problema VRP como una generalización del problema TSP. O sea que TSP sería un caso particular del problema VRP como a su vez VRP sería un caso particular del problema MDVRP.

El problema TSP es un problema NP-Duro, demostrado por Richard Karp en 1972 [10]. También los problemas VRP y MDVRP son NP-duros [11]. Esta es la razón por la cual el objetivo que se plantea generalmente es encontrar una buena solución y no la que minimiza el costo total (solución óptima).

La complejidad NP-Duro de estos problemas, que aumenta exponencialmente a medida que lo hace el número de clientes, dificulta el desarrollo de métodos que resuelvan el problema de manera óptima en un tiempo razonable. No obstante, y a pesar de su elevado costo computacional, existen métodos exactos aplicados a este tipo de problemas que serán mencionados posteriormente. El enfoque más habitual a la hora de resolver estos problemas es el de aplicar métodos heurísticos o meta heurísticos, capaces de generar soluciones cercanas a la óptima sin incurrir en altos tiempos de ejecución y carga computacional.

2 Variantes en los problemas de ruteo de vehículos

La diversidad de aplicaciones donde asuntos de ruteo pueden ser encontrados ha llevado a plantear diferentes variantes de los problemas VRP y MDVRP. Mencionaremos algunas de ellas.

2.1 Ventanas de tiempo

L. Bodin et al, en su estado del arte sobre el ruteo de vehículo [3], presenta la restricción de ventanas de tiempo. La misma restringe en una ventana horaria el tiempo en el cuál un cliente puede ser atendido por un vehículo.

Una variante a esta restricción se puede encontrar en [12]. Donde Lei Wen et al, plantean que la misma no es estricta. Esto quiere decir que se puede dar servicio a un cliente fuera de su ventana de tiempo, pero esto aumentaría el costo de la ruta del vehículo. Distinguen las restricciones de ventana de tiempo entre hard time window y soft time windows. Ellos (Lei Wen et al) utilizan soft time Windows para el abordaje del problema. En hard time Windows el vehículo no podría dar servicio al cliente fuera de la ventana de horario bajo ningún concepto.

2.2 Tipos de flota disponible

L. Bodin et al, en su estado del arte sobre el ruteo de vehículo [3], presenta la siguiente distinción entre los tipos de vehículos que podrían componer la flota.

Flota homogénea: todos los vehículos de la flota son idénticos.

Flota heterogénea: hay varios tipos de vehículos con distintas capacidades.

L. Bodin et al, presenta una situación en el planteo “The Fleet Size and Mix Problem” en la cual la flota heterogénea de los proveedores de servicio se compone por vehículos comprados y vehículos alquilados. En este caso esta situación de flota heterogénea implica en este planteo que los costos de las rutas estén compuestas por un costo fijo (el del alquiler del vehículo) y un costo variable (el costo de transitar la ruta). En esta formulación se asumió que la cantidad de vehículos de cada tipo a disposición es ilimitada.

En [11] Kumar et al, proponen las siguientes dos opciones en la formulación del problema VRP. Dicen que la cantidad de vehículos los cuales dispone la flota, puede ser una cantidad dada a priori, o la misma podría llegar a ser una variable de decisión.

Baldacci et al, presentan en [13] una variante de VRP, en la cuál una flota de vehículos caracterizados por diferentes capacidades y costos está disponible para las actividades de distribución. En este caso clasifica a esta variable de VRP como “Mixed Fleet VRP” o “Heterogeneous Fleet VRP”. Baldacci plantea que hay cierta homogeneidad en llamar a Heterogeneous VRP como la variante que limita el número de vehículos, y Fleet Size and Mix VRP a la variable que no limita la cantidad de vehículos.

Baldacci [13] muestra una clasificación de las variantes VRP según costos fijos y variables de la flota, Fleet Size and Mix y Heterogeneous VRP

2.3 Periodicidad

Tan y Beasley [14] definen a la variante de PVRP (Period VRP) de la siguiente manera:

The Period Vehicle Routing Problema (PVRP) es el problema de designar rutas de vehículos para todos los días de un período dado de T días donde no todos los clientes requieren servicio en todos los días del período. Típicamente si un cliente requiere k visitas ($k <= T$) durante el período entonces estas k visitas podrían ocurrir en una manera determinada dado un número de posibles maneras. Por ejemplo si un cliente requiere servicio dos días en un período de 5 días entonces las posibles combinaciones de días para brindar el servicio podrían ser Lunes/Miércoles, Martes/Jueves, o Miércoles/Viernes donde estas serían las únicas combinaciones de días aceptables.

Francis y Smilowitz en [15] proponen una variante al problema recién descrito. El mismo, PVRP-SC (The Period Vehicle Routing Problem with Service Choice) presenta la siguiente característica. Mientras que en PVRP, las frecuencia de visitas al cliente en un período de tiempo está predeterminada, en PVRP-SC la frecuencia es una variable de decisión. En la definición de PVRP recién presentada, las únicas opciones eran que un cliente sea visitado en una de las siguientes combinaciones: Lunes/Miércoles, Martes/Jueves, o Miércoles/Viernes. En PVRP-SC, si bien se plantea una frecuencia mínima (por ejemplo que el cliente tenga que ser visitado por lo menos un día a la semana), dicha frecuencia podría ser mayor y es una variable de decisión. La frecuencia con que se visita a un cliente en este caso influye en el costo de la ruta. O sea que impactaría en la función objetivo del problema. Los beneficios de una mayor frecuencia podrían representar los ahorros en el cliente de costos de almacenamiento, o la disposición del cliente de pagar por más visitas en la semana.

Tan y Beasley en [14] hacen notar que PVRP es una generalización del problema VRP, el cual tendría un período de un solo día.

Mingozzi y Valleta, en [16] relacionan al PVRP con el MDVRP ya mencionado. Dicen que MDVRP puede ser visto como un caso especial de PVRP donde cada día del período de planeamiento corresponde a un depósito diferente y donde cada cliente requiere ser visitado una única vez en el período tomado en cuenta.

2.4 Otras variantes en los problemas de ruteo de vehículos

Algunas otras variantes que se pueden encontrar en los problemas de ruteo de vehículos son las siguientes:

2.4.1 SDVRP (Split Delivery VRP)

En [17] se plantea la variante SDVRP la cual es una relajación del VRP, en donde se elimina la restricción de que cada cliente sea visitado una sola vez por un vehículo, admitiéndose que el mismo puede ser visitado cualquier cantidad de veces por distintos vehículos. Es decir que la demanda del cliente puede ser partida de forma tal que sea satisfecha por varios vehículos (Split Delivery). Este problema fue planteado e investigado por Dror y Trudeau entre 1989 y 1990, los cuales mostraron que al permitir entregas partidas en la resolución del VRP se puede lograr una disminución de la cantidad de rutas y por lo tanto una reducción del costo total de la solución.

2.4.2 SVRP (Stochastic VRP)

Se trata de un VRP en que uno o varios componentes del problema son aleatorios. Se pueden clasificar en tres diferentes tipos de SVRP:

- Clientes estocásticos: Cada v_i cliente está presente con probabilidad p_i y ausente con probabilidad $1 - p_i$.
- Demandas estocásticos: La demanda d_i de cada cliente v_i es una variable aleatoria.
- Tiempos estocásticos: Los tiempo de servicio y los tiempos de viajes son variables aleatorias.

En SVRP, dos etapas se realizan para obtener una solución. Una primera solución se determina antes de conocer las realizaciones de las variables aleatorias. En una segunda etapa, un recurso o acción correctiva se pueden tomar cuando se conocen los valores de las variables aleatorias.

El objetivo es reducir al mínimo la flota de vehículos y la suma del tiempo de viaje necesaria para abastecer a todos los clientes con valores aleatorios en cada ejecución para los clientes que se sirven, sus demandas y/o los tiempos de servicio y de viaje [18].

2.4.3 VRPPD (VRP Pickup and Delivery)

El VRPPD (VRP Problema con Entregas y Devoluciones) es una variante del VRP donde se contempla la posibilidad de que un cliente que ha recibido un envío disponga además de cierta mercancía que necesita ser recogida. Por lo tanto, se debe tener en cuenta que los productos que los clientes introducen en el vehículo no deben nunca exceder la capacidad del vehículo. Esta restricción dificulta aún más el problema de planificación y puede conllevar una mala utilización de la capacidad de los vehículos, incrementar las distancias reco0rridas o la necesidad de utilizar una flota más amplia. Por consiguiente, es habitual considerar situaciones restrictivas en las que todos los envíos comienzan en el depósito y todas las devoluciones vuelven al depósito central. Así se impide la posibilidad de intercambio de mercancías entre clientes. Una alternativa es la de relajar la restricción de que todos los clientes deben ser visitados al menos una vez. Y otra simplificación habitual es la de considerar que cada vehículo debe entregar todos los productos antes de comenzar con las devoluciones. Así pues, el objetivo es minimizar la flota de vehículos y la suma de los tiempos de transporte con la restricción de que cada vehículo debe tener suficiente capacidad para transportar los productos que van a ser entregados y aquellos que debe recoger en cada cliente para traerlos de vuelta al depósito.

Para que una solución se considere factible, la cantidad total asignada a cada ruta no debe exceder la capacidad del vehículo que sirve esa ruta y dicho vehículo tendrá suficiente capacidad para recoger las devoluciones que cada cliente tenga en su poder [18].

2.4.4 VRPB (VRP with Backhauls)

El problema VRPB (VRP con viajes de regreso) es como un VRP clásico en el que los clientes pueden recibir o entregar productos. Así, se necesita un VRPPD para tener en cuenta las mercancías que los clientes devuelven deben caber en el vehículo que les acaba de hacer la entrega. El supuesto fundamental de que todas las entregas se pueden hacer antes de iniciar las devoluciones puede tenerse en cuenta.

Las cantidades que se deben entregar y recoger son conocidas de antemano. Un VRPB es similar al VRPPD pero con la restricción estricta de que las entregas para cada ruta deben ser completadas antes de realizar ninguna devolución.

El objetivo es encontrar un conjunto de rutas que minimice la distancia total recorrida [18].

2.4.5 Variantes en la Ruta

En algunos planteos de problemas de ruteo se consideran restricciones con respecto a las rutas que deberán transitar los vehículos. Por ejemplo, en la formulación de “The Truck Dispatching Problem” de B. Golden, la cual busca minimizar la distancia recorrida de los camiones, se plantea que el tiempo que le lleva a un camión transitar su ruta no debería superar un máximo dado [8]. La misma podría representar la restricción de que un chofer de camión por las leyes de tránsito no podría manejar más de tantas horas diarias.

O, como se indica en [19] donde se restringe que una ruta no puede superar un costo máximo especificado.

2.4.6 Variantes en la función objetivo (funciones Multi-Objetivo)

Al estudiar los problemas de ruteo de vehículos, se encuentran variantes en cuanto a la función objetivo que se plantea. Hasta ahora se mencionó que el problema es minimizar el costo total de las rutas que realizan los vehículos. Pudiendo representar dicho costo, la distancia o el tiempo que implican recorrer la ruta. Pero también se encuentran algunas variantes al respecto. Por ejemplo, García-Najera, en [20] se propone un problema de ruteo de vehículos en donde se proponen minimizar el costo total de las rutas y a su vez minimizar la cantidad de rutas (o vehículos) simultáneamente. Lo presenta como un problema Bi-objetivo.

Otro ejemplo es el estudio realizado por Robert Bowerman en [21] donde presenta una formulación matemática multi-objetivo del problema de ruteo del bus escolar urbano. En el mismo se contempla, además del criterio de eficiencia, que para Bowerman la solución más eficiente es la que utiliza menos vehículos, el criterio de equidad de las rutas para los estudiantes. En el mismo dice que las rutas deben ser equilibradas. Una forma de proveer equidad en las rutas podría ser haciendo un "balance de carga" de los buses, que implica que la cantidad de estudiantes que lleva cada ruta debería ser equivalente. A su vez plantea un "balance de largo" el cual implica que no deberían existir rutas muy distintas en el largo de su recorrido. También se toma en cuenta la distancia que un alumno camina por día desde su casa a la parada del bus. Bowerman plantea 5 funciones matemáticas en la formulación del problema del bus.

$f1$ =*Distancia total de las rutas de los buses.*

$f2$ =*Distancia que deben caminar los estudiantes.*

$f3$ =*Balanceo de carga.*

$f4$ =*Balanceo del largo.*

Con estas definiciones de funciones, el objetivo es minimizar la función ($f1, f2, f3, f4$) sujeto a un conjunto de restricciones (por ejemplo que un estudiante no puede caminar más de x distancia, o que el largo de una ruta no puede superar y). Teniendo en cuenta que el balance de carga y largo es menor cuanto más parecidas sean las cargas y los largos de las rutas.

2.4.7 Variantes en la capacidad de los depósitos en problemas multi-depósito

Es la variante en los problemas de ruteo con múltiples depósitos en los cuales estos están limitados por una capacidad establecida. A diferencia de los clásicos problemas MDVRP, en los cuales se asume que un depósito tiene capacidad ilimitada y las restricciones de capacidad son únicamente para los vehículos. [22]

3 Formulación Matemática

Generalmente a estos problemas se los modela a través de grafos ponderados.

Sea un grafo $G = (V, E, C)$. El conjunto de nodos $V = \{0, 1, \ldots, N\}$ representa los sitios que participan en el problema, es decir, clientes y depósitos, donde N es la cantidad total de clientes y depósitos. [3] [23]

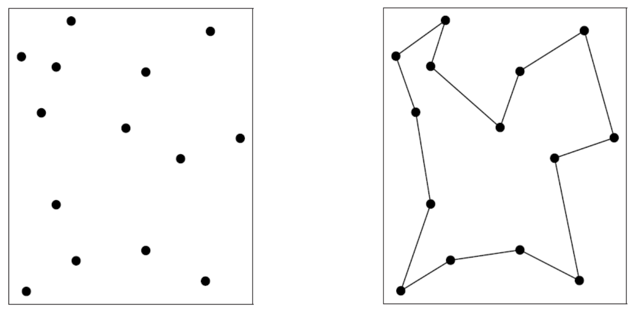
La existencia de un arco $(i, j) \in E$ indica que es posible transportarse desde el sitio representado por i al sitio representado por j. Es usual que a cada arco $(i, j) \in E$ se le asocie un costo $c_{ij} \in C$ que indica la manera más económica de transportarse de i a j. C es la matriz que representa los costos de los arcos.

Cuando se trata de un único depósito, generalmente el mismo se representa con el nodo V_0. Cuando se trata de varios depósitos, se suele dividir el conjunto V en dos subconjuntos V_c y V_d donde V_c ={V_1, V_2,…V_n} es el conjunto de los clientes y V_d ={V_{n+1},V_{n+2}….,V_{n+m} el conjunto de los depósitos [12] [4]. Wen y Meng en [12] utilizan un grafo dirigido en donde c_{ij} representa el costo de transportarse de i a j. Sin embargo, Wang utiliza un grafo sin dirigir ya que c_{ij} representa la distancia entre el nodo i y el nodo j. L. Bodin et al utiliza un grafo no dirigido ya que para él c_{ij} representa también la distancia de ir de i a j y por lo tanto c_{ij} sería igual c_{ji}.

Una ruta es un ciclo simple en G que representa la secuencia de visitas realizadas por el vehículo que recorre la ruta. La ruta comienza y finaliza en el mismo depósito en caso del problema VRP y MDVRP, o ciudad origen en el caso de TSP. El costo de una ruta se obtiene sumando los costos de los arcos que forman el ciclo.

En la mayor parte de los casos G será un grafo completo, pues en una red de transporte real dados dos sitios cualesquiera existe una manera de transportarse de uno al otro.

Como se muestra a continuación, cada problema particular tendrá sus propias características y restricciones de modelado. Se mostrarán algunos ejemplos de modelado y formulación matemática.



(a) Una instancia de del TSP (b) Una solución factible

Figura 1.1: Un TSP de ejemplo y una solución.

3.1 Formulación Matemática de TSP

La siguiente formulación del TSP como problema de programación entera binaria fue propuesta por Dantzig, Fulkerson y Johnson [9] en 1954:

$$Min \sum_{i=1}^{N}\sum_{j=1}^{N} c_{ij}x_{ij} \tag{1.1}$$

s.a.

(1.2)

(1.3)

$$x_{ij} \in \{0,1\} \; \forall \; (i,j) \in E$$

(1.4)

$$\sum_{i=1}^{N} x_{ij} = 1 \; \forall \; j = 1,2,\ldots,N$$

(1.5)

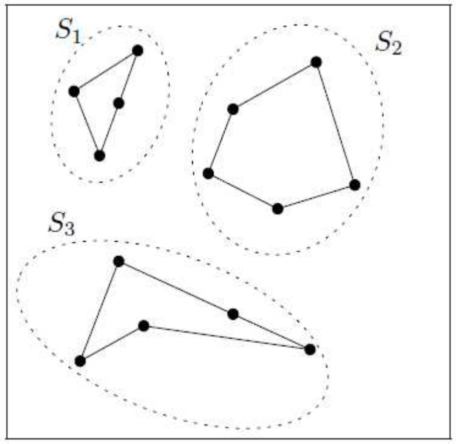


Figura 1.2: Una solución para la instancia de la Figura 1.1(a) formada por 3 sub-tours.

Las variables binarias x_{ij} indican si el arco (i,j) pertenece a la ruta ($x_1ij = 1)$ o no $(x_{ij} = 0)$. La función objetivo (1.1) establece que el costo total de la solución es la suma de los costos de los arcos utilizados. La restricción (1.2) restringe los valores que puede tomar x_{ij} y las restricciones (1.3) y (1.4) indican que la ruta debe llegar y abandonar cada nodo exactamente una vez. Finalmente en la restricción (1.5) se utiliza el conjunto S para prohibir soluciones sub-tour que cumplan las restricciones de asignación (1.2), (1.3) y (1.4). Esta restricción es llamada restricción de eliminación de sub-tours e impone que todo subconjunto de nodos S debe ser abandonado al menos una vez. Si no se impusiera esta restricción se estaría admitiendo soluciones que constan de más de un ciclo, como la que se muestra en la *Figura 1.2*. Esta solución está formada por tres sub-tours y viola la restricción (1.5) para los conjuntos S_1, S_2 y S_3.

Se han propuesto varias alternativas para el conjunto S. Algunas de estas alternativas son:

1. $S = \left\{(x_{ij}): \sum_{i \in B} \sum_{j \in B} x_{ij} \geq 1;\ B \subset E, \qquad B \neq \emptyset\right\}$
2. $S = \left\{(x_{ij}): \sum_{i \in B} \sum_{j \in B} x_{ij} \geq |B| - 1;\ B \subset E, \qquad B \neq \emptyset\right\}$
3. $S = \{(x_{ij}): y_i - y_j + Nx_{ij} \leq N - 1; 1 \leq i \neq j \leq N - 1;\ y_i \in \mathcal{R}\}$

3.2 Formulación Matemática de VRP

El problema de ruteo de vehículos (VRP) como ya se mencionó, se consideró por primera vez por Dantzig y Ramser [7], que desarrollaron un enfoque heurístico utilizando las ideas de programación lineal. En este enfoque los vehículos solo tienen restricciones de capacidad y costo máximo de la ruta que recorren.

A continuación la formulación de este problema [8] [19]

$$Min \sum_{i=1}^{N}\sum_{j=1}^{N}\sum_{k=1}^{V} c_{ij}x_{ijk} \tag{1.6}$$

s.a.

$$\sum_{i=1}^{N}\sum_{k=1}^{V} x_{ijk} = 1 \qquad \forall\, j = 1,2,\dots,N-1 \tag{1.7}$$

$$\sum_{j=1}^{N}\sum_{k=1}^{V} x_{ijk} = 1 \qquad \forall\, i = 1,2,\dots,N-1 \tag{1.8}$$

$$\sum_{i=1}^{N} x_{ihk} - \sum_{j=1}^{N} x_{hjk} = 0 \qquad \forall\, k = 1,2,\dots,V;\; h = 1,2,\dots,N \tag{1.9}$$

$$\sum^{N} Q_i \sum^{N} x_{ijk} \le P_k \qquad \forall\, k = 1,2,\dots,V \tag{1.10}$$

$$\sum_{i=1}^{N}\sum_{j=1}^{N} c_{ij}x_{ijk} \le T_k \qquad \forall\, k = 1,2,\dots,V \tag{1.11}$$

$$\sum_{j=1}^{N-1} x_{Njk} \le 1 \qquad \forall\, k = 1,2,\dots,V \tag{1.12}$$

$$\sum_{i=1}^{N-1} x_{iNk} \le 1 \qquad \forall\, k = 1,2,\dots,V \tag{1.13}$$

$$x_{ijk} \in \{0,1\} \qquad \forall\, i,j,k \tag{1.14}$$

$$y_i - y_j + Nx_{ijk} \le N-1 \qquad 1 \le i \ne j \le N-1, 1 \le k \le V \tag{1.15}$$

Donde

V = Número de vehículos

P_k = Capacidad del vehículo k

T_k = Costo máximo permitido para la ruta de vehículo k

Q_i = Demanda del nodo i , $Q_N = 0$

x_{ijk} = 1, si el par i, j pertenece a la ruta del vehículo k , 0 en otro caso.

En la formulación anterior, las restricciones (1.7) y (1.8) aseguran que cada cliente es atendido por uno y sólo un vehículo. La continuidad de la ruta está representada por (1.9). La restricción (1.10) representa las limitaciones de capacidad del vehículo y (1.11) representa las limitaciones de costo de cada ruta. Las restricciones (1.12) y (1.13) aseguran que la disponibilidad de vehículos no sea superada. Finalmente la restricción (1.15) se utiliza para prohibir soluciones sub-tour. Esta última restricción también se puede es escribir como una desigualdad:

$$y_i - y_j + N \sum_{k=1}^{V} x_{ijk} \leq N - 1 \qquad 1 \leq i \neq j \leq N - 1$$

Se asume que $1 \leq i \leq N$ y $1 \leq k \leq V$,

$\max(Q_i) \leq \min(P_k)$

La demanda en cada nodo es menor o a lo sumo igual a la capacidad de cada vehículo.

3.3 Formulación Matemática de MDVRP

La formulación del problema de MDVRP se presenta a partir de la formulación vista anteriormente de VRP. Siendo $(N + 1 \dots . N + M)$ los M depósitos. Dicha formulación se encuentra en [19], donde además se presentan distintas formulaciones para el mismo problema.

$$Min \sum_{i=1}^{N+M} \sum_{j=1}^{N+M} \sum_{k=1}^{V} c_{ij} x_{ijk} \tag{1.16}$$

s.a.

$$\sum_{i=1}^{N+M} \sum_{k=1}^{V} x_{ijk} = 1 \qquad \forall\, j = 1, 2, \dots, N-1 \tag{1.17}$$

(1.18)

$$\sum_{j=1}^{N+M} \sum_{k=1}^{V} x_{ijk} = 1 \qquad \forall\, i = 1, 2, \dots, N-1$$

(1.19)

$$\sum_{i=1}^{N+M} x_{ihk} - \sum_{j=1}^{N+M} x_{hjk} = 0 \qquad \forall\, k = 1, 2, \dots, V;\; h = 1, 2, \dots, N+M$$

(1.20)

$$\sum^{N+M} \quad \sum^{N+M}$$

$$\sum_{i=1}^{N+M} \sum_{j=1}^{N+M} c_{ij} x_{ijk} \leq T_k \qquad \forall\, k = 1, 2, \dots, V \tag{1.21}$$

$$\sum_{i=1}^{N+M} \sum_{j=1}^{N} x_{ijk} \leq 1 \qquad \forall\, k = 1, 2, \dots, V \tag{1.22}$$

$$\sum_{j=N+1}^{N+M} \sum_{i=1}^{N} x_{ijk} \leq 1 \qquad \forall\, k = 1, 2, \dots, V \tag{1.23}$$

$$x_{ijk} \in \{0,1\} \qquad \forall\, i, j, k \tag{1.24}$$

$$y_i - y_j + (M + N) x_{ijk} \leq N + M - 1 \qquad 1 \leq i \neq j \leq N-1, 1 \leq k \leq V \tag{1.25}$$

V = Número de vehículos

P_k = Capacidad del vehículo k

T_k = Costo máximo permitido para la ruta de vehículo k

Q_i = Demanda del nodo i , $Q_N = 0$

---

x_{ijk} = 1, si el par i, j pertenece a la ruta del vehículo k , 0 en otro caso.

En la formulación anterior, las restricciones (1.17) y (1.18) aseguran que cada cliente es atendido por uno y sólo un vehículo. La continuidad de la ruta está representada por (1.19). La restricción (1.20) representa las limitaciones de capacidad del vehículo y (1.21) representa las limitaciones de costo de cada ruta. Las restricciones (1.22) y (1.23) aseguran que la disponibilidad de vehículos no sea superada. Finalmente la restricción (1.25) se utiliza para prohibir soluciones sub-tour.

4 Revisión de publicaciones de Multi-Depot Vehicle Routing Problem

En el trabajo «A literature review on the vehicle routing problem with multiple depots» [24] publicado por Montoya en febrero del 2015, se muestra el aumento en la cantidad de publicaciones de MDVRP y sus variantes desde la publicación inicial de Kulkarni and Bhave 1985 [19]. El grafico se presente en a continuación en la *Figura 1.3.*

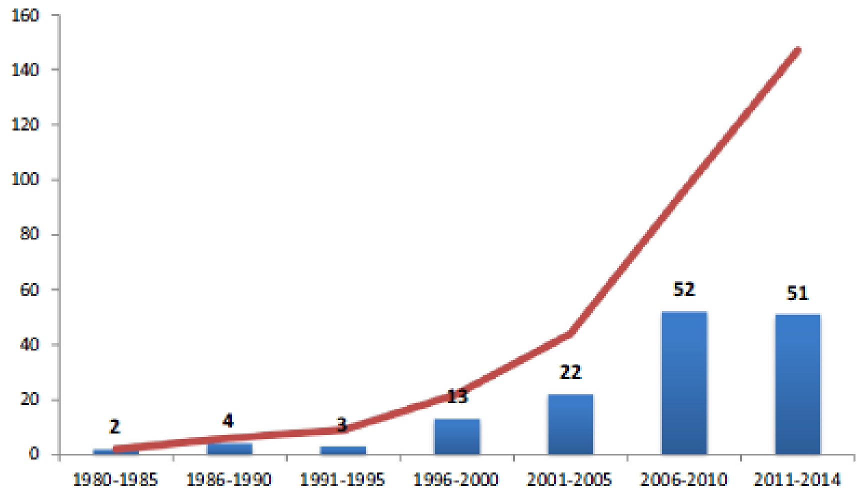


Figura 1.3: Distribución de publicaciones por año de MDVRP

Las variantes de problemas de ruteo de vehículos vistas anteriormente como ventana de tiempo, tipo de flota, periodicidad y recogida y entrega (pick up and deliver) son las que contienen mayor número de publicaciones sobre MDVRP, en cambio otras variantes como variaciones en las rutas Stocastic MDVRP fueron menos estudiadas. Igual así existen numerosas publicaciones de estos tipos de problemas.

Es Importante destacar que al contar con varios depósitos la capacidad de los mismos puede ser una nueva restricción, limitando así el número de clientes asignados a cada depósito. Esta variante en algunos trabajos se referencia como CMDVRP (Capacitated MDVRP) pero en otros simplemente se le llama MDVRP. [25]

Otro aspecto a destacar al contar con más de un depósito; es la variantes de MDVRP en la que los vehículos tengan que pasar por depósitos particulares en su recorrido, o finalizar el recorrido en un depósito distinto al inicial. A esta variante de MDVRP se la conoce como MDVRPI (Multi-Deport Vehicle Routing Problem with Inter-Depot Routes). Existen soluciones para MDVRPI como por ejemplo en [26].

A su vez, el problema de varios depósitos y los recorridos particulares entre los depósitos genera una nueva gama de problemas donde la búsqueda de soluciones está directamente relacionada con MDVRP, como por ejemplo el Track and Trailer Vehicule Routing Problem (TTVRP) estudiado en [27], donde se presente su semejanza con

MDVRP [24]. El problema de TTVRP presentado en [27] optimiza el recorrido para las exportaciones e importaciones así como el plan de rutas para camiones y tráiler dado un conjunto de clientes. Siendo necesario visitar un depósito, puerto y el cliente. En el caso de las exportaciones se inicia el recorrido desde el depósito pasando por el cliente a cagar la mercadería y luego al puerto. Para el caso de las importaciones, se parte del puerto, se descarga el contenedor en el cliente y se lleva el contenedor vacío al depósito. Estos problemas tienen dos particularidades implícitas, una es el manejo de ventanas de tiempo para los clientes y otra es que el objetivo final requiere encontrar la mejor ruta, así como la cantidad mínimas de camiones y los distintos tipos de camiones necesarios.

Otro aspecto a considerar en las publicaciones de MDVRP son las relacionadas a los objetivos finales de la solución. Como se vio anteriormente en las variantes de los problemas de ruteo, el objetivo final puede ser una variable o una función multi-objetivo. En la revisión de MDVRP de 2015 [24] se analizan a grandes rasgos las soluciones multi-objetivo de este problema, en donde se presentan múltiples variables de decisión para la solución final. Muchas veces estos objetivos pueden ser contradictorios como por ejemplo minimizar el número de vehículos y maximizar el nivel de servicios. Existen numerosas publicaciones sobre MDVRP multi-objetivo (MO-MDVRP) aunque su número es mucho menor a las publicaciones de Single Objetive MDVRP.

Según el análisis de 147 publicaciones de MDVRP publicado en [24], aproximadamente 12% Corresponden a MO-MDVRP y entre dichas publicaciones las funciones objetivos varían entre demanda, balanceo de carga de vehículos, número de vehículos, costo/distancia y otras. Centrándose en la mayoría de las publicaciones únicamente en el costo/distancia (un 80%).

Otro aspecto del artículo «A literature review on the vehicle routing problem with multiple depots» [24] es la descripcion de las distintos tipos de soluciones y los metodos de resolucion del problema de MDVRP.

5 Métodos para la resolución de problemas de ruteo de vehículos

A continuación se analizaran distintos métodos de solución para los problemas MDVRP y sus variantes. A lo largo de la presentación de soluciones también se encontraran los problemas clásicos como TSP y VRP a modo de ejemplos de métodos de solución. La solución de estos dos tipos de problemas es la base para la solución de problemas más complejos de MDVRP y sus variantes. La variantes estudiadas tendrán en común las restricciones clásicas de los problemas de ruteo plateadas por Laporte et al 1996 [28].

1. Cada ruta comienza y termina en el mismo depósito.
2. Cada cliente es atendido exactamente una vez.
3. La demanda total de cada ruta no excede la capacidad del vehículo.
4. El costo total de la distribución es minimizado.

Cómo ya se mencionó, se ha demostrado que los problemas de ruteo de vehículos aquí descritos son problemas NP-Duros. Y por lo tanto, el enfoque más habitual para solucionar estos problemas es a través de heurísticas y meta-heurísticas ya que para números grandes de clientes el costo computacional de los métodos exactos es demasiado elevado.

A continuación se hace mención a los métodos desarrollados para encontrar la solución óptima (métodos exactos) en los problemas de ruteos de vehículos para luego pasar a revisar los métodos heurísticos.

5.1 Métodos Exactos

G. Dantzig, R. Fulkerson, y S. Johonson [9] en el año 1954 abordan el problema de encontrar una solución óptima para TSP (Traveling Salesman Problem). Muestran que la cantidad de posibilidades para encontrar la solución óptima es finita. Para n ciudades las posibilidades son ${((n-1)!)}/{2}$. En este estudio se delinea una manera de aproximarse al problema que, al menos, algunas veces permite encontrar el camino óptimo y además probar que el camino encontrado es el óptimo. Se concluye que el método mostrado es factible para encontrar soluciones óptimas, pero únicamente para un número moderado de ciudades. El problema plantado como ejemplo en dicho estudio consta de 49 ciudades.

En [19] R.V. Kulkarni plantea a los problemas de ruteo de vehículos como problemas de Programación Entera Lineal, se concluye que los métodos de solución de estos, aún no han sido suficientemente desarrollados como para abarcarlos en tiempos razonables de cómputo para cantidades grandes de clientes (o ciudades).

En el libro "Survey Combinatorial Optimization" [28], G. Laporte presenta un capítulo sobre algoritmos exactos para el problema de ruteo de vehículos. En el mismo realizan un sondeo sobre los algoritmos exactos existentes hasta el momento y nos muestran que al parecer todos caen en una de las siguientes categorías de tipos de algoritmo:

i) Búsqueda arborescente
ii) Programación dinámica (DP)
iii) Programación entera linean (ILP)

La última categoría es muy extensa y cuenta con el mayor esfuerzo de investigación en los últimos años. Se subdivide en tres subcategorías:

i) Formulación de particionamiento del conjunto
ii) Formulación de flujo de vehículos
iii) Formulación del flujo de mercancía

Entrar en detalle en cada uno de los métodos de solución exacta se aleja del propósito de este estado del arte. Por lo cual se limita únicamente a mencionar la existencia sobre el trabajo realizado al respecto. Haciendo énfasis que para grandes cantidades de clientes (o ciudades, puntos, etc.), los métodos exactos requieren demasiado procesamiento de cómputo, por lo cual el enfoque utilizado para encarar este tipo de problemas (sobre todo con grandes cantidades de clientes) es heurístico. Dicho enfoque aplica para todas las variantes de ruteo de vehículos. Tanto para TSP, VRP, MDVRP, y cualquiera de sus variantes dado que la complejidad de los problemas no parece disminuir en ningún caso.

En [29] se resolvió el problema de MDVRP con ventanas de tiempo y flota heterogénea para un conjunto de 20 clientes. En dicha publicación se resuelve el problema a nivel óptimo a través de programación lineal entera en un tiempo de 0.5 a 3 segundos, sirviendo estos como punto de comparación para futuras investigaciones en el campo de las heuristicas y meta-heuristicas.

5.2 Métodos Heurísticos

A continuación se presenta una breve referencia a lo que se entiende por heurística y meta-heurística.

En [23] dice que las heurísticas (para soluciones de VRP) son procedimientos simples que realizan una exploración limitada del espacio de búsqueda y dan soluciones de calidad aceptable en tiempos de cálculo generalmente moderados. A su vez, se presentan a las meta heurísticas como procedimientos genéricos de exploración del espacio de soluciones para problemas de optimización y búsqueda.

En general las meta heurísticas obtienen mejores resultados que las heurísticas clásicas, pero incurriendo en mayores tiempos de ejecución.

En [30] los autores F. Glover y G. A. Kochenberg, introducen al lector en el libro indicando que:

> *"Las Meta heurísticas son métodos de solución que orquestan una interacción entre procedimientos de mejora local y estrategias a un nivel superior para crear procesos capaces de escapar de óptimos locales y de esta forma realizar una búsqueda más robusta del espacio de soluciones."*

5.3 Heurísticas para VRP

Si bien el objetivo de este "estado del arte" es el problema MDVRP (Multi Depot Vehicle Routing Problem), sería imposible abarcar el problema del mismo sin antes hacer un sondeo de los métodos de solución para problemas con un único depósito (VRP). Por lo cual, a continuación, sin entrar en demasiado detalle, se presentan las heurísticas que se han desarrollado para VRP. La cantidad de heurísticas existentes para este problema es bastante extensa, por lo cual nos vamos a centrar en las clásicas.

5.3.1 Algoritmo de Ahorros de Clarke and Wright

A continuación se muestra la idea general de dicho algoritmo basándose en los apuntes de J. Lysgaard [31] el cuál proporciona además un ejemplo sobre el mismo. La idea es la siguiente:

El concepto básico del "ahorro" expresa el costo ahorrado obtenido por juntar dos rutas en una misma ruta como se ilustra en la imagen a continuación:

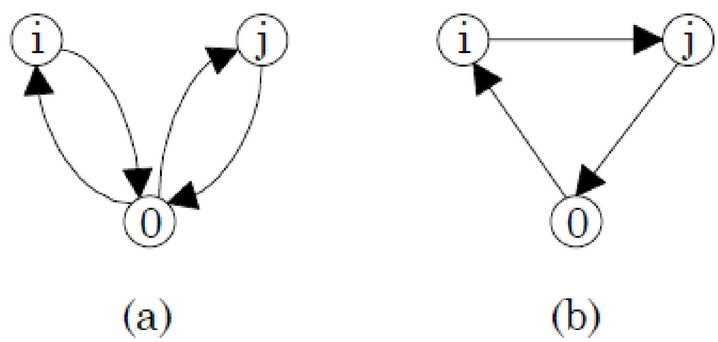


Figura 1.4

Inicialmente los consumidores i y j son visitados en rutas separadas (a). Una alternativa a esto es visitar a los dos clientes en la misma ruta, por ejemplo como lo ilustrado en (b). El ahorro por hacer esto puede ser calculado. Denotando el costo de transporte entre i y j con c_{ij}, el costo total de transporte D_a, en la *figura 1.4 (a)*, es:

$$D_a = c_{0i} + c_{i0} + c_{0j} + c_{j0}$$

De la misma forma, el costo de transporte D_b en la *figura 1.4 (b)* es:

$$D_b = c_{0i} + c_{ij} + c_{j0}$$

Al combinar las dos rutas, se obtiene el siguiente ahorro S_{ij} :

$$S_{ij} = D_a - D_b = c_{i0} + c_{0j} - c_{ij}$$

Los mayores valores S_{ij} indicarán que la unión de esas rutas es más atractiva en comparación con otras de menor ahorro. También se deberán verificar las restricciones del problema como la capacidad del vehículo.

Con esta idea básica sobre el ahorro al unir rutas es que se forma el algoritmo.

Existen distintas variantes y extensiones a la versión básica del algoritmo de ahorros. Como por ejemplo, se puede distinguir entre la versión secuencial y la versión paralela. En la secuencial se construye de a una ruta a la vez, en cambio en la versión paralela se van construyendo varias ruta al mismo tiempo. Además se encuentra la versión del algoritmo basada en matching. Por las mismas y otras extensiones del algoritmo se sugiere consultar [23].

5.3.2 Heurísticas de Inserción

Como se indica en [23], las heurísticas de inserción son métodos constructivos en los cuales se crea una solución mediante sucesivas inserciones de clientes en las rutas. En cada iteración se tiene una solución parcial cuyas rutas sólo visitan un subconjunto de los clientes y se selecciona un cliente no visitado para insertar en dicha solución. En las heurísticas de inserción secuencial sólo se considera insertar clientes en la última ruta creada. La principal desventaja de este enfoque es que los últimos clientes no visitados tienden a estar dispersos y por lo tanto las últimas rutas construidas son de costo muy elevado. Las heurísticas de inserción en paralelo surgen para remediar esta deficiencia, permitiendo insertar un cliente en cualquiera de las rutas de la solución. Esta distinción en similar a la hecha para las dos versiones del Algoritmo de Ahorros de Clarke and Wright.

Existen varias heurísticas del tipo de Inserción. A continuación se muestra a modo de ejemplo la idea general de las heurística de **Inserción Secuencial de Mole & Jameson** [32]

Inserción Secuencial de Mole & Jameson

El algoritmo puede ser pensado en términos de repetir una secuencia de tres pasos. En el primer paso se determina el lugar más ventajoso en la ruta emergente para cada cliente que aún no está asignado. Se utiliza el criterio del costo que se agrega al insertar al cliente entre dos clientes ya pertenecientes a la ruta. Cuando el costo es el mínimo se determina que esa es la posición más ventajosa. Se entiende como ruta emergente a la ruta que se está construyendo en esta etapa del algoritmo.

En el segundo paso se identifica al siguiente cliente no asignado el cuál se va a asignar a la ruta emergente. Aquí se podría utilizar el que implica un costo mínimo también, pero además, generando incentivos adicionales para los clientes que se encuentran más dispersos. De esta forma se evita dejar para el final los clientes lejanos al depósito lo cual podría resultar en rutas ineficientes.

En el tercer paso se explora si alguna reasignación del orden de los clientes en la ruta es más eficiente. Luego se verá algo de optimización en este estado del arte.

5.3.2.1 Heurísticas de Inserción para VRPTW (Solomon)

Solomon en [33] publica un diseño y análisis de heurísticas para problemas de ruteo con restricciones de ventanas de tiempo. Ahí se pueden encontrar varias adaptaciones de heurísticas para VRP para contemplar las restricciones de ventanas de tiempo. Una de ellas es una adaptación de la versión paralela del algoritmo de ahorros de Clarke and Wright. No se va a entrar en detalle sobre estos tipos de heurísticas. Por más información de heurísticas para VRPTW consultar [33]

5.3.3 Heurísticas de dos fases

En [23] se pueden encontrar las siguientes familias de estrategias para heurísticas para VRP. Asignar Primero – Rutear Después por un lado y Rutear Primero – Asignar Después por otro. El termina "Asignar" refiere al hecho de asignar un cliente a una ruta. Y "Rutear" como el nombre lo indica es establecer la ruta.

Como se indica en [23], en el caso Asignar Primero – Rutear Después se busca generar grupos de clientes, también llamados clusters, que estarán en una misma ruta en la solución final. Luego, para cada cluster se crea una ruta que visite todos los clientes. Las restricciones de capacidad son consideradas en la primera etapa, asegurando que la demanda total de cada cluster no supera la capacidad del vehículo. Por lo tanto, construir las rutas para cada cluster es un TSP. Un ejemplo de esta estrategia es la heurística de Barrido o Sweep, en la cual los clusters se forman girando una semirrecta con origen en el depósito e incorporando los clientes “barridos” por dicha semirrecta hasta que se viole la restricción de capacidad. Luego cada cluster es ruteado resolviendo un TSP.

En este caso solo se tienen en cuenta restricciones de capacidad de los vehículos. Sería distinto el algoritmo con más restricciones, como por ejemplo con la restricción de que el largo total de una ruta no puede ser mayor a un valor predeterminado.

En las heurísticas de Rutear Primero-Asignar Después, lo primero que se hace, como lo dice el nombre de la estrategia, es calcular una ruta que visite a todos los clientes. Para hacer esto basta con resolver un problema TSP. En general esta ruta no va a respetar las restricciones del problema y se debe partir en varias rutas, cada una de las cuales, respetaría las restricciones [23].

5.4 Heurìsticas para MDVRP

Se han propuesto numerosas heurísticas para el problema de enrutamiento de vehículos con múltiples depósitos en los últimos años [24], este tema se abordó a inicios de la década del 90´ y fueron estudiados MDVRP así como sus variantes con flotas heterogéneas, periodicidad, ventanas de tiempo, etc. A continuación se detalla una heurística de dos fases aplicada a MDVRP; existiendo además otras heurísticas como se describe en la revisión de publicaciones de Montoya [24].

5.4.1 Heurísticas de dos Fases para MDVRP y sus variantes

Un enfoque estudiado en [34] por Libertad Tansini y Omar Viera, utiliza una heurística de dos fases donde la primera fase es de asignación de clientes a depósitos (zonificación, asignación, clusterización) y la segunda es la resolución de rutas para cada depósito. Es necesario destacar que aunque la heurística es del tipo de dos fases no es igual a la estudiada anteriormente para el caso de VRP ya que la etapa de asignación es de clientes a clústeres (zonas) y no de clientes a rutas.

5.4.1.1 Etapa de asignación

Para la asignación de clientes a zonas, en la publicación [34] se comparan los resultados obtenidos por los distintos algoritmos. A continuación se describen algoritmos de asignación por urgencia, barrido, cíclica y por zona. Luego se incluirá la variación de ventanas de tiempo [35] para el caso de asignación por urgencias y asignación por zonas.

1) Asignación a través de urgencias

La urgencia o prioridad que tienen los clientes determina la forma de asignarlos. Un cliente con más urgencia se asigna primero. La urgencia es una manera de definir una relación de precedencia entre los clientes.

A continuación se mencionan dos formas de asignar a través de urgencias, asignación en paralelo y asignación simplificada.

1.1) Asignación en paralelo

En esta asignación la urgencia para cada cliente se calcula teniendo en cuenta todos los depósitos al mismo tiempo.

Se calcula como:

$$\mu_c = \left(\sum_{dep \in D} d(c, dep) \right) - d(c, dep')$$

Donde $d(c, dep)$ es la distancia entre el cliente c y el depósito dep , D es el conjunto de depósitos y $d(c, dep')$ es la distancia entre el cliente c y el depósito más cercano dep' . El cliente con mayor valor de μ_c será asignado al depósito más cercano.

Esta heurística compara el costo de la asignación de un cliente para su depósito más cercano con el costo de asignar el cliente a cualquier otro depósito.

1.2) Asignación simplificada

En esta heurística sólo dos depósitos están implicados en la evaluación de la urgencia de cada cliente mediante el cálculo:

$$\mu_c = d(c, dep'') - d(c, dep')$$

Donde $d(c, dep'')$ es la distancia entre el cliente c y el segundo depósito más cercano dep'' y $d(c, dep')$ es la distancia entre el cliente c y el depósito más cercano dep' . El cliente con mayor valor de μ_c será asignado al depósito más cercano.

Esta heurística compara el costo de asignar un cliente al depósito más cercano con el costo de asignar a un cliente con el segundo depósito más cercano.

2) Barrido de asignación

En esta heurística, los clientes son "barridos" en la dirección del depósito con mayor demanda insatisfecha.

En primer lugar, es necesario determinar un depósito dep * con mayor demanda insatisfecha. La evaluación de la urgencia se realiza con el siguiente cálculo:

$$\mu_c = d(c, dep^*) - d(c, dep')$$

La urgencia se mide como la diferencia entre la asignación de un cliente para su depósito más cercano y el depósito con mayor demanda insatisfecha, por lo tanto un valor elevado en la urgencia significa que debería ser asignado al depósito más cercano.

3) Asignación Cíclica

El procedimiento consiste en asignar de una manera cíclica, un cliente a la vez. En primer lugar, el algoritmo asigna a cada depósito el cliente más cercano. Luego se asigna a cada depósito, el cliente más cercano al último cliente asignado a dicho depósito. En general, la asignación es muy pobre para los últimos clientes asignados.

4) Asignación por Zona

Una zona está definida por un depósito y los clientes asignados a él. El algoritmo trata de construir un grupo compacto de clientes para cada depósito. Se analizan dos algoritmos para la resolución de dicho problema. [23]

4.1) Propagación de coeficientes

Para esta forma se definen coeficientes de atracción para los depósitos y para los clientes ya asignados a depósitos. La asignación se hace en función de una distancia escalada (definida a continuación) con los clientes no asignados. El siguiente cliente a asignar es el que minimiza la distancia escalada, siendo la fórmula de la distancia escalada la siguiente:

$$d_{scaled}(c, c') = d(c, c') * coef_{c'}$$

El cálculo del coeficiente de atracción de *c* que fue asignado por su cercanía al cliente *c´* es el siguiente:

$$[\![coef]\!]_{\downarrow c} = \min [\![(1, [\![coef]\!]_{\downarrow}(c')]\!] + ([\![coef]\!]_{\downarrow}(c') * [\![deg]\!]_{\downarrow}(c'))$$

El coeficiente de degradación $deg_{c'}$ es valorado arbitrariamente para todos los depósitos y clientes (ejemplo 0.5). Los nuevos clientes asignados tienen coeficientes más altos por lo cual tienen menos atracción.

4.2) Zonificación por 3 criterios

El procedimiento utilizado por este algoritmo para incluir clientes en un clúster es en base a la distancias promedio a los clústeres, la varianza de la distancia a los clientes en los clústeres y distancia al cliente más cercano en cada clúster. Se aplican cotas que para cada uno de estos 3 criterios, aplicando el segundo si el primero no llega a la cota y el tercero si el segundo no llega a la cota. Por ejemplo, para un cliente y un clúster, dichas cotas se establecen comparando el porcentaje de mejora con respecto a este cliente con los otros clúster.

5) Métodos de Asignación con ventanas de tiempo.

En [35] se modifican los métodos de Asignación por zona para considerar ventanas de tiempo. De este modo los clientes con ventanas de tiempo similares son agrupados en la misma zona (clúster).

En dicha publicación se definen los conceptos de *Afinidad* y *Cercanía* para luego presentar el criterio de asignación paralela simplificada (dentro de la categoría Asignación por urgencia). Finalmente se presenta la asignación por zonificación de 3 criterios (dentro de la categoría Asignación por Zona) la cual se calcula en función de la distancia *Angulo*.

Definiciones previas.

Afinidad (i,d) = $\{(\sum_{j \in C(d) \cup \{d\}} e^{-(DTW(i,j) + TV(i,j))})/|C|\}$

Donde C son los Clientes y D son los Depósitos

DTW(i,j) mide la distancia en la ventana de tiempo entre i y j (siendo i y j clientes o depósitos). Para el cliente *i*, e_i y l_i representan el inicio y fin de la ventana de tiempo.

$$DTW(i,j) = \begin{cases} e_j - l_i \; si \; l_i > e_j \\ e_i - l_j \; si \; l_j < e_i \\ 0 \; en \; otro \; caso \end{cases}$$

TV(i,j) es el tiempo de viajar de i a j.

Cercania(i,j) = d(i,j) / Afinidad (i,d).

d(i,j) es la distancia entre i y j.

$$\textbf{\textit{Angulo (i,j)}} = Cos^{-1}\left(\frac{x_i x_j + y_i y_j + y_i y_j + e_i e_j}{\left(x_i^2 + y_i^2 + l_i^2 + e_i^2\right)^{\frac{1}{2}}\left(x_j^2 + y_j^2 + l_j^2 + e_j^2\right)^{\frac{1}{2}}}\right)$$

Donde *x* e *y* son las coordenadas, *l* y *e* es el inicio y fin de la ventana de tiempo.

5.1) Asignación por urgencia con ventanas de tiempo, Asignación paralela simplificada.

Ecuación de Urgencias, solo dos depósitos son incluidos en esta ecuación.

$$\mu_c = \textbf{\textit{Cercania}}\left(c, dc'(c)\right) - \textbf{\textit{Cercania}}\left(c, dc''(c)\right)$$

La variable c es un cliente, $dc'(c)$ es el depósito más cercano a c y $dc''(c)$ es el segundo depósito más cercano.

5.2) Asignación por Zona con ventanas de tiempo, Zonificación por 3 criterios.

Para este caso la distancia es en base a distancia *Angulo* definida anteriormente. La misma parte de la comparación de vectores al explorar los algoritmos de "clustering" [35].

Los tres criterios son el distancia *Angulo* promedio al clúster, varianza de la distancia *Angulo* promedio al clúster y distancia *Angulo* al cliente más cercano del clúster.

En la publicación antes mencionada se ejecutaron distintas corridas de este algoritmo en donde se observa que el cálculo en base a distancia *Angulo* es mejor que el cálculo en base a la distancia en un mapa o euclidiana.

6) Heurística de asignación Hibrida para depósitos con capacidad limitada.

En [25] se presenta una heurística hibrida entre la asignación Simplificada (Heurística de Urgencia) y zonificación por tres criterios (Asignación por zona).

Para esto se parte de que cada cliente pertenece a uno de estos conjuntos.

1) CSA, clientes no asignados a ningún depósito.

2) $clust_i$, conjunto de clientes asignados al depósito i .

Por otro lado la asignación de un cliente a un depósito es factible si:

1) El depósito D_i puede cumplir la demanda del cliente.

2) El cliente pertenece CSA

3) Si cumple la condición de urgencia u_j calculada para cada cliente *CEj*.

Para esta heurística, u_j es la minimización de una función que toma como parámetros el costo de asignar el cliente al depósito más cercano en comparación con el próximo más cercano. El parámetro u_j es calculado para todos los clientes *CEj*. Como muestra la siguiente ecuación.

$$\mu_{CEj} = costo(CEj, dep^*) - costo(CEj, dep')$$

Si después de la evaluación de la función μ_{CEj} es positivo se asigna al deposito dep^* , sino se asigna al que haga mínima μ_{CEj} .

5.4.1.2 Etapa de Construcción de rutas para cada depósito

Una vez que los clientes han sido asignados al depósito, fin de la etapa de asignación, se procede a realizar el enrutamiento. Para la resolución se utiliza cualquier algoritmo de ruteo, los n clientes están asignados a los m depósitos, ahora el problema de ruteo de MDVRP se puede ver como m problemas de ruteo de VRP, donde se utiliza un algoritmo de resolución de VRP para cada uno de los depósitos. En [34] se utilizó el algoritmo de Clarke and Wright.

En otras publicaciones se crearon nuevas formas de resolver este problema a través de meta-heurísticas las cuales explicaremos en la sección 5.5. La heurística de asignación hibrida vista en la sección de zonificación fue utilizada para generar la solución inicial, luego de esto en [25] se presenta la Meta-heurística MOSS (Multi Objetive Scatter Search) en donde a partir de un conjunto de soluciones aleatorias, y a través de selecciones sistemáticas y estratégicas se obtiene una mejor solución creando las rutas y optimizando la asignación de clientes a depósitos de la etapa de zonificación. La construcción del conjunto de referencia para el MOSS y metodologías aplicadas se encuentra detallada en [22].

5.5 Meta-Heurísticas para VRP

Siguiendo con la referencia de [23], se presenta a continuación meta heurísticas para la resolución de problemas VRP. Algoritmos de hormigas, Búsquedas Tabú y Algoritmos Genéticos son meta heurísticas representantes de tres paradigmas diferentes. Los Algoritmos de Hormigas son procedimientos basados en agentes que utilizan métodos constructivos aleatorizados y cooperan entre si compartiendo información. Los algoritmos de búsqueda Tabú son métodos de búsqueda local que aceptan empeorar las soluciones para escapar de los óptimos locales. Los Algoritmos Genéticos se basan en mantener un conjunto de soluciones lo suficientemente diverso como para cubrir gran parte del espacio de soluciones.

5.6 Meta–Heurísticas para MDVRP

A continuación se muestran algunas meta-heurísticas desarrolladas para resolver problemas de ruteos de vehículos con varios depósitos.

5.6.1 Particle Swarm Optimization (PSO)

Lei Wen y Fanhua Meng, publicaron en el 2008 el artículo “An Improved PSO for the Multi-Depot Vehicle Routing Problem with Time Windows”. [12]

En el mismo explican el algoritmo PSO utilizado. Particle Swarm Optimization. En castellano, Optimización por Enjambre de Partículas es un algoritmo de optimización basado en la teoría de partículas. La idea principal de PSO es modelar el vuelo de un enjambre de pájaros (partículas) alrededor de una cumbre. El estado de una partícula en el espacio de búsqueda está definido por dos factores: posición y velocidad. La posición y velocidad de la partícula i en un espacio de búsqueda de d-dimensiones puede ser representado como $X_i = (x_{i1}, x_{i2}, x_{i3}, \dots x_{id})$ y $V_i = (v_{i1}, v_{i2}, v_{i3}, \dots v_{id})$ respectivamente. Cada partícula conoce su mejor posición $P_i = (p_{i1}, p_{i2}, p_{i3}, \dots p_{id})$ que obtuvo en su camino recorrido. Y también conoce la mejor posición global, denotado como $P_g = (p_{g1}, p_{g2}, p_{g3}, \dots p_{gd})$ que es la mejor posición que obtuvo alguna partícula del enjambre. En cada iteración se calcula una nueva velocidad para cada partícula.

$$v_{id}^{k+1} = wv_{id}^{k} + c_1 r_1 * \left(p_{id} - x_{id}^{k}\right) + c_2 r_2 * \left(p_{gd} - x_{id}^{k}\right)$$

En donde c_1 y c_2 son constantes llamadas coeficientes de aceleración, w es llamada peso del factor de inercia, r_1 y r_2 son dos números aleatorios distribuidos uniformemente en el rango [0,1]. Por lo tanto, la posición de cada partícula es actualizada en cada iteración de acuerdo a la siguiente ecuación:

$$v_{id}^{k+1} = x_{id}^{k} + v_{id}^{k+1} \qquad 1 \leq i \leq n; 1 \leq d \leq D$$

En el caso de utilizar la meta-heurística PSO en el problema MDVRPTW, las posiciones de las partículas son las soluciones candidatas para el ruteo. En cada posición se calcula el costo total para poder compararlas y de esta forma saber cuál es la mejor posición conocida de cada partícula y la mejor posición global.

El primer paso en la solución planteada es encontrar una solución factible inicial. Para esto se asignan los clientes al depósito más cercano. Luego para cada depósito se busca una solución para el ruteo utilizando al algoritmo de ahorros de Clarke and Wright. Luego se establece la posición inicial para un conjunto de partículas variando aleatoriamente la solución factible recién encontrada. Para esto se cambian aleatoriamente algunos clientes de depósito y se asignan a nuevas rutas. Luego de esta etapa inicial comienza una etapa iterativa en donde las partículas van buscando el óptimo en el espacio de soluciones.

Para esto es necesario codificar las soluciones de ruteo en posiciones de partículas en un espacio multidimensional. También es necesario tener en cuentas que el algoritmo PSO aplica en espacios continuos, cuando las soluciones de ruteo están en un espacio discreto. No se entra en detalle en este documento de cómo se codifican las partículas y de cómo se calcula la actualización de la posición de la partícula en cada iteración.

El objetivo es mostrar cómo es que la meta-heurística PSO explora el espacio de soluciones.

5.6.2 Tabu Serch

En [36] “A Tabú Search Heuristic For The Multi-Depot Vehicle Routing Problem, la estrategia que utiliza la meta-heurística Tabú Search también comienza a partir de una solución inicial. Para esto se asignan los clientes al depósito más cercano y luego se rutean los vehículos utilizando una heurística para VRP.

Luego que se tiene la solución inicial, Tabú Search se basa en un algoritmo llamado FIND. El mismo consiste en tres fases: Fast Improvement, INtensification, y Diversification.

Cada una de estas fases se basa en alguno, o todos, de los siguientes procedimientos básicos:

1-route: Se utiliza para post-optimizar una ruta simple de vehículos. Se basa en aplicar el algoritmo 4-opt [37]. Este algoritmo de mejora parte de una ruta inicial y prueba todas las combinaciones posibles que surgen de cambiar 4 aristas del grafo a la vez (viendo a la ruta como un grafo con vértices y aristas donde el vértice representa al cliente y la arista el camino entre dos clientes). De esta forma se busca el óptimo local en el espacio de soluciones.

2-route: Se utiliza para buscar mejoras en la solución moviendo clientes provenientes de dos rutas distintas. Podrían ser rutas asignadas a depósitos distintos.

3-route: Similar a 2-route pero se intercambian clientes provenientes de 3 rutas.

Al aplicar FIND, la mejor solución encontrada y su valor es recordado, pero como se permite deteriorar la función objetivo, la solución tenida en cuenta en el algoritmo en cada iteración no es necesariamente la mejor solución. Cuando un cliente es movido de su ruta actual, moverlo nuevamente a su ruta original es declarado un movimiento tabú por un número aleatorio de iteraciones.

A continuación se explican las tres fases de FIND:

<u>Fast Improvement</u>: En esta fase, el algoritmo intenta mejorar aplicando repetidamente los siguientes pasos:

- Inter-depot: aplicar intercambios 2-route entre rutas de dos diferentes depósitos.
- Intra-depot: aplicar intercambios 2-route entre rutas del mismo depósito.
- 3-route: intercambiar clientes entre tres rutas.

La secuencia es repetida hasta que no hay mejoras por θ iteraciones consecutivas. En cada uno de estos tres pasos, cualquier movida que encuentra una mejora, es implementada inmediatamente como la solución actual a tomar en cuenta. En otro caso, la mejor solución no-tabú que deteriora la función objetivo, es implementada. Cuando se implementa un movimiento de clientes entre rutas, se aplica 1-route a todas las rutas involucradas en el movimiento.

Intensification:

Esta fase intensifica la búsqueda de rutas mejores, comenzando con la mejor solución conocida y trabajando con un único depósito a la vez. Aplica iteraciones de intra-depot a las rutas de cada depósito.

Diversification:

Esta fase aplica una combinación Inter-depot e Intra-depot, moviendo clientes entre distintos depósitos, con el objetivo de realizar una exploración más extensa del espacio de soluciones.

6 Post Optimización y mejoras

Una vez que se tiene una solución para el problema, se puede intentar mejorarla 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. Para obtener s^* puede buscarse la mejor solución de $N(s)$ o tomar la primera solución de $N(s)$ que mejore el costo.

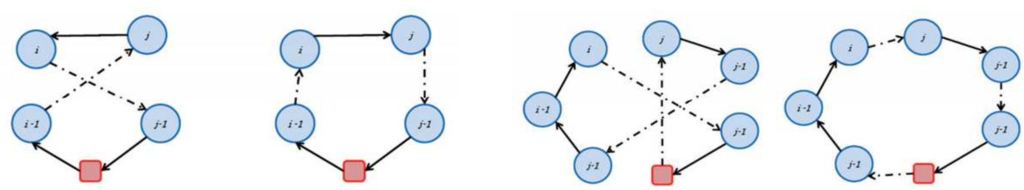
Usualmente se define $N(s)$ como las soluciones que pueden obtenerse aplicando a s alguna regla o procedimiento sencillo llamado movida. Las movidas para el VRP pueden clasificarse en movidas de una ruta y movidas multi-ruta. En las movidas de una ruta los clientes que se visitan en una ruta no varían luego de la aplicación del operador, lo que varía es el orden en que se realizan las visitas. En las movidas multi-ruta, además de cambios en el orden de las visitas suele modificarse el conjunto de clientes visitados en cada ruta.

6.1 Operador λ–Intercambio

El λ-intercambio definido por Lin [38] consiste en eliminar λ arcos (camino entre dos clientes consecutivos) de la solución (λ > 1) y reconectar los λ segmentos restantes. Una solución se dice λ-óptima si no puede ser mejorada utilizando λ-intercambios. Se llama λ-opt a un algoritmo de búsqueda local que utiliza λ-intercambios hasta alcanzar una solución λ-óptima.

El algoritmo λ–opt divide la ruta en λ segmentos y los reordena de todas las formas posibles para obtener nuevas soluciones que sean mejores. La división en segmentos también se hace de todas las formas posibles. La combinación de todas las posibles divisiones en segmentos con todas las posibles reordenaciones de los segmentos da lugar a una gran cantidad de nuevas soluciones. En cada paso del algoritmo, se procede a generar todas las nuevas soluciones de forma sistemática. Durante la generación, el algoritmo puede presentar dos tipos de comportamiento: cuando encuentre una solución mejor que la actual, la devuelve; o espera hasta haber explorado todas las nuevas soluciones y devuelve la mejor de ellas [39].

Este algoritmo es $\mathbf{O(n^{\lambda})}$ en el peor caso, siendo $\mathbf{n}$ el número de clientes. Usualmente se implementan 2-intercambios y 3-intercambios. Posibles movidas de este tipo se muestra en la Figura 1.5



(a) Ejemplo de 2-intercambios (b) Ejemplo de 3-intercambios

Figura 1.5 Ejemplo de 2-intercambios

6.2 Operador Or–opt

Una versión reducida del algoritmo 3-opt es el algoritmo Or-opt [40], que consiste en eliminar una secuencia de k clientes consecutivos de la ruta y colocarlos en otra posición de la ruta, de modo que permanezcan consecutivos y en el mismo orden. Primero se realizan las movidas con $k = 3$, luego con $k = 2$ y finalmente con $k = 1$. Si una ruta visita n clientes existen $O(n^2)$ de estas movidas. En la Figura 1.6 se muestra un ejemplo del algoritmo Or-opt.

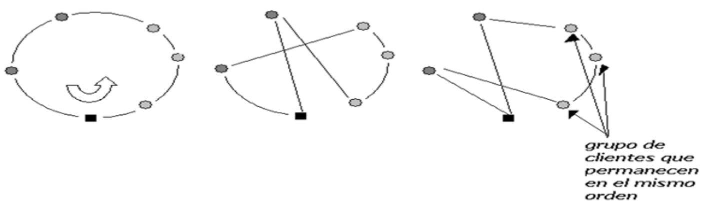


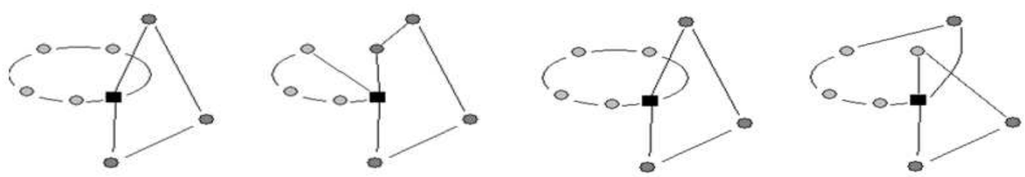
Figura 1.6 Ejemplo de algoritmo Or-opt

6.3 Operadores de Van Breedam

Van Breedam [41] propuso dos operadores para intercambiar clientes entre un par de rutas:

- Operador String Relocation (SR): Una secuencia de m clientes es transferida de una ruta a la otra manteniendo el orden en la ruta original.
- Operador Exchange (SE): Una ruta envía una secuencia de m clientes a la otra y esta última envía otra secuencia de n clientes a la primera.

En la figura 1.7 se muestran la aplicación de los operadores String Relocation y String Exchange.



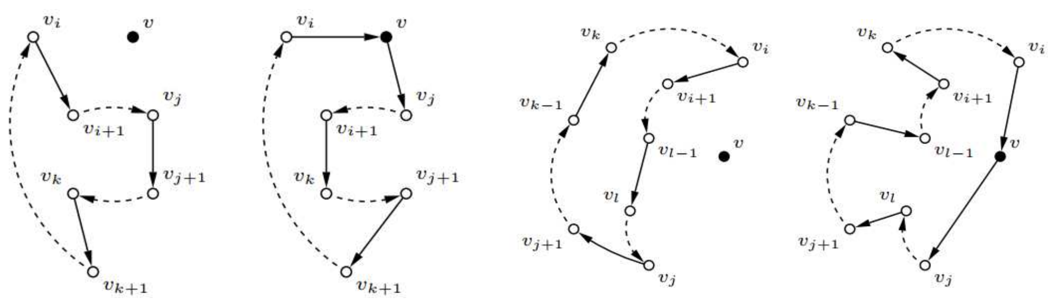
(a) Operador String Relocation (b) Operador String Exchange

Figura 1.7 Operadores String Relocation y String Exchange

6.4 GENI y GENIUS

6.4.1 GENI

Las inserciones generalizadas [42] (GENI) tienen como principal característica que la inserción de un cliente v en una ruta no necesariamente ocurre entre dos clientes adyacentes. Sin embargo, después de la inserción estos dos clientes se convierten en adyacentes a v . Si v_k es un cliente de la ruta, denotamos v_{k+1} a su sucesor y v_{k-1} a su predecesor. La inserción de un cliente v entre los clientes v_i y v_j (no necesariamente consecutivos) se puede realizar de dos formas, se ilustran en la Figura 1.8



(a) Inserción GENI de Tipo I (b) Inserción GENI de Tipo I

Figura 1.8 Inserción GENI (GENeralizad Insertions)

Inserción Tipo I

Se considera un cliente v_k en el camino de v_j a v_i ($v_k \neq v_i$ y $v_k \neq v_j$). La inserción consiste en eliminar los arcos (v_i, v_{i+1}), (v_j, v_{j+1}) y (v_k, v_{k+1}) y agregar los arcos (v_i, v), (v, v_j), (v_{i+1}, v_k) y (v_{j+1}, v_{k+1}). Es necesario además, invertir el sentido de los caminos $(v_{i+1}, \dots, v_j)$ y $(v_{j+1}, \dots, v_k)$. Una inserción de este tipo se muestra en la Figura 1.8 (a).

Inserción Tipo II

Se debe seleccionar v_k en el camino de v_j a v_i de modo que $v_k \neq v_j$ y $v_k \neq v_{j+1}$ y además v_l en el camino de v_i a v_j tal que $v_l \neq v_i$ y $v_l \neq v_{i+1}$. La inserción consiste en eliminar los arcos (v_i, v_{i+1}), (v_{l-1}, v_l), (v_j, v_{j+1}) y (v_{k-1}, v_k), y agregar los arcos (v_i, v), $[\![(v, v]\!]_j)$, (v_l, v_{j+1}), (v_{k-1}, v_{l-1}) y (v_{i+1}, v_k). Además, debe revertirse el orden de los caminos $(v_{i+1}, \cdots, v_{l-1})$ y $(v_l, \cdots, v_j)$. Una inserción de este tipo se muestra en la Figura 1.8 (b).

6.4.2 GENIUS

El algoritmo de post-optimización GENIUS [42] comienza considerando el primer cliente de la ruta: se le elimina mediante Unstringing y se le vuelve a insertar utilizando Stringing.

El procedimiento Stringing consiste en realizar una inserción de GENI de Tipo I o Tipo II con *p-vecindades* y el Unstringing es el inverso de una inserción GENI de Tipo I o Tipo II. La *p-vecindad* del cliente v se define como el conjunto de los p clientes de la ruta más cercanos a v según los costos del problema.

El proceso de eliminar mediante Unstringing y volver a insertar utilizando Stringing puede incrementar el costo de la solución. Si la solución es mejorada, se repite el proceso con el segundo cliente de la nueva ruta. El proceso termina luego de eliminar e insertar el último cliente de la ruta.

7 Soluciones de software existentes para MDVRP

En Febrero de 2014 el sitio “OR/MS Today” público un informe donde se provee la información de los programas existentes para el problema de enrutamiento de vehículos [43]. El mismo es extraído de un cuestionario a los distintos proveedores. La encuesta está dividida en 17 secciones que abarcan las especificaciones técnicas de dichos programas. En la tabla 1.1 se puede observar las distintas soluciones de software existentes para los problemas de MDVRP:

Producto	Vendedor	Año	Plataformas que soporta				
			Windows	iOS	Android	Software basado en Web como servicio	Otros
ArcGIS for TransportationAnalytics	Esri	2012	Si	Si	Si	Si	
Descartes Routing, Mobile & Telematics	Descartes	1995	Si	No	Si	Si	
DISC	MJC2	1990	Si	Si	Si	Si	UNIX & Linux
eRoute Logistics	WM Logistics	2002	Si	No	Si	Si	
JOpt.SDK	DNA Evolutions GmbH	2006	Si	No	No	Si	
Logistics Optimizer	Runzheimer International	2005	No	No	No	Si	
Logvrp	Netakil	2010	Si	Si	Si	Si	REST + SOAP Web Service API
Optrak vehicle routing software	Optrak Distribution Software Ltd	1992	Si	No	No	Si	
ORTEC Routing and Dispatch	ORTECÂ	2003	Si	Si	Si	Si	
Paragon Routing and Scheduling Optimizer	Paragon Software Systems, Inc	1997	Si	No	No	Si	
Roadnet Transportation Suite	Roadnet Technologies	1983	Si	Si	Si	Si	
Routist.com	KKT srl	2013	No	No	No	Si	
StreetSync Basic	RouteSolutions	2008	Si	No	No	Si	
StreetSync Desktop	RouteSolutions	2005	Si	No	No	No	
StreetSync Pro	RouteSolutions	2011	Si	No	No	Si	
TMW Appian DirectRoute	TMW Systems	1996	Si	No	No	Si	
TruckStops	MapMechanics	1983	Si	No	No	No	Server or PC

Tabla 1.1 Distintas soluciones de software existentes para los problemas de MDVRP

En la tabla 1.1 se puede observar que la gran mayoría del software relevado se desarrolló en los últimos años, eso reafirma el aumento en la cantidad de publicaciones sobre los problemas de ruteo de vehículos.

Las distintas plataformas que soporta el software relevado se puede ver que casi el 100% es basado en Web services. Las ventajas de contar con este tipo de servicio es que no se encuentra atado a ningún sistema operativo ni ningún lenguaje de programación y se encuentra disponible a través de Internet o Intranet.

La gran mayoría del software puede ejecutarse en Windows (en sus diferentes versiones) y actualmente han surgido software para Android e iOS como por ejemplo ArcGIS for TransportationAnalytics de ESRI. Otros sistemas que ya tienen sus años se han actualizado para poder soportar los nuevos sistemas operativos como por ejemplo Roadnet Transportation Suite de Roadnet Technologies.

Dentro de los nuevos sistemas operativos, Android es el que soporta más software en comparación con iOS.

También hay otras plataformas que soportan software de ruteo de vehículos como UNIX & Linux se puede ejecutar el software DISC. El software logvrp soporta REST + SOAP Web Service API, donde son los servicios SOAP son Web services que estandarizan su información a través de mensajes XML, en formato SOAP. Los servicios REST son Web services que estandarizan su información a través de mensajes XML, en formato SOAP.

8 Bibliografía

[1] I. Gallegos Mateos, A. Gómez Gómez y D. Arguelles Martino, «A hybrid method for the resolution of the MDVRP,» pp. 45-64, 2013.

[2] A. Schrijver, «On the History of Combinatorial Optimization,» 1960.

[3] L. Bodin, «Routing and scheduling of vehicles and crew: The State of the Art,» *Comput. & Ops Res.,* vol. 10, nº 2, pp. 63-211, 1983.

[4] Y. Wang, «Research of Multi-Depot Vehicle Routing Problem by Cellular Ant Algorithm,» *Journal of Computers,* vol. 8, nº 7, pp. 1722-1727, 2013.

[5] Surekha y S. Sumathi, «Solution to Multi-Depot Vehicle Routing Problem Using Genetic Algorithms,» *World Applied Programming,* vol. 1, nº 3, pp. 118-131, 2011.

[6] J. Carlsson, D. Ge, A. Subramaniam, A. Wu y Y. Ye, «Solvin Min-Max Multi-Depot Vehicle Routing Problem,» 2006.

[7] G. B. Dantzig y J. H. Ramser, «The Truck Dispatching Problem,» *Management Science,* vol. 6, nº 1, pp. 80-91, 1959.

[8] B. L. Golden, «Vehicle Routing Problems: Formulations and Heuristic Solution Techniques,» *Technical Reports,* nº 113, 1975.

[9] G. Dantzig, D. Fulkerson y S. Johnson, «Solution of a Large-Scale Traveling-Salesman Problem,» *Journal of the Operations Research Society of America,* vol. 2, nº 4, pp. 393-410, 1954.

[10] R. M. Karp, «Reducibility Among Combinatorial Problemas,» 1971.

[11] S. N. Kumar y R. Panneerselvam, «A Survey on the Vehicle Routing Problem and Its Variants,» *Intelligent Information Management,* nº 4, pp. 66-74, 2012.

[12] L. Wen y F. Meng, «An Improved PSO for the Multi-Depot Vehicle Routing Problem with Time Windows».

[13] R. Baldacci, M. Battarra y D. Vigo, «Routing a Heterogeneous Fleet of Vehicles,» *Technical Report DEIS OR.INGCE 2007/1,* 2007.

[14] C. Tan y J. Beasley, «A Heuristic Algorithm for the Period Vehicle Routing Problem,» *OMEGA Int. J. of Mgmt Sci.,* vol. 12, nº 5, pp. 497-504, 1984.

[15] P. Francis y K. Smilowitz, «The Period Vehicle Routing Problem with Service Choice,» *Teansportation Science,* vol. 40, nº 4, pp. 439-454, 2006.

[16] A. Mingozzi y A. Valleta, «An exact algorithm for period an multi-depot vehicle routing problems».

[17] S. Moscatelli, «Split Delivery Vehicle Routing Problem: Heuristic based Algorithms,» 2007.

[18] J. Rodriguez Perez, «Caracterización, Modelado y Determinación de las Rutas de la Flota en una Empresa de Rendering,» 2012.

[19] R. V. Kulkarni y P. R. Bhave, «Integer programming formulations of vehicle Routing Problems,» *Eurorean Journal of Operational Research,* vol. 20, pp. 58-67, 1985.

[20] A. Garcia-Najera y J. A. Bullinaria, «Bi-objective Optimization for the Vehicle Routing Problem with Time Windows,» School of Computer Science, University of Birmingham.

[21] R. Bowerman, B. Hall y P. Calamai, «A Multiobjetive Optimization Approach to Urban School Bus Routing: Formulation and Solution Method,» 1995.

[22] S. N. Lopez Franco, «Metodologías Diseñadas para la construcción y actualización dede Referencia y sus operadores evolutivos generados por un metaheuristica híbrida multiobjetivo basada en Busqueda Dispersa para la solucion de un MDVRP,» de *Eleventh LACCEI Latin American and Caribbean Conference for Engineering and Technology* , Cancun, Mexico , Agosto 14-16 2013.

[23] A. Olivera, «Heurísticas para Problemas de Ruteo de Vehículos,» Instituto de Computación, Facultad de Ingeniería, Universidad de la República, Montevideo, Uruguay., 2004.

[24] J. R. Montoya-Torres, J. López Franco, S. Nieto Isaza, H. Felizzola Jiménez y N. Herazo-Padilla, «A literature review on the vehicle routing problem with multiple depots,» *Computers & Industrial Engineering,* vol. 79, pp. 115-129, 2015.

[25] S. N. I. Julian López Franco, «Heuristica para la generación de un conjunto de referencia MDVRP de Soluciones que Resuelvan el Problema de Ruteo de Vehículos con Múltiples Depósitos MDVRP,» *Tenth LACCEI Latin American and Caribbean Conference (LACCEI'2012), Megaprojects: Building Infrastructure by fostering*, 2012.

[26] J.-F. C. G. L. Benoit Crevier, «The multi-depot vehicle routing problem with inter-depot routes,» *European Journal of Operational Research,* nº 176, p. 756–773, 2007.

[27] Y. C. K.C. Tan, «A hybrid multi-objective evolutionary algorithm for solving truck and trailer vehicle routing problems,» *European Journal of Operational Research*, nº 172, p. 855–885, 2005.

[28] G. Laporte y Y. Nobert, Survey Combinatorial Optimization.

[29] J. L. F. N. H. P. Santiago Nieto Isaza, «Desarrollo y codificacion de un Modelo Matematico para la Optimizacion del MDVRP,» de *LACCEI*, Panama, 2012.

[30] F. Glover y G. A. Kochenberger, Handbook of Metaheuritics, Kluwer Academic Publishers, 2003.

[31] J. Lysgaard, «Clarke & Wright's Savings Algorithm».

[32] Mole y Jameson, «A Sequential Route-Building Algorithm Employing a Generalised Savings Criterion».

[33] Solomon, «Algorithms for the vehicle routing problem with time windows constraints».

[34] L. T. 0. V. D. Giosa, «New assignment algorithms for the multi-depot vehicle routing problem,» *Journal of the Operational Research Society,* vol. 53, nº 9, pp. 977-984, 2002.

[35] O. V. L. Tansini, «New measures of proximity for the assignment algorithms in the MDVRPTW,» *Journal of Operational Reserch Society,* vol. 57, nº 3, pp. 241-249, 2006.

[36] J. Renaudl, G. Laporte y F. F. Boctor, «A tabu search heuristics for the multi-depot vehicle routing problem,» *Computers & Operations Research,* vol. 23, nº 3, pp. 229-235, 1996.

[37] S. Lin, Computer Solution of the traveling Salesman Problem, 1965.

[38] S. Lin, «Computer solutions of the traveling salesman problem,» *Bell System Technical Journal 44,* pp. 2245-2269, 1965.

[39] J. Chicano Garcia, «Tecnicas de Optimizacion Modernas para problemas Complejo,» Úniversidad de Málaga, Málaga, 2003.

[40] I. Or, Traveling salesman-type combinatorial optimization problems and theeir relation to the logistics of regional blood banking, 1976.

[41] A. V. Breedam, «Improvement heuristics for the Vehicle Routing Problem based on Simulated Annealing,» *European Journal of Operational Research 86,* pp. 480-490, 1995.

[42] M. Gendreau, A. Hertz y G. Laporte, «New Insertion and Postoptimization procedures for the traveling Salesman Problem,» *Operation Research,* vol. 40, nº 6, pp. 1086-1094, 1992.

[43] «OR/MS Today, February 2014,» feb 2014. [En línea]. Available: http://www.orms-today.org/surveys/Vehicle_Routing/vrss.html.