**UNIVERSIDAD DE LA REPÚBLICA**

Facultad de Ingeniería

Instituto de Computación

“Periodic VRP”

Introducción Estado del Arte de Proyecto de Grado

Javier de Prado

Alejandro García

Francisco Güella

Tutores

Sandro Moscatelli

Omar Viera

Índice General

[Introducción 4](#_Toc395113926)

[Definición del problema 6](#_Toc395113927)

[Métodos y soluciones de VRP 7](#_Toc395113928)

[Modelos exactos 7](#_Toc395113929)

[Métodos aproximados 9](#_Toc395113930)

[Heurísticas clásicas. 9](#_Toc395113931)

[Metaheuristicas. 10](#_Toc395113932)

[Problemas MDVRP 11](#_Toc395113933)

[Algoritmos de Asignación para MDVRP 11](#_Toc395113934)

[Asignación a través de urgencias 11](#_Toc395113935)

[Asignación paralela 11](#_Toc395113936)

[Asignación simplificada 12](#_Toc395113937)

[Barrido de asignación 12](#_Toc395113938)

[Asignación Cíclica 12](#_Toc395113939)

[Algoritmos de Ruteo para MDVRP 13](#_Toc395113940)

[Método de Clarke and Wright 13](#_Toc395113941)

[Software para el problema de VRP. 14](#_Toc395113942)

# Introducción

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

“Problema del Agente Viajero” o problema del viajante: 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?

En inglés se lo conoce como TSP (Travelling Salesman Problem).

Es el problema de encontrar un ciclo simple que visite todos los nodos del problema y cuyo costo total sea mínimo.

Dicho problema lo estudiaron ya en 1856 Kirkman y Hamilton. Y luego Kowalewsky en 1917. No está claro cuándo fue que se trató el problema matemáticamente por primera vez [1].

Con dos ciudades, encontrar el mejor camino es sencillo. Hay únicamente dos opciones, primero visitamos la ciudad A y luego la ciudad B, o primero visitamos la ciudad B y luego la ciudad A. (2!/2) Si tuviésemos tres ciudades las posibilidades aumentan a 6!/2, primero A luego B luego C, ó primero A luego C luego B, ó primero B luego A…. etc.

Si tenemos n ciudades, asumiendo que cada par de ciudades está unida por un camino, la cantidad de posibilidades (rutas) son n!/2 [2]. Por ejemplo 10 ciudades es 3628800/2. El problema TSP es un problema NP-Duro, demostrado por Richard Karp en 1972 [3] Por lo tanto el uso de de métodos exactos de optimización para resolver este problema nos darían tiempos de procesamiento de CPU no aceptables.

Al problema de determinar rutas de vehículos para darle servicio a un conjunto de clientes, los cuales tiene una determinada demanda, y los vehículos tienen una determinada capacidad, se lo conoce como un  VRP (*Vehicle Routing Problem)*. Problema de ruteo de vehículos. En 1959 Dantzig y Ramser [2] realizaron una formulación del problema VRP. Consistía en buscar la forma óptima de distribuir combustible entre varios clientes. Dicha formulación se presenta como una generalización del problema TSP en el cual se toman en cuenta las capacidades de carga de los vehículos y la demanda de los clientes.

Si además se agrega la restricción de que los clientes solo pueden ser servidos en algún período de tiempo específico, al problema se lo conoce como VRPTW (*Vehicle Routing Problem with time windows*). [4] [5]

En estos problemas tenemos un único depósito desde el cual inician las rutas que atienden los clientes. Una variante a estos problemas es tener múltiples depósitos. Por lo cual un cliente puede ser atendido por cualquiera de los depósitos. A este problema se lo conoce como (MDVRPTW) (*Multi depot* *Vehicle Routing Problem with Time Windows).*  [6]

Estos son solo algunos de los tipos de problema VRP que se han formulado.

VRP, VRPTW, MDVRPTW al igual que su antecesor TSP también son problemas NP-completos [4]. O sea que para encontrar la solución óptima, la cantidad de posibilidades son tantas que serían cálculos demasiado grandes por lo cual el tiempo de CPU sería inaceptable.

           De esto surge la necesidad de buscar heurísticas y metaheurísticas que puedan dar buenas soluciones en tiempos razonables.

Por ejemplo, el proceso de transporte de mercancías se encuentra presente en muchos de los sistemas de producción, representando una parte importante (entre el 10% y el 20%) del coste final del producto. Así, la utilización de este tipo de problemas o procedimientos ha dado lugar a un ahorro de entre el 5% y el 20% en el coste total de transporte. El interés en VRP ha sido motivado por su relevancia práctica y su considerable dificultad. Decenas de algoritmos y métodos han sido propuestos para la búsqueda de buenas soluciones [7].

En [7], se modela el problema general de VRP mencionando las siguientes características sobre clientes, vehículos y objetivos del problema:

Los clientes tienen una demanda, podrían tener también:

* Una ventana de tiempo,
* Tiempos de carga y de descarga,
* Restricciones en cuanto al tipo de vehículo que puede darle servicio.

Los vehículos podrían tener las siguientes características:

* Que tengan un depósito hogar
* Una capacidad determinada
* Posibles divisiones de compartimento que determinan que tipos de bienes se pueden transportar
* Costos, por distancia, tiempo, ruta, etc

Los objetivos del problema podrían ser:

* Minimización del costo global de transporte.
* Minimización de la flota de vehículos
* Balanceo de rutas para el tiempo y la carga
* Minimización de penalizaciones asociadas con el servicio parcial a clientes o cualquier combinación de dichos objetivos.

Para la variante de multi-depósito al problema VRP (MDVRP), encontramos la siguiente formulación del problema en [6]. En el MDVRPTW, los clientes pueden ser servidos por uno de varios depósitos. Como en VRP, cada vehículo debe partir y volver al mismo depósito y el tamaño de flota de cada depósito debe estar en el rango especificado entre un mínimo y un máximo. El caso de MDVRPTW puede ser visto como un problema de clustering en el sentido de que la solución es un conjunto de vehículos distribuidos en cada depósito.

## Definición del problema

Según el Articulo “New Assignment Algorithms for the Multi-Depot Vehicle Routing Problem” [6] , el problema de MDVRPTW consiste en determinar un conjunto de rutas de vehículos de forma que:

1. Cada ruta comienza y finaliza en el mismo depósito.
2. Los requerimientos de un cliente son servidos por un vehículo.
3. Las ventanas de tiempo de ambos, del cliente y del depósito son respetadas
4. La suma de los requerimientos servidos no excede la capacidad de cada vehículo
5. El costo total está minimizado

Se ha propuesto resolver el problema MDVRPTW en dos fases. Una primera fase de asignación de clientes a depósitos, y una segunda fase para el rutas de los vehículos a los clientes.

Idealmente, mejores resultados serían obtenidos tratando las dos fases simultáneamente, pero con problemas de más de 100 clientes, estos algoritmos simultáneos no son tratables computacionalmente [6].

Otra variante del problema VRP es FSMVRP [5]. En este caso FSMVRP es una extensión del VRP en la que se elimina la hipótesis de que los vehículos son idénticos. En particular los costos y las capacidades de los vehículos varían.

Otra variante más es VRPMT. Vehicle routing problem with multiple trips. Esta es una variante en la cual se dispone de una cantidad fija de vehículos y cada uno de estos puede recorrer más de una ruta en el mismo período de planificación [5].

# Métodos y soluciones de VRP

Desde que el problema de ruteo de Vehículos fue formulado por primera vez, numerosos métodos han sido propuestos para su resolución. Los mismos pueden separarse en algoritmos exactos y algoritmos aproximados. Los algoritmos exactos encuentran la solución óptima, pero al ser VRP un problema NP-HARD suelen requerir tiempos de ejecución muy altos, siendo impracticables para la mayoría de los casos de la vida real. Los métodos aproximados como heurísticas y  metaheurísticas devuelven una solución aproximada en tiempo de ejecución mucho menor.

Describiremos algunos de los algoritmos exactos, heurísticas y metaheuristica a modo de introducir una idea general sobre su funcionamiento, sin entrar en los detalles de los mismos.

## Modelos exactos

Come se dijo anteriormente los métodos exactos obtienen el resultado óptimo luego de su terminación, siendo normalmente el tiempo de ejecución del algoritmo prohibitivo para casos de la práctica. Aunque es importante destacar que en algunos casos se obtienen buenas aproximaciones en pasos intermedios de la ejecución. (buscar ref)

La mayoría de los métodos exactos pueden clasificarse en 3 categorías, Búsqueda arborescente, Programación Dinámica, Generación de columnas (referencia de la maestría agregar información de c/u)

Procederemos a describir brevemente uno de los métodos exactos más utilizados de branch and bound, el método k-tree. Dicho método [8] es uno de los más conocidos para la resolución del CVRP y fue exitoso en resolver casos con 71 clientes, sin embargo hay instancias menores que no fueron resueltas.

El enfoque de K-tree puede ser extendido para casos reales con costos asimétricos, ventanas de tiempo y flotas de vehículos no uniformes.

Explicaremos conceptualmente un algoritmo branch and bound, el cual utiliza la estrategia de divide y vencerás. En el mismo se examina el espacio entero de soluciones S, esta es la etapa de “Bouding” y se hace una relajación del problema. Resolviendo esta relajación del problema se obtiene un elemento de S o la cota inferior de la solución óptima. Si se obtiene un elemento de S entonces el problema está resuelto.

En otro caso, si no se obtiene un elemento de S, se divide S en n subconjuntos (subproblemas), Estos son un conjunto de sub-problemas candidatos esperando para ser procesados. Esta etapa se conoce como branching, luego se procesa cada uno de los subproblemas candidatos detectando 4 posibles situaciones.

* Se encuentra una solución mejor que la cota inferior, se remplaza la cota inferior y se continúa.
* No se encuentra solución al subproblema (se descarta, pune).
* Se compara la cota inferior encontrada, con la cota superior (mejor solución encontrada hasta el momento), si es mayor o igual entonces se descarta.
* Si no se pudo descartar entonces se realiza un branch del subproblema y los subproblemas hijos son agregados a la lista de candidatos activos. Continuamos de esta manera hasta que la lista de candidatos sea vacía, en tal caso nuestra mejor solución es la solución óptima.

## Métodos aproximados

Dados los elevados tiempos necesarios para ejecutar los métodos exactos, es necesario modelos más prácticos como el heurístico para la resolución del problema. Dichos métodos se basan en un análisis de cómo encontrar una buena solución. Este análisis puede ser basado en experiencia e intuición del creador, o incluso en sistemas naturales con un comportamiento similar. A través de heurísticas se puede llegar a una buena solución, pero no se tienen garantías de que sea la solución óptima. En referencia a los tiempos de ejecución, son considerablemente menores y en algunos casos permite ejecutar el problema con miles de nodos.

Para el problema particular de VRP se han propuesto heurísticas clásicas y metaheurísticas. Las heurísticas clásicas realizan una exploración muy limitada del espacio de soluciones. En general son problemas constructivos de una pasada o métodos simples de búsqueda local por lo que los tiempos de ejecución son relativamente bajos. Las metaheurísticas buscan hacer una exploración más detallada sobre el espacio de soluciones. Estos tipos de algoritmos además de métodos constructivos y búsqueda local agregan el uso de estrategias de combinación de soluciones a modo de poder ir mejorando las soluciones en cada iteración y comparándola con las demás soluciones posibles. (poner referencia de marestria).

## Heurísticas clásicas.

Inicialmente veremos la base de los métodos constructivos en los cuales se crea una solución de forma gradual. Por lo general se trabaja sobre un conjunto parcial que es aumentado en cada iteración según una regla valida.

El algoritmo de ahorro de Clark and Wrigth [9] es de los algoritmos heurísticos para VRP más conocidos. Dicho método aplica a problemas donde el número de vehículos no es fijo (es una variable de decisión) y además funciona correctamente en problemas de direccionamiento directo e indirecto. Se considera un algoritmo heurístico constructivo.

Consiste principalmente en analizar el costo ahorrado de juntar rutas y consta de 3 fases. La primera fase consta de la inicialización y cálculos previos de costos ahorrados para el nodo x. Luego en la segunda fase se mejora la secuencia, se ejecuta en paralelo y se calcula el camino más corto con los nodos adyacentes a x. La última fase consiste en una ejecución secuencial para unir rutas.

Otras heurísticas se clasifican como los métodos de dos fases. Consisten en descomponer el problema en dos componentes naturales del mismo, hacer la agrupación de vehículos en posibles rutas (Clúster) y la construcción de dichas rutas (routes). Permitiendo en caso de ser necesario realimentación entre ambas fases. Existen métodos donde se ejecuta primero la fase de construcción de rutas para luego ejecutar la agrupación de vehículos a rutas y viceversa (buscar referencias de método de dos fases).

Hay varios algoritmos de dos fases para soluciones de VRP como por ejemplo Algoritmo de Pétalos (Referencia), Algoritmo de Fisher and Jaikumar [10] el cual utiliza GAP (Generalized Assignment Problem) para formar el clúster. Talliard’s [11], este se basa en definir los vecinos utilizando el mecanismo de generación λ-interchange [12]. (Esplicar métodos debajo)

FRAN, FIJATE QUE ACÁ PONGO ALGO DE POSTOPTIMIZACIÓN PARA EL CASO DE DOS FASES:

En [7]( «The Vehicule Routing Problem,» Toth\_Vigo ) se muestran procedimientos para mejorar las rutas simples. La mayoría de ellos se puden describir en térmios del mecanismo de ʎ-opt, Lin’s[poner Lins].

[]Lin’s define ʎ-opt de un ruta de la siguiente forma:

Definición: Una ruta es ʎ -opt si es imposible obtener una ruta con menor costo por reemplazar ʎ de sus arcos por cualquier otro conjunto de ʎ arcos.

El chequeo de la ʎ-optimalidad de una solución puede realizarse en tiempo O(nʎ).

Toth\_Vigo[7] nos muestra que varias modificaciones a este esquema de mejoras de rutas han sido propuestas. Muchas de las heurísticas del tipo de dos fases, pueden incluir alguna forma de reoptimización de este estilo en pasos intermedios. Algunas modifican el ʎ dinamicamente durante la búsqueda, otras consisten en desplazar cadenas de 3,2 o 1 vertices consecutivos a otra localización.

En este aspecto el algorimto de ahorro de Clark and Wrigth es distinto ya que este es de tipo constructivo.

## Metaheuristicas.

Para llegar a una mejor solución que las heurísticas clásicas, es necesario aplicar técnicas que exploren de una forma más eficiente el espacio de soluciones del problema. Estas técnicas son procedimientos genéricos para problemas de optimización y búsqueda sobre el espacio de soluciones posibles. En general las metahuristicas proponen mejores soluciones que las heurísticas pero requieren mayor uso computacional y tiempo de ejecución. Siendo estos tiempos mucho menores que los algoritmos exactos que exploran  todas las soluciones. Así mismo las metahuristicas pueden combinar técnicas generando algoritmos híbridos.

Un ejemplo de Método es el de **Búsqueda local** que se utiliza en las Heurísticas. Gran cantidad de técnicas para VRP utilizan la búsqueda local como base, por lo que es importante destacarla. La búsqueda local en su caso básico, busca definir para cada solución s un conjunto de soluciones N(s) llamado conjunto de soluciones vecinas. Definimos una operación movida la cual transforma una solución a otra solución de la vecindad. En dicho algoritmo se considera una solución inicial al problema. Luego se itera y en cada iteración se busca una solución de menor costo en la vecindad, si no hay ninguna se llegó a un mínimo local respecto a la vecindad. En otro caso se aplica una movida y se continúa iterando. (poner referencia de marestria).

Sobre dicho algoritmo de búsqueda línea básica, se generaron nuevas técnicas que entran en el campo de las metaheuristica, Por ejemplo Simulated Annealing que está dotada de un mecanismo para escapar de mínimos locales; fue propuesta por Kirkpatrick 1983 [13]. Tabu Search  fue propuesta por Glover [14] y acepta movidas que deterioran la solución para llegar a una solución final mejor.

Luego podemos agregar Variable Neighborhood Search (VNS) [15] la cual explota el hecho de que la optimalidad local es un concepto que depende de la definición de la vecindad; para continuar la búsqueda luego de encontrar un óptimo local. La Metauristica Greedy Randomized Adaptive Search Procedure (GRASP) [16]la cual consiste de dos fases; construcción de una solución inicial y luego búsqueda local.  También hay otras metaheuristicas como Algoritmos Genéticos, colonias de hormigas etc.

# Métodos y soluciones de VRSPTW

Una de las variantes de VRP incluye restricciones temporales en la construcción de las rutas (VRPTW).En [poner Salomon] se presenta cómo VRSPTW (Vehicle routing and scheduling problema with time Windows). VRSPTW emergió como una importante área para el progreso del tratamiento más realista de las complicaciones y generalizaciones del modelo básico de ruteo. Las restricciones de ventana de tiempo surgen naturalmente en los negocios que trabajan con distribuciones. Algunos ejemplos incluyen repartos en los bancos, reparto del correo, buses escolares etc.

[Salomon] Presenta los siguientes conceptos, notaciones y supuestos para la explicación de las heurísticas utilizadas para los casos de restricciones temporales:

Asume una flota de autos homogénea.

El servicio a un cliente *i*, *i*=1,..*n* durante si unidades de tiempo, puede comenzar en un tiempo bi, con una ventana de tiempo definida por ei (earliest) and li(latest) en la cual el cliente *i* va a permitir el inicio del servicio.

Por lo tanto si un vehículo viaja directamente desde el cliente *i* al cliente *j* y arriba demasiado temprano a *j*, deberá esperar, esto es: bj=max{ej,bi+si+tij} donde tij es el tiempo del viaje directo de i a j. Notar que los tiempo bi son variables de decisión.

Se asume que el costo del viaje directo desde *i* a *j* está dado por cij =α1dij+ α2(bj-bi) α1, α2>=0.  dij es la distancia directa de *i* a *j*. Por ejemplo, si α1=0yα2=1, lo que se busca es minimizar el tiempo total de distribución.

Se asume también que el número de vehículos usados es libre. Los vehículos dejan el depósito denotado por nodo 0, en el tiempo e0, como más temprano, y deben retornar al depósito en tiempo l0 a más tardar. Notar que el tiempo de partida desde el depósito es una variable de decisión.

Algunas de las clases de heurísticas mencionadas por salomon[] son:

-**Savings Heurístics**: Son una extension de Calrke and Wright .

-**A time oriented , Nearest-Neighbor Heuristics**. Son métodos secuenciales para la construcción de la ruta en la cual en cada iteración se busca al cliente no servido más cercano. En esta aproximación se toma en cuenta tanto la cercanía geográfica como temporal.

-**Insertion Heuristics,** Luego de inicializar cada ruta, estos métodos usan dos criterios en cada iteración para insertar un nuevo cliente *u* en la ruta en construcción entre dos clientes adyacente *i* y *j*.

-**A Time-Oriented Sweep Heuristic.** Estos métodos cuentan con una fase de clustering y otra fase de shceduling.

# Problemas MDVRP

## Algoritmos de Asignación para MDVRP

En [6] se enfoca principalmente en la asignación (agrupamiento) de los clientes a los almacenes mediante las siguientes heurísticas:

### Asignación a través de urgencias

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

### Asignación paralela

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

Se calcula como:

donde es la distancia entre el cliente y el depósito , es el conjunto de depósitos y es la distancia entre el cliente y el depósito más cercano . El cliente con mayor valor de será asignado el 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.

### Asignación simplificada

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

donde es la distancia entre el cliente y el segundo depósito más cercano y es la distancia entre el cliente y el depósito más cercano . El cliente con mayor valor de será asignado el depósito más cercano.

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

### 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 \* con mayor demanda insatisfecha, la evaluación de la urgencia se realiza con el siguiente cálculo:

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.

### Asignación Cíclica

El procedimiento consiste en asignar de una manera cíclica, un cliente a la vez. La heurística asigna el cliente más cercano al último asignado, para el mismo depósito como este último.

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.

## Algoritmos de Ruteo para VRP

La asignación de rutas se realiza mediante el siguiente algoritmo:

Método de Clarke and Wright

En 1964 Clarke & Wright publicaron un algoritmo para la solución de ese tipo de problemas de enrutamiento [9]. Este algoritmo se basa en un concepto llamado ahorro. A continuación describiremos el mismo:

1. Inicialización. Cada uno de los vehículos atiende a un cliente.
2. Para todos los clientes con calcular los ahorros para la unión de los ciclos usando el par
3. Ordenar los ahorros en forma descendente.
4. Tomar el par del tope de la lista de ahorros, unir dos ciclos separados con el par
5. Los nodos pertenecen a ciclos separados
6. No se supere la capacidad máxima del vehículo
7. y son primero o el último cliente en sus ciclos.
8. Repetir paso 4 hasta que la lista de ahorros se termina o las capacidades no permiten más fusión.

## Mencionar Solomon, Potvin & Rousseau, la de dullaert para el caso de flotas heterogeneas

## Algoritmos de Post optimización para MDVRP

Trabajar ¡!!

# Software para el problema de VRP.

A partir de OR/MS Today, Febrery 2014 se provee la información de los programas existentes para el problema de enrutamiento de vehículos. 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. Por una información más detallada se puede visitar el sitio web de los mismos. [17]

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Platforms Supported** | | | | |
| **Windows** | **iOS** | **Android** | **Web-based software as a Service (SaaS)** | **Other (Specify)** |
| ArcGIS for TransportationAnalytics | Esri | 2012 | y | y | y | y |  |
| Descartes Routing, Mobile & Telematics | Descartes | 1995 | y |  | y | y |  |
| DISC | MJC2 | 1990 | y | y | y | y | UNIX & Linux |
| eRoute Logistics | WM Logistics | 2002 | y |  | y | y |  |
| JOpt.SDK | DNA Evolutions GmbH | 2006 | y |  |  | y |  |
| Logistics Optimizer | Runzheimer International | 2005 |  |  |  | y |  |
| logvrp | Netakil | 2010 | y | y | y | y | REST + SOAP Web Service API |
| Optrak vehicle routing software | Optrak Distribution Software Ltd | 1992 | y |  |  | y |  |
| ORTEC Routing and Dispatch | ORTECÂ | 2003 | y | y | y | y |  |
| Paragon Routing and Scheduling Optimizer | Paragon Software Systems, Inc | 1997 | y |  |  | y |  |
| Roadnet Transportation Suite | Roadnet Technologies | 1983 | y | y | y | y |  |
| Routist.com | KKT srl | 2013 |  |  |  | y |  |
| StreetSync Basic | RouteSolutions | 2008 | y |  |  | y |  |
| StreetSync Desktop | RouteSolutions | 2005 | y |  |  |  |  |
| StreetSync Pro | RouteSolutions | 2011 | y |  |  | y |  |
| TMW Appian DirectRoute | TMW Systems | 1996 | y |  |  | y |  |
| TruckStops | MapMechanics | 1983 | y |  |  |  | Server or PC |

Bibliografía

|  |  |
| --- | --- |
| [1] | A. Schrijver, «On the history of combinatorial optimization,» [En línea]. Available: http://homepages.cwi.nl/~lex/files/histco.pdf. [Último acceso: 03 Agosto 2014]. |
| [2] | D. G. B. Ramser J. H., «The Truck Dispatching Problem,» [En línea]. Available: http://andresjaquep.files.wordpress.com/2008/10/2627477-clasico-dantzig.pdf. [Último acceso: 03 Agosto 2014]. |
| [3] | R. M. Karp, «Reducibility Among Combinatorial Problemas,» 1971. [En línea]. Available: http://www.seas.upenn.edu/~bhusnur4/cit596\_spring2014/karp-1971.pdf. [Último acceso: 03 Agosto 2014]. |
| [4] | S. N. Kumar y R. Panneerselvam, «A Survey on the Vehicle Routing Problem and Its Variants,» [En línea]. Available: http://dx.doi.org/10.4236/iim.2012.43010. [Último acceso: 03 Agosto 2014]. |
| [5] | A. Olivera, «Heurísticas para Problemas de Ruteo de Vehículos,» [En línea]. Available: https://www.fing.edu.uy/inco/pedeciba/bibliote/reptec/TR0408.pdf. [Último acceso: 03 Agosto 2014]. |
| [6] | D. Giosa, L. Tansini y O. Viera, «New Assignment Algorithms for the Multi-Depot Vehicle Routing Problem». |
| [7] | P. Toth y D. Vigo, «The Vehicule Routing Problem,» 2001. [En línea]. Available: http://www.dim.uchile.cl/~tcapelle/BIBLIOGRAFIA%20TESIS/Toth\_Vigo\_-\_The\_vehicle\_routing\_problem.pdf. [Último acceso: 03 Agosto 2014]. |
| [8] | M. L. Fisher, «Optimal Solution of Vehicle Routing Problems Using Minimum K-trees,» *Operations Research,* nº 42, pp. 626-642, 1994. |
| [9] | G. Clarke y J. W. Wright, «Scheduling of Vehicles from a Central Depot to a Number of Delivery Points,» 1964, pp. 568-581. |
| [10] | J. R. Fisher M., 1981. [En línea]. Available: http://neo.lcc.uma.es/vrp/bibliography-on-vrp/#FisherJaikumar81. [Último acceso: 03 Agosto 2014]. |
| [11] | E. D. Taillard, 1993. [En línea]. Available: http://neo.lcc.uma.es/vrp/bibliography-on-vrp/#Taillard93. [Último acceso: 03 Agosto 2014]. |
| [12] | I. Osman, 1993. [En línea]. Available: http://neo.lcc.uma.es/vrp/bibliography-on-vrp/#Osman93. [Último acceso: 03 Agosto 2014]. |
| [13] | S. Kirkpatrick, C. D. Gelatt y M. P. Vecchi, «Optimization by Simulated Annealing,» *Science, New Series,* vol. 220, nº 4598, pp. 671-680, (May 13, 1983. |
| [14] | «Future Paths for Integer Programming and Links to Artificial Intelligence,» *Computers and Operations Research,* vol. 13, pp. 533-549, 1986. |
| [15] | N. M. V. N. S. P. Hansen, En Handbook of Metaheuristics 145–184, Kluwer Academic Publishers, 2003. |
| [16] | G. K. (. F. Glover, Chapter 8, Handbook of Metaheuristics, 221-249, Kluwer Academics, 2003, p. http://sci2s.ugr.es/docencia/metaheuristicas/GRASP.pdf. |
| [17] | «OR/MS Today, February 2014,» feb 2014. [En línea]. Available: http://www.orms-today.org/surveys/Vehicle\_Routing/vrss.html. |
| [18] | W. W. Clarke G., 1964. [En línea]. Available: http://neo.lcc.uma.es/vrp/bibliography-on-vrp/#ClarkeWright64. [Último acceso: 03 Agosto 2014]. |