**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](#_Toc394865542)

[Definición del problema 6](#_Toc394865543)

[Métodos y soluciones de VRP 7](#_Toc394865544)

[Modelos exactos 7](#_Toc394865545)

[Métodos aproximados 9](#_Toc394865546)

[Heurísticas clásicas. 9](#_Toc394865547)

[Metaheuristicas. 10](#_Toc394865548)

[Bibliografía 12](#_Toc394865549)

# 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. Encontrar la ruta óptima ya es más difícil.  100 ciudades… demasiado, incluso para una computadora. El problema TSP es un problema NP-Duro, demostrado por Richard Karp en 1972 [karp.pdf] 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*).[3,4]

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).*  [TANSINI.pdf]

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 [3]. 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 [5].

En [5], 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 con las siguientes posibles 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 [TANSINI.PDF]. 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

El 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

[TANSINI.PDF]

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.[TANSINI.PDF]

Otra variante del problema VRP es FSMVRP [4] . 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 [4].

# 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 métodos para algoritmos exactos, heuristicas 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 de un modelo exacto, aproximándose este a una heurística.

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

Procederemos a describir brevemente uno de los métodos exactos más utilizados de branch and bound, el método k-tree. Dicho método [] 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 meta heurísticas, 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 meta heurísticas 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.

## 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 aumentada en cada iteración según una regla valida.

El algoritmo de ahorro de Clark and Wrigth [6] 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 de funcionar correctamente en problemas de direccionamiento directo e indirecto y se considera un algoritmo heurístico constructivo.

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

Otras heurísticas se clasifican como los métodos de dos fases, consiste en descomponer 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 la agrupación de vehículos a rutas y viceversa.

Hay varios algoritmos de dos fases para este tipo de soluciones de VRP como por ejemplo Algoritmo de Pétalos, Algoritmo de Fisher and Jaikumar [7] el cual utiliza GAP para formar el clúster. Talliard’s [8], este se base en definir los vecinos utilizando el mecanismo de generación λ-interchange [9].

## 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. Dependiendo del contexto y las técnicas se construye un algoritmo de la solución. En general las metauristicas 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 Técnica es la **Búsqueda local**, muchas de las técnicas para VRP utilizan la búsqueda local como base por lo que es importante destacar su importancia. En su caso básico busca definir para cada solución s un conjunto de soluciones N(s) llamado conjunto de soluciones vecinas. Considerando 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 coste 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.

Existen Otras técnicas como Simulated Annealing que está dotada de un mecanismo para escapar de mínimos locales; fue propuesta por Kirkpatrick [84]. Tabu Search  fue propuesta por Glover [66], y acepta movidas que deterioran la solución para llegar a una solución final mejor.

Luego podemos agregar Variable Neighborhood Search (VNS) [98, 78] 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) la cual consiste de dos fases; construcción de una solución inicial y luego búsqueda local.  También hay otras metaheuristicas como Algoritmos Geneticos, colonias de hormigas etc.

# Bibliografía

1. Alexander Schrijver, “On the history of combinatorial optimization”.

<http://homepages.cwi.nl/~lex/files/histco.pdf.>

Consultado 03/08/2014.

1. G. B. Dantzig and J. H. Ramser, “The Truck Dispatching Problem”.

<http://andresjaquep.files.wordpress.com/2008/10/2627477-clasico-dantzig.pdf> Consultado 03/08/2014.

1. Suresh Nanda Kumar and Ramasamy Panneerselvam, “A Survey on the Vehicle Routing Problem and Its Variants”. <http://dx.doi.org/10.4236/iim.2012.43010>. Consultado 03/08/2014.
2. Alfredo Olivera, “Heurísticas para Problemas de Ruteo de Vehículos”.

<https://www.fing.edu.uy/inco/pedeciba/bibliote/reptec/TR0408.pdf>. Consultado 03/08/2014.

1. Paolo Toth and Danielle Vigo, “The Vehicule Routing Problem”. 2001.

<http://www.dim.uchile.cl/~tcapelle/BIBLIOGRAFIA%20TESIS/Toth_Vigo_-_The_vehicle_routing_problem.pdf>.

Consultado 03/08/2014.

1. Clarke and Wright. 1964.

[http://neo.lcc.uma.es/vrp/bibliography-on-vrp/#ClarkeWright64](http://neo.lcc.uma.es/vrp/bibliography-on-vrp/%23ClarkeWright64)

Consultado 03/08/2014.

1. [Fisher and Jaikumar. 1981](http://neo.lcc.uma.es/vrp/bibliography-on-vrp/#FisherJaikumar81).

[http://neo.lcc.uma.es/vrp/bibliography-on-vrp/#FisherJaikumar81](http://neo.lcc.uma.es/vrp/bibliography-on-vrp/" \l "FisherJaikumar81).

Consultado 03/08/2014.

1. Talliard. 1993.

<http://neo.lcc.uma.es/vrp/bibliography-on-vrp/#Taillard93>

Consultado 03/08/2014.

1. Osman 1993

[http://neo.lcc.uma.es/vrp/bibliography-on-vrp/#Osman93](http://neo.lcc.uma.es/vrp/bibliography-on-vrp/%23Osman93)

Consultado 03/08/2014.