

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

FACULTAD DE INGENIERÍA MECÁNICA Y ELÉCTRICA

DIVISIÓN DE ESTUDIOS DE POSGRADO



ALGORITMO HÍBRIDO BASADO EN UN MÉTODO DE  
APROXIMACIONES SUCESIVAS PARA EL PROBLEMA DE RUTEO DE  
VEHÍCULOS HETEROGÉNEO

POR

TATIANA QUINTERO QUINTERO

EN OPCIÓN AL GRADO DE

MAESTRÍA EN CIENCIAS

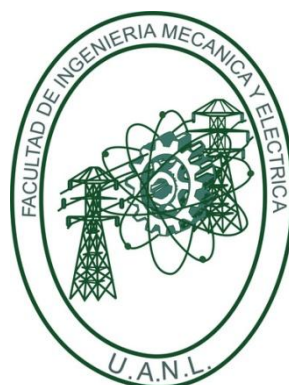
EN INGENIERÍA DE SISTEMAS

SAN NICOLÁS DE LOS GARZA, NUEVO LEÓN, JUNIO DEL 2012

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

FACULTAD DE INGENIERÍA MECÁNICA Y ELÉCTRICA

DIVISIÓN DE ESTUDIOS DE POSGRADO



ALGORITMO HÍBRIDO BASADO EN UN MÉTODO DE  
APROXIMACIONES SUCESIVAS PARA EL PROBLEMA DE RUTEO DE  
VEHÍCULOS HETEROGÉNEO

POR

TATIANA QUINTERO QUINTERO

EN OPCIÓN AL GRADO DE  
MAESTRÍA EN CIENCIAS  
EN INGENIERÍA DE SISTEMAS

SAN NICOLÁS DE LOS GARZA, NUEVO LEÓN, JUNIO DEL 2012

Universidad Autónoma de Nuevo León

Facultad de Ingeniería Mecánica y Eléctrica

División de Estudios de Posgrado

Los miembros del Comité de Tesis recomendamos que la Tesis «Algoritmo híbrido basado en un método de aproximaciones sucesivas para el problema de ruteo de vehículos heterogéneo», realizada por la alumna Tatiana Quintero Quintero, con número de matrícula 1541921, sea aceptada para su defensa como opción al grado de Maestría en Ciencias en Ingeniería de Sistemas.

El Comité de Tesis

---

Dra. Sara Verónica Rodríguez Sánchez

Asesor

---

Dra. Ada M. Álvarez Socarrás

Revisor

---

Dr. Fco. Javier Faulin Fajardo

Revisor

Vo. Bo.

---

Dr. Moisés Hinojosa Rivera

División de Estudios de Posgrado

San Nicolás de los Garza, Nuevo León, junio del 2012

*A mis padres,  
porque siempre me ha apoyado en cada decisión que he tomado.*

*A mis hermanas,  
porque con perseverancia se logran grandes cosas.*

*A mi amore,  
porque ningún sueño es inalcanzable, TE AMO!*

# ÍNDICE GENERAL

<b>Agradecimientos .....</b>	<b>XI</b>
<b>Resumen .....</b>	<b>XIII</b>
<b>Introducción.....</b>	<b>1</b>
1.1 Descripción del problema .....	1
1.2 Hipótesis .....	2
1.3 Justificación.....	2
1.4 Objetivos .....	3
1.5 Estructura de la tesis .....	3
<b>Antecedentes.....</b>	<b>5</b>
2.1. El problema de ruteo de vehículos.....	5
2.1.1 Algunas variantes del problema de ruteo de vehículos .....	7
2.2 Métodos de solución para el problema de ruteo de vehículos.....	11
2.2.1 Algoritmos exactos .....	11
2.2.2 Métodos heurísticos.....	11
2.2.3 Metaheurísticas.....	18
2.3 Simulación de Monte Carlo.....	20
2.3.1 Antecedentes de la simulación .....	20
2.3.2 Las técnicas de Monte Carlo.....	23
2.3.3 Aplicación de la simulación de Monte Carlo al problema de ruteo de vehículos. ....	23
2.4 Aportaciones en literatura al problema de ruteo de vehículos heterogéneo.....	27

<b>Descripción y modelación del problema de ruteo de vehículos heterogéneo</b>	<b>35</b>
3.1 Parámetros y conjuntos.....	36
3.2 Variables del problema .....	37
3.2.1 Variables de decisión .....	37
3.2.2 Variables auxiliares .....	37
3.3 Objetivo del problema .....	38
3.4 Restricciones.....	38
<b>Metodología propuesta para el problema de ruteo de vehículos heterogéneo.....</b>	<b>40</b>
4.1 Metodología .....	40
4.1.1 Método de solución para el problema de ruteo de vehículos capacitado (solveCVRP(vCap)).....	43
<b>Experimentación computacional.....</b>	<b>48</b>
5.1 Descripción de las instancias .....	48
5.1.1 Instancias que se adecuaron al problema de ruteo de vehículos heterogéneo .....	48
5.1.2 Instancias especiales para el problema de ruteo de vehículos heterogéneo .....	51
5.1.3 Instancias del grupo A .....	53
5.1.4 Instancias del grupo B .....	54
5.1.5 Instancias del grupo T .....	55
5.2 Resultados y comparación de metodologías .....	56
<b>Conclusiones y trabajo futuro .....</b>	<b>69</b>
6.1 Conclusiones .....	69
6.2 Trabajo futuro.....	70
<b>Datos de instancias utilizadas .....</b>	<b>71</b>

<b>Algoritmo .....</b>	<b>75</b>
<b>Bibliografia .....</b>	<b>78</b>

# ÍNDICE DE FIGURAS

2.1 Ejemplo de rutas hechas por la flota de vehículos .....	6
2.2 Procedimiento de Clarke y Wright .....	13
2.3 Movimientos de búsqueda local .....	17
2.4 Movimiento $\lambda$ -Intercambio .....	18
4.1 Ejemplificación del método SAM .....	43
4.2 Unión de rutas .....	46
5.1 Algunas distribuciones de nodos del conjunto A .....	49
5.2 Algunas distribuciones de nodos del conjunto B .....	50
5.3 Algunas distribuciones de nodos del conjunto T .....	52
5.4 Comparación de resultados: SAM vs CWS del conjunto A .....	57
5.5 Comparación de resultados: SAM vs CWS del conjunto B .....	58
5.6 Comparación de resultados: SAM vs CWS del conjunto T (FC) .....	59
5.7 Comparación de resultados: SAM vs CWS del conjunto T (VC) .....	60
5.8 Comparación de resultados: SAM vs BS del conjunto T (FC) .....	61
5.9 Comparación de resultados: SAM vs BS del conjunto T (VC) .....	62
5.10 Comparación de tiempos de ejecución: SAM vs CWS del conjunto A....	63
5.11 Comparación de tiempos de ejecución: SAM vs CWS del conjunto B....	64
5.12 Comparación de tiempos de ejecución: SAM vs CWS del conjunto T(FC) .....	65
Comparación de tiempos de ejecución: SAM vs BS del conjunto T(FC) .....	66
5.14 Comparación de tiempos de ejecución: SAM vs CWS del conjunto T(VC) .....	67



5.15 Comparación de tiempos de ejecución: SAM vs BS del conjunto	
T(VC) .....	68

# ÍNDICE DE TABLAS

2.1 Simulación de Monte Carlo aplicada al VRP .....	24
3.1 Variantes del HVRP presentadas en literatura .....	29
3.2 Métodos relevantes de solución para el HVRP.....	31
5.1 Ejemplo de modificación de una instancia .....	51
5.2 Descripción de instancias del grupo A.....	53
5.3 Descripción de instancias del grupo B.....	54
5.4 Descripción de instancias del grupo T .....	55

## AGRADECIMIENTOS

Primero que nada quiero agradecer a mis padres, Francisco Javier y María Leticia, por su apoyo incondicional en cada una de las etapas de mi vida. A mis hermanas, Nathaly y Margarita, por estar siempre para mí cuando necesito con quien platicar.

Agradezco a mi prometido, Luis Sanabria, quien me habló sobre PISIS y su programa de posgrado. Gracias por hacer de mi vida una aventura y estar para mí en todo momento amor.

Agradezco a todos los profesores del Posgrado de Ingeniería de Sistemas, por los conocimientos que me compartieron a lo largo de estos 2 años.

Agradezco a mi asesora de tesis, Dra. Sara Rodríguez por el apoyo, consejos y paciencia brindados.

Agradezco al Dr. Ángel Juan, por el apoyo brindado durante el tiempo que estuve en Barcelona, España.

Agradezco a mi comité de tesis, al Dr. Javier Faulin por brindarme tiempo, conocimientos y por recibirme durante mi estancia académica en Pamplona, España, además de aportar ideas para el trabajo. A la Dra. Ada Álvarez por contribuir con la revisión del mismo.

A mis compañeros de PISIS, Yajaira, Vanesa, Lucero, muchas gracias por su amistad.

Finalmente, agradezco a CONACYT por la beca de tiempo completo para la realización de mi maestría y por la beca mixta otorgada para realizar una estancia en el extranjero. Y a la Facultad de Ingeniería Mecánica y Eléctrica y a la Universidad Autónoma de Nuevo León por el apoyo financiero otorgado para la realización de mi maestría.

# RESUMEN

Tatiana Quintero Quintero.

Candidato para el grado de Maestro en Ciencias en Ingeniería de Sistemas.

Universidad Autónoma de Nuevo León.  
Facultad de Ingeniería Mecánica y Eléctrica.

Título del estudio:

## ALGORITMO HÍBRIDO BASADO EN UN MÉTODO DE APROXIMACIONES SUCESIVAS PARA EL PROBLEMA DE RUTEO DE VEHÍCULOS HETEROGÉNEO

Número de páginas: 95

**OBJETIVOS Y MÉTODO DE ESTUDIO:** En esta tesis se aborda el problema de ruteo de vehículos heterogéneo (HVRP por sus siglas en inglés Heterogeneous Vehicle Routing Problem) que consiste en; dada una flotilla de vehículos con características diferentes, encontrar un conjunto de rutas de costo mínimo donde, se visite a un conjunto de clientes dispersos geográficamente para satisfacer su demanda.

El HVRP tiene numerosas aplicaciones en las operaciones logísticas. Este problema es un problema de programación entera, el cual cae dentro de la categoría de problemas NP-Hard, significando que los

esfuerzos computacionales requeridos para resolver este problema se incrementan exponencialmente con su tamaño. Visto el HVRP como un Problema de Optimización Combinatoria (OC), existen diferentes métodos de solución, los cuales son aplicados a HVRP de diferentes maneras y con diferentes enfoques.

El presente estudio aporta una alternativa de solución basada en un método de aproximaciones sucesivas, el cual, es una variante del algoritmo de los ahorros propuesto por Clarke y Wright (1964) y que además hace uso de las técnicas de memoria cache para mejorar las soluciones obtenidas.

Así, el objetivo general de la tesis consiste en diseñar, e implementar un algoritmo híbrido basado en un método de aproximaciones sucesivas para el problema de ruteo de vehículos heterogéneos.

Para dar validación a dicha metodología se realizó un experimento computacional.

CONTRIBUCIONES Y CONCLUSIONES: La principal contribución de esta tesis es proponer un nuevo método de solución que aporte soluciones eficientes al problema de ruteo de vehículos heterogéneos.

Firma del asesor: \_\_\_\_\_

Dra. Sara V. Rodríguez Sánchez

## CAPÍTULO 1

# INTRODUCCIÓN

### 1.1 DESCRIPCIÓN DEL PROBLEMA

El problema de ruteo de vehículos (VRP por sus siglas en inglés Vehicle Routing Problem) es uno de los problemas más estudiados en la literatura de Investigación de Operaciones. Su importancia radica en la gran variedad de aplicaciones en la vida real. Los problemas de ruteo son aquellos que tratan sobre la distribución de bienes entre depósitos y clientes. A grandes rasgos, el VRP consiste en, dado un conjunto de clientes y depósitos dispersos geográficamente, y una flota de vehículos, determinar un conjunto de rutas de costo mínimo que comiencen y terminen en los depósitos, para que los vehículos visiten a los clientes.

Estos problemas tienen múltiples variaciones las cuales dependen de la naturaleza del problema a resolver y de las características de la empresa en dónde se aplicará. Ejemplos de tales variaciones pueden encontrarse en la capacidad de los vehículos, la frecuencia de visitas al cliente hasta cumplir con su demanda, en los tipos de entregas y recolecciones, así como en los tiempos y condicionantes de la entrega.

En esta tesis tomaremos como supuestos principales, que el problema será: capacitado (capacidades específicas de los vehículos, los clientes tienen cierta demanda), así como la característica de

heterogeneidad (tenemos una flotilla de vehículos con características diferentes).

## 1.2 HIPÓTESIS

Dada la complejidad computacional del problema de ruteo de vehículos con flota heterogénea se vuelve difícil el resolverlo con métodos exactos, en un tiempo razonable. La hipótesis principal consiste en que el heurístico basado en un método de aproximaciones sucesivas obtendrá buenas soluciones para el HVRP, tales que, minimicen los costos del sistema actual en un tiempo razonable.

## 1.3 JUSTIFICACIÓN

La distribución de bienes desde un depósito hacia diferentes clientes juega un papel importante hoy en día en la gestión logística. Su correcta planificación genera ahorros significativos en el costo global del producto. Esto es porque los costos de transporte pueden llegar a representar hasta el 20% del costo final de los productos (Olivera, 2004). Estos ahorros, ya sea por usar menos vehículos y por consiguiente menos choferes, disminuyen entre el 5% y el 20% de los costos de transporte (Liong Choong et al., 2008).

La gran ventaja del enfoque que nosotros proponemos, es que se apega a la realidad ya que en las empresas es muy común el tener flotas de vehículos con diferentes características, siendo lo más habitual el que varíen las capacidades de dichos vehículos.



## 1.4 OBJETIVOS

El objetivo principal de esta tesis es el diseño e implementación de un heurístico basado en un método de aproximaciones sucesivas para el problema de ruteo de vehículos heterogéneos.

Por lo tanto se plantean los siguientes objetivos particulares:

1. Estudiar el problema de ruteo de vehículos heterogéneo.
2. Desarrollo e implementación de una metodología de solución.
3. Analizar los resultados obtenidos mediante la experimentación computacional.

## 1.5 ESTRUCTURA DE LA TESIS

La tesis está formada por seis capítulos. En el capítulo dos se abordan los antecedentes del problema de ruteo de vehículos, algunas variantes del problema y los diferentes métodos de solución que se muestran en literatura. El capítulo culmina con una sección dedicada a la Simulación de Monte Carlo y sus aplicaciones en este tipo de problemas.

En el capítulo tres se describe formalmente al problema de ruteo de vehículos heterogéneo, así como sus parámetros y variables, y se presenta su modelo matemático.

En el capítulo cuatro se describe la metodología propuesta para dar solución al problema de Ruteo de Vehículos Heterogéneo. El heurístico está basado en un método de aproximaciones sucesivas, el cual es una variante del algoritmo de los ahorros propuesto por Clarke y Wright

(1964) y que además hace uso de las técnicas de memoria cache para mejorar las soluciones obtenidas.

El capítulo cinco básicamente nos muestra algunos resultados obtenidos con la metodología propuesta.

Y finalmente, en el capítulo seis se muestran las conclusiones y el trabajo a futuro.

## CAPÍTULO 2

# ANTECEDENTES

### 2.1. EL PROBLEMA DE RUTEO DE VEHÍCULOS

Uno de los primeros estudios que se realizaron sobre el problema de ruteo de vehículos se remonta al año '59. Dantzig y Ramser (1959) formularon el primer modelo matemático, el cual consiste en la modelación matemática de un problema de distribución de gasolina a las diferentes estaciones de servicio. Los autores además describen el primer algoritmo de aproximación.

El Problema de Ruteo de Vehículos (VRP, por sus siglas en inglés) tiene muchas aplicaciones en una amplia variedad de industrias. Algunas de esas aplicaciones las podemos ver en empresas que distribuyen alimentos perecederos, bebidas, en los servicios públicos, operadores logísticos, recolección de basura, servicio postal, entre otros.

En concreto el VRP surge como una generalización del **problema del agente viajero** (TSP, por sus siglas en inglés). En este problema, un viajero tiene que visitar un conjunto de ciudades, pasando por cada ciudad exactamente una vez, para luego volver a la ciudad de partida, de tal forma que se minimice la distancia recorrida. Cuando se presenta el caso de existir más de un viajero, estamos hablando del problema de los  $m$  – agentes viajeros ( $m$ -TSP)

La definición formal del problema de ruteo de vehículos es, dado un conjunto de clientes y depósitos distribuidos geográficamente y una flotilla de vehículos, encontrar un conjunto de rutas que visiten solo una vez a los clientes, que empiecen y terminen en los depósitos, tal que el costo total sea minimizado.

El VRP tiene tres características principales, que son los clientes, el depósito y la flota de vehículos. Un **cliente** es una entidad que tiene cierta **demanda** de bienes. En algunos casos, la demanda no es un producto tangible sino un servicio. Los bienes son enviados desde un **depósito** mediante el uso de una flotilla de **vehículos**. Cada vehículo tiene una capacidad definida, que puede ser igual para todo el conjunto o puede haber sub-conjuntos con capacidades diferentes. Los subconjuntos se expresan en términos de peso, volumen, número de clientes, entre otros. Así mismo, cada vehículo empieza y termina su **ruta** (indicando la secuencia en que los clientes serán visitados, ver la Figura 2.1) en el depósito, satisfaciendo la demanda de todos los clientes y minimizando los costos.

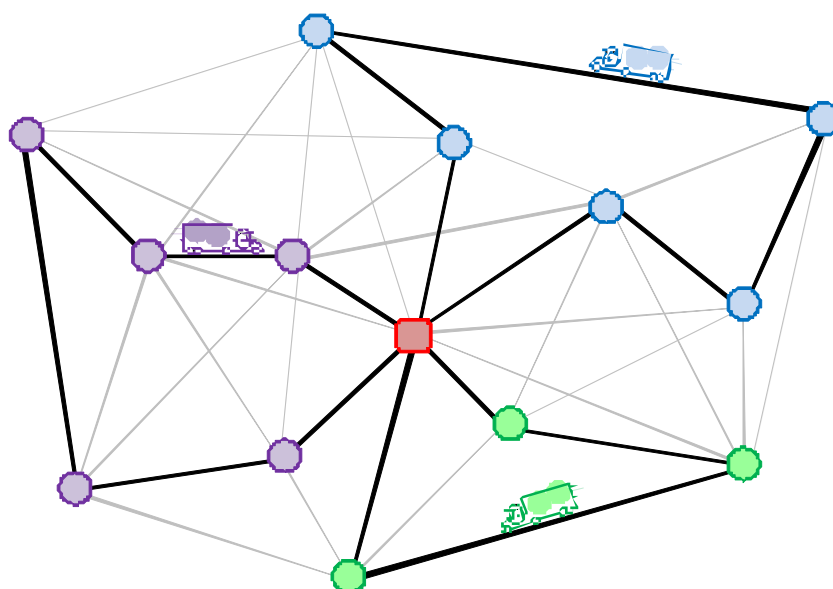


Figura 2.1 Ejemplo de rutas hechas por la flota de vehículos

Fuente: <http://www.r-bloggers.com/wp-content/uploads/2010/11/vrp.jpg>

Este problema opera sobre una red de caminos conocida también con el término de grafo. En el grafo los arcos representan los tramos de carretera y cuyos vértices corresponden a las uniones de caminos, a las locaciones del depósito y de los clientes. Los arcos pueden ser dirigidos y no dirigidos, dependiendo de si pueden viajar en una sola dirección o en ambas direcciones. Cada arco tiene asociado un costo, el cual, por lo general, representa su longitud, y los tiempos de recorrido que posiblemente dependen del tipo de vehículo (Toth y Vigo, 2002).

El problema de ruteo de vehículos es un problema muy conocido que cae dentro de la categoría de NP-duro. Lo anterior quiere decir que, para poder encontrar una solución óptima, se necesita un esfuerzo computacional muy grande. Esto equivale a decir, que el tiempo que tarda el problema tratando de encontrar una solución óptima crece con el tamaño del problema. Entonces, se puede decir que entre más grande sea el problema más complicado será encontrar una solución óptima, o en su defecto una solución que sea factible.

Este problema sigue siendo muy popular entre la comunidad científica porque, hasta la fecha, no existe un algoritmo que lo resuelva en tiempo polinomial de manera óptima. Se han propuesto varios métodos para la solución del VRP y todas sus variantes.

### 2.1.1 ALGUNAS VARIANTES DEL PROBLEMA DE RUTEO DE VEHÍCULOS

El campo de aplicación del problema de ruteo de vehículos es muy amplio. Cada campo presenta características propias surgiendo con ello un gran número de variaciones a dicho problema. Por ejemplo, problemas donde la capacidad de los vehículos es heterogénea, o problemas con ventanas de tiempo, tamaño máximo de ruta, permitir entregas divididas,

entre otros. Las variantes más estudiadas de la literatura se presentan a continuación.

#### PROBLEMA DE RUTEO DE VEHÍCULOS CAPACITADO

El problema de ruteo de vehículos capacitado (CVRP por sus siglas en inglés) es la variante más común del problema de ruteo de vehículos. En esta variante se incluyen las restricciones de capacidad de los vehículos la cual no debe de ser sobrepasada, esto es, la suma de las demandas de los clientes de cada ruta no debe de ser mayor a la capacidad del vehículo que la recorre. Además, los clientes tienen una demanda específica a ser servida. En otras palabras, el problema de ruteo de vehículos capacitado es el problema en donde un conjunto de vehículos idénticos, localizados en un depósito central, deben de ser ruteados para satisfacer la demanda conocida de los clientes, sujeto a las restricciones de capacidad de los vehículos (Baldacci et al., 2007).

En este problema la demanda de los clientes no solo tiene un peso sino también una forma; los vehículos no solo tienen una capacidad sino también un espacio limitado.

#### PROBLEMA DE RUTEO DE VEHÍCULOS CON FLOTA HETEROGÉNEA

Otra de las variantes clásicas (y más apegadas a la vida real) del problema de ruteo de vehículos capacitado se presenta cuando la flotilla de vehículos está formada por vehículos con características y/o capacidades diferentes, en este caso estamos ante el problema de ruteo de vehículos con flota heterogénea (HVRP, por sus siglas en inglés).

Este problema también es conocido como problema de ruteo de vehículos con flota mixta, y los primeros en considerarlo de forma estructurada fueron Golden et al. (1984). Baldacci et al. (2007) realizaron

una recopilación de las diferentes variantes de este problema y de quienes las han abordado hasta esa fecha, la cual se explicara más a detalle en la sección 2.4.

#### PROBLEMA DE RUTEO DE VEHÍCULOS CON VENTANAS DE TIEMPO

El Problema de Ruteo de Vehículos con Ventanas de Tiempo (VRPTW, por sus siglas en inglés) surge cuando se impone un periodo de tiempo específico en que debe de ser entregado el producto. En esta variación, además de tener las restricciones de capacidad, se agregan restricciones sobre la hora de entrega de los bienes, es decir, solo se puede visitar al cliente en un periodo establecido de tiempo (ventana de tiempo).

#### PROBLEMA DE RUTEO DE VEHÍCULOS CON CARGAS Y DESCARGAS

En ocasiones la naturaleza del problema no solo requiere realizar entregas, sino también se tiene que recoger artículos y regresarlos al depósito (como en el caso de la distribución de bebidas). Cuando se presenta esta variación, se le conoce como el problema de ruteo de vehículos con cargas y descargas (VRPPD, por sus siglas en inglés).

Este problema se puede dividir en dos sub-problemas, uno de entrega de bienes (linehaul) y otro de recolección de bienes (backhaul), indicando que algunos vehículos realizaran las entregas y otros las recolecciones. Cuando todas las entregas tienen que hacerse antes de cualquier recolección estamos ante el problema de ruteo de vehículos con recolecciones (VRPB, por sus siglas en inglés).

Es importante hacer notar que algunos autores no consideran diferencia alguna entre el VRPPD y el VRPB (Liong et. al, 2008).

## PROBLEMA DE RUTEO DE VEHÍCULOS CON ENTREGAS DIVIDIDAS

Si se relaja la restricción de que el cliente sea visitado solo una vez, estamos ante el problema de ruteo de vehículos con entregas divididas (SDVRP, por sus siglas en inglés). En esta variante se permite que un mismo cliente sea visitado por dos o más vehículos. Además una característica importante de este problema es que permite que las demandas de los clientes sean mayores a la capacidad de los vehículos.

## PROBLEMA DE RUTEO DE VEHÍCULOS ESTOCÁSTICO

En las formulaciones anteriores se asume que todos los parámetros necesarios para su implementación son conocidos. Sin embargo, esto no es siempre posible, por ejemplo cuando no se conozca el número exacto de clientes a servir. En este entorno estamos ante un problema de ruteo de vehículos estocástico (VRPS, por sus siglas en inglés).

## RUTEO EN LOS ARCOS

Cuando en lugar de visitar todos los nodos de un grafo se requiere visitar todos los arcos, estamos hablando del problema de ruteo en los arcos (ARP, por sus siglas en inglés). En esta variante, los clientes, en lugar de estar situados en los vértices, se encuentran sobre los arcos de la red. Un ejemplo de esto es la recolección de basura, ya que los camiones tienen que recorrer todas las calles (arcos) para completar el servicio.

Para una revisión más extensa sobre el problema de ruteo de vehículos capacitado y sus variantes, se recomienda el capítulo 1 del libro de Toth y Vigo (2002).



## 2.2 MÉTODOS DE SOLUCIÓN PARA EL PROBLEMA DE RUTEO DE VEHÍCULOS

Se distinguen tres clases de métodos de solución para el problema de ruteo de vehículos: algoritmos **Exactos**, métodos **Heurísticos** y las **Metaheurísticas**.

### 2.2.1 ALGORITMOS EXACTOS

Muy pocos métodos exactos se han propuesto para la resolución del VRP, ya que, dada la complejidad del problema, son muy pesados computacionalmente, es decir, tardan un tiempo excesivo en resolver instancias pequeñas, de no más de 50 clientes (Olivera, 2004).

De entre los métodos exactos que se destacan, están el **Ramificación y Acotamiento** (*Branch and Bound*) y el **Ramificación y Corte** (*Branch and Cut*). En general, estos dos métodos trabajan bajo el mismo concepto de crear un árbol de soluciones e ir “podando” ramas, bajo un criterio establecido, hasta llegar a una solución óptima.

Se han propuesto también algoritmos basados en Programación Dinámica, los cuales aceleran los cálculos mediante la relajación del espacio de estados. Por otra parte, la implementación del método de Generación de Columnas ha resultado un tanto efectivo para los problemas con ventanas de tiempo muy ajustadas (Olivera, 2004).

### 2.2.2 MÉTODOS HEURÍSTICOS

Estos métodos se empezaron a desarrollar entre 1960 y 1990. En los últimos años se han propuesto una gran cantidad de métodos

heurísticos para la solución del problema de ruteo de vehículos. Estos métodos realizan una búsqueda limitada en el espacio de soluciones, las cuales encuentran soluciones buenas en un tiempo corto. Tienen la desventaja de que no garantizan obtener el valor óptimo, pero como ventaja encontrarán soluciones robustas, sencillas y rápidas que pueden ofrecer una mejora considerable al sistema actual.

Las heurísticas son usadas cuando no existe un método exacto o bien cuando el método exacto se “estanca”, es decir llega a un punto en que no encuentra una solución factible en un periodo largo de tiempo. Estas suelen clasificarse en tres categorías: heurísticas constructivas, heurísticas de dos fases, y métodos de mejora, las cuales se explicaran a continuación.

#### A) HEURÍSTICAS CONSTRUCTIVAS

Las Heurísticas Constructivas en cada paso van agregando un nodo a la vez a la solución (de ahí su nombre), mientras se cuida que el costo no sea muy alto, hasta obtener una solución factible.

Quizás la heurística constructiva más popular debido a su simplicidad y velocidad es el **Algoritmo de los Ahorros de Clarke y Wright** ó simplemente algoritmo de los ahorros. Esta heurística fue desarrollada por Clarke y Wright en 1964.

El algoritmo de los ahorros trabaja del siguiente modo: en primer lugar asigna a cada cliente un vehículo, después, trata de combinar dos rutas, siempre y cuando la capacidad del vehículo lo permita. Para determinar el orden en que serán combinadas las rutas se hace uso del ahorro que genera la unión de dos clientes de dos rutas diferentes. Dicho ahorro se calcula de la siguiente manera:  $s_{ij} = c_{0i} + c_{0j} - c_{ij}$ , estos ahorros

son ordenados de forma descendiente y se selecciona el que genere más ahorro y se agrega a la ruta.

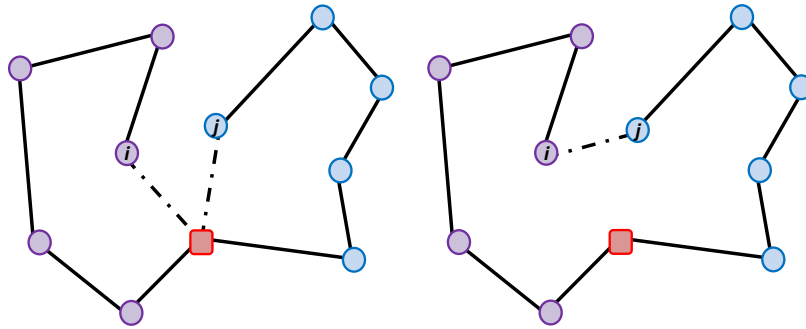


Figura 2.2 Procedimiento de Clarke y Wright

Este algoritmo tiene dos enfoques, el Paralelo y el Secuencial. El enfoque paralelo va creando rutas simultáneamente, mientras que el enfoque secuencial crea una ruta a la vez. Rand (2009) ofrece una gran revisión sobre el algoritmo de los ahorros aplicado a las variantes más importantes del VRP.

Dentro de esta clase están también las **Heurísticas de Inserción**, estas, en cada iteración agregan un cliente nuevo a la ruta, respetando las restricciones de capacidad del vehículo. Tiene dos enfoques similares a los del Algoritmo de los Ahorros. En su versión secuencial esta heurística solo considera insertar clientes en la última ruta creada. Lo anterior tiene la desventaja de que los últimos clientes no visitados por lo general tienden a estar dispersos provocando que las últimas rutas tengan un costo elevado.

Uno de los algoritmos más representativos es el de **Mole y Jameson** (1976). Este algoritmo utiliza los parámetros  $\lambda$  y  $\mu$  para expandir una ruta bajo construcción, en donde mide la costo de inserción para cada vértice. Dicho costo se calcula de la siguiente forma:  $\alpha(i, j, k) = c_{ik} + c_{kj} - \lambda c_{ij}$  e inserta el vértice con mayor rendimiento

$\beta(i, k, j) = \mu c_{ok} - \alpha(i, k, j)$ . Finalmente, optimiza la ruta actual mediante un movimiento 3-opt (el cual se explicará más adelante). Y se realiza esta operación hasta que no exista una nueva inserción factible por realizar.

Debido a la clara desventaja de la versión secuencial, se desarrolló la versión en paralelo, la cual permite insertar clientes en cualquiera de las rutas de la solución.

Como ejemplo para este tipo de heurísticas está el algoritmo de **Christofides et al. (1979)**. Este algoritmo está dividido en dos fases, en la primera fase utiliza el algoritmo de inserción secuencial para obtener un conjunto de rutas factibles. En la segunda fase, para cada ruta encontrada en la primera fase, se selecciona un cliente y se inicializa una ruta nueva para cada uno de ellos. Entonces los demás clientes no ruteados se van insertando siguiendo un criterio de peso. El criterio utilizado es la diferencia entre dos inserciones potenciales. Una vez que todos los clientes pertenecen a alguna ruta, se mejoran las rutas mediante el movimiento 3-opt.

## B) HEURÍSTICAS DE DOS FASES

Las Heurísticas de Dos Fases consisten en descomponer el problema en dos componentes naturales. En una fase se agrupan los vértices en rutas o grupos factibles y en la otra fase se construye la ruta real. Es posible que haya ciclos de retroalimentación entre ambas fases.

Se tienen dos clases de este tipo de heurística, la cuales son: métodos de Agrupar Primero, Rutear Después, y métodos de Rutear Primero, Agrupar Después.

Los **Métodos de Agrupar Primero, Rutear Después** proceden en dos fases. En la primera fase se crean los grupos de clientes o clústeres, los

clientes pertenecientes a este grupo estarán en una misma ruta en la solución final. Al momento de generar los grupos debe de tomarse en cuenta que la suma de las demandas asociadas a los clientes no debe de sobrepasar la capacidad del vehículo. En la segunda fase se crean las rutas para cada uno de los grupos de clientes, aplicando un TSP, el cual se resolverá de forma exacta o aproximada, dependiendo del tamaño del clúster.

Un método característico de este tipo es la Heurística de Barrido o *Sweep*. Ésta genera los clústeres girando una semirrecta con origen en el depósito “barriendo” a los clientes a su alcance e incorporándolos al grupo mientras no se viole la restricción de capacidad del vehículo. Y después, se aplica a cada clúster un TSP, ya sea de forma exacta o aproximada, para definir la secuencia de la ruta.

Los **Métodos de Rutear Primero, Agrupar Después**, al igual que los Métodos de Asignar Primero, Rutear Después, trabajará en dos fases. En una primera fase, resuelve un TSP incluyendo a todos los clientes. En esta fase se dejan de lado las restricciones del problema lo que hace infactible a esta ruta. En la segunda fase, se crean rutas factibles particionando la ruta creada en la primera fase.

Como ejemplo están los Algoritmos de Pétalos, el cual supone que existe un conjunto de rutas factibles, pero con la peculiaridad de que cada cliente está incluido en varias rutas. Entonces lo que hace este algoritmo es encontrar un sub-conjunto de rutas factibles de costo mínimo, en las cuales, cada cliente es visitado solo una vez. Esto se formula como un Problema de Partición de Conjuntos (SPP).

### C) HEURÍSTICAS DE MEJORA

Y por último, pero no menos importantes, están las Heurísticas de Mejora. Estas pueden operar de dos formas diferentes, una es que operen sobre un vehículo individual, y la otra forma es que operen sobre varias rutas al mismo tiempo.

En este grupo están los algoritmos de **Búsqueda Local**, los cuales se aplican una vez que se tiene una solución inicial para el problema. Básicamente, realizan pequeños movimientos en los arcos con la finalidad de mejorar la solución actual. Esto es, para cada solución  $s$  se define un conjunto de soluciones vecinas  $N(s)$ , entonces parte de  $s$  y la reemplaza por un solución  $s^* \in N(s)$  de menor costo. Se repite este procedimiento hasta que no se encuentre otra solución mejor. A la solución encontrada al final se le conoce como óptimo local.  $N(s)$  son las soluciones que pueden ser obtenidas aplicando algún procedimiento sencillo o regla de movida a  $s$ . Estos procedimientos pueden ser movidas de una ruta, en donde los clientes en la ruta siguen siendo los mismos después del movimiento, lo que cambia es el orden en que son visitados por el vehículo. Ó bien, pueden ser movidas entre rutas, que en este procedimiento, además de cambiarse el orden en que se visitan los clientes, también hay cambios en los conjuntos de los clientes visitados en cada ruta.

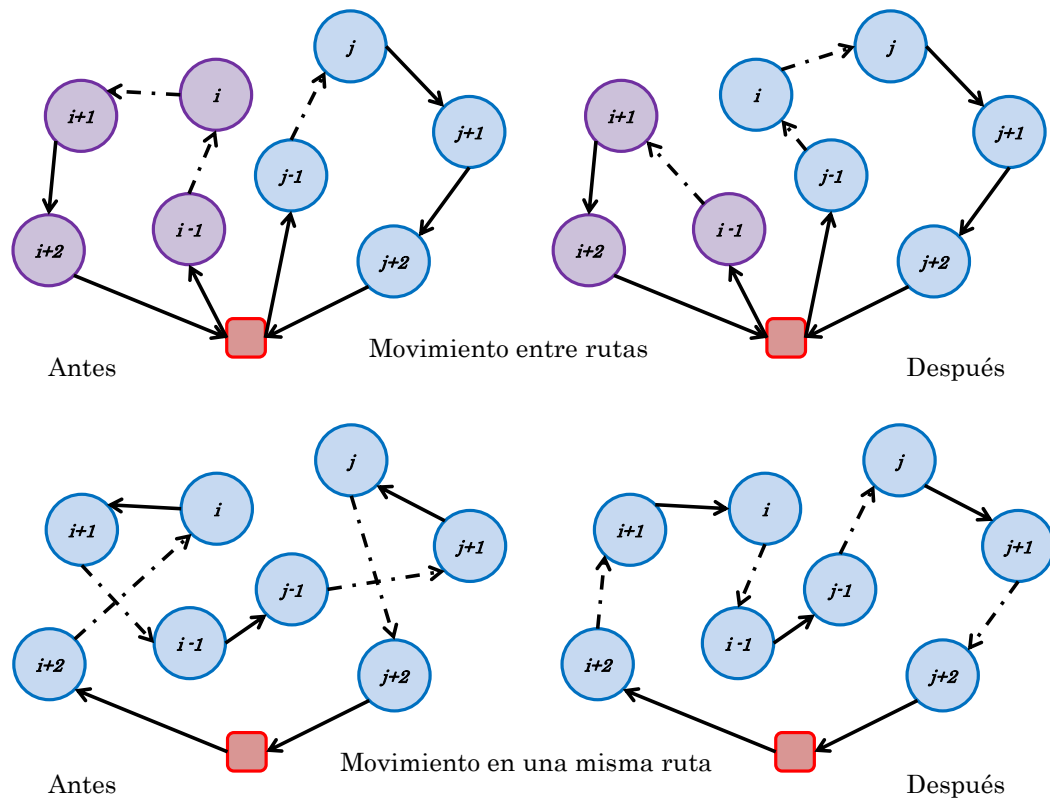


Figura 2.3 Movimientos de búsqueda local

El **Movimiento** u **Operador**  $\lambda$ -**Intercambio** también pertenece a la clase de las heurísticas de mejora y es uno de los más conocidos. Éste consiste en eliminar  $\lambda$  arcos de la solución y reconectar los  $\lambda$  segmentos restantes.

Se le conoce como  $\lambda$ -opt a un algoritmo de búsqueda local que utiliza  $\lambda$ -intercambios hasta alcanzar una solución  $\lambda$ -óptima, la cual se alcanza cuando la solución ya no puede ser mejorada utilizando  $\lambda$ -intercambios. Comúnmente se implementan 2-intercambios ó 3-intercambios.

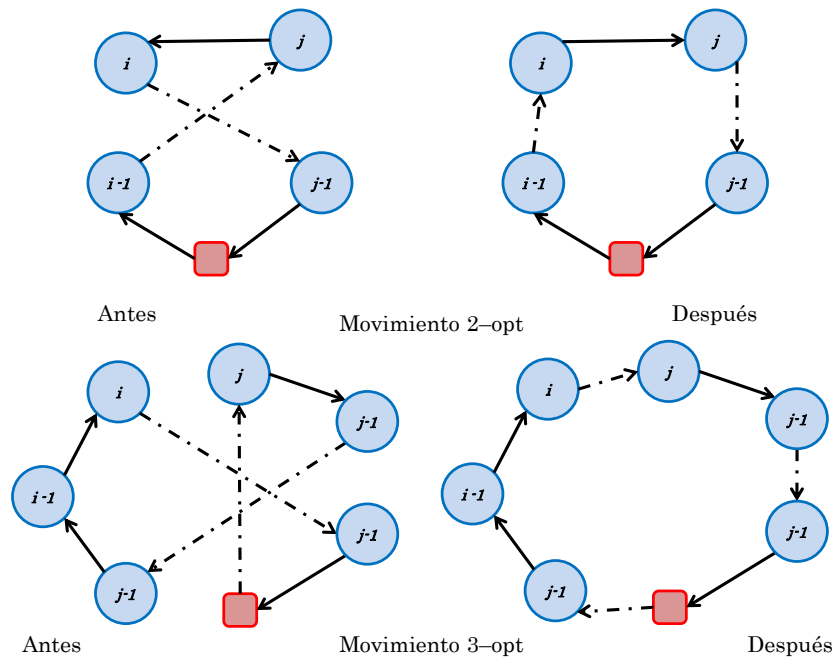


Figura 2.4 Movimiento  $\lambda$ -Intercambio

### 2.2.3 METAHEURÍSTICAS

Una metaheurística es un procedimiento diseñado para encontrar mejores soluciones que una heurística simple, ya que ésta utiliza a las heurísticas clásicas como subrutinas. Las metaheurísticas realizan una mejor exploración en el espacio de soluciones.

En los últimos años se han propuesto varias metaheurísticas para el problema de ruteo de vehículos. Generalmente son procedimientos que exploran el espacio de soluciones para encontrar soluciones buenas. Estas utilizan los métodos heurísticos como subrutinas. Las metaheurísticas permiten deteriorar las soluciones intermedias o incluso rozar la infactibilidad en el proceso de búsqueda.

Las principales metaheurísticas que se han aplicado al problema de ruteo de vehículos son el Recocido Simulado (SA), Búsqueda Tabú (TS), y los Sistemas de Hormigas (AS). El recocido simulado y la



búsqueda tabú empiezan a trabajar desde una solución inicial conocida, y en cada iteración se mueven de una solución a otra en un mismo vecindario hasta que se cumple una condición dada. En los sistemas de hormigas en cada iteración se crean varias soluciones usando información de las iteraciones anteriores.

Taillard et al. (1998) puntualiza que la búsqueda tabú, y los sistemas de hormigas son métodos que guardan información sobre las soluciones encontradas y la utilizan para encontrar soluciones aún mejores.

#### A) SISTEMAS DE HORMIGAS

Los Sistemas de Hormigas (AS) están inspirados en la estrategia utilizada por las hormigas cuando están en busca de alimentos, por ejemplo, cada vez que una hormiga encuentra comida, en el regreso al hormiguero va dejando un rastro de feromona, la cual es detectada por otras hormigas que a su vez van dejando más feromonas, haciendo que ese camino sea la mejor opción para encontrar comida.

#### B) RECOCIDO SIMULADO

El Recocido Simulado (SA) es un algoritmo de búsqueda para problemas de optimización global, el cual tiene como objetivo general encontrar una buena aproximación al valor óptimo de una función en un espacio de búsqueda grande, conocido como óptimo global.

#### C) BÚSQUEDA TABÚ

Otra metaheurística importante es la Búsqueda Tabú (TS) la cual examina la secuencia de soluciones como en el recocido simulado, pero el

siguiente movimiento se hace al vecindario donde se encuentre una solución mejor a la actual.

## 2.3 SIMULACIÓN DE MONTE CARLO

### 2.3.1 ANTECEDENTES DE LA SIMULACIÓN

La simulación es una de las técnicas de la investigación de operaciones. Esta herramienta hace uso de las computadoras para imitar (simular) la operación de un proceso o sistema (Hillier and Liberman, 2001). Por ejemplo, la simulación es utilizada frecuentemente para realizar análisis riesgosos de procesos financieros mediante la repetición de la evolución de las transacciones involucradas para generar un perfil con las posibles salidas. La simulación también es ampliamente utilizada para analizar sistemas estocásticos que continúan operando indefinidamente. Para tales sistemas, la computadora genera y guarda las ocurrencias de varios eventos que manejan al sistema tal como si fueran operados física o manualmente.

#### LA ESENCIA DE LA SIMULACIÓN

La técnica de simulación ha sido una herramienta importante del diseñador. Por ejemplo, simular el vuelo de un avión en un túnel de viento es una práctica común cuando se diseña un nuevo avión. Otra alternativa sería el construir aviones reales con diseños alternativos y probarlos en vuelos reales para elegir el diseño final, pero esto sería tan caro como inseguro.

La simulación juega un papel esencialmente similar en muchos estudios de investigación de operaciones. Sin embargo, en lugar de diseñar un avión, al equipo de investigación de operaciones le concierne

desarrollar un diseño o procedimiento para algunos sistemas estocásticos, que son sistemas que evolucionan probabilísticamente con el tiempo. Algunos de esos sistemas estocásticos se parecen a los ejemplos de cadenas de Markov y sistemas de líneas de espera y otros más complicados. En lugar de utilizar un túnel de viento, el funcionamiento del sistema real se imita usando distribuciones de probabilidad para la generación aleatoria de varios eventos que ocurren en el sistema.

Esta es típicamente utilizada cuando el sistema estocástico involucrado es muy complejo para ser analizado satisfactoriamente por los modelos matemáticos existentes.

#### SIMULACIÓN DE EVENTOS DISCRETOS VS CONTINUOS

En la **simulación de eventos discretos** los cambios en el estado del sistema ocurren instantáneamente en puntos aleatorios en el tiempo como resultado de eventos discretos. Por ejemplo, en los sistemas de línea de espera, donde el estado del sistema está dado por el número de clientes en el sistema, los eventos discretos que cambian este estado son la llegada de nuevos clientes y la salida de otros cuando han sido servidos.

La **simulación de eventos continuos** es aquella en donde los cambios del estado de sistema ocurren continuamente en el tiempo. Por ejemplo, si el sistema de interés es el vuelo de un avión, entonces su estado está definido por la posición actual de dicho avión, y ese estado cambia continuamente en el tiempo.

#### NÚMEROS ALEATORIOS

Existen en literatura varias tablas de números aleatorios que fueron generados usando un dispositivo físico, como un disco giratorio,

una de estas tablas contiene un millón de dígitos aleatorios, publicada por *The Rand Corporation*.

Al procedimiento que realiza una computadora para obtener números aleatorios se le conoce como **generador de números aleatorios**. Este generador es un algoritmo que realiza secuencias de números que siguen una distribución de probabilidad específica y poseen la característica de aleatoriedad.

Una secuencia de números significa que el algoritmo produce muchos números aleatorio de forma serial. La distribución de probabilidad implica la probabilidad de ocurrencia en que cada número será generado por el algoritmo.

Se puede dividir a los números aleatorios en dos categorías principales, los **números aleatorios enteros** y los **números pseudoaleatorios**. Existen los números aleatorios enteros, los cuales son una observación de una distribución uniforme discreta sobre el rango  $\underline{n}, \underline{n}+1, \dots, \bar{n}$ , así como también existen los números aleatorios uniformes, que son observaciones de una distribución uniforme continua en el intervalo  $[a, b]$ .

Estrictamente hablando, los números generados por una computadora no deben de ser llamados números aleatorios porque son predecibles y éstos se pueden reproducir dado que se utiliza un generador de números aleatorios, por lo cual se les suele llamar números pseudoaleatorios.

#### TÉCNICAS DE REDUCCIÓN DE LA VARIANZA

En simulación, es importante obtener tanta información precisa como sea posible de la cantidad de simulaciones que pueden ser

realizadas. Desafortunadamente, en la práctica existe una tendencia de aplicar la simulación sin adecuar la eficiencia al diseño de experimentos. Entonces, para reducir la varianza entre resultados de una simulación a otra se han empleado las técnicas de reducción de la varianza a las cuales también se les llama **técnicas de Monte Carlo** (un término también aplicado a la simulación en general).

### 2.3.2 LAS TÉCNICAS DE MONTE CARLO

Las técnicas de Monte Carlo son métodos no determinísticos usados para aproximar expresiones matemáticas complejas y costosas de evaluar con exactitud. El nombre Monte Carlo de esta clase de métodos fue dado por John von Neumann y Stanislaw Ulam en 1946. Estos métodos proporcionan soluciones aproximadas a una gran variedad de problemas matemáticos posibilitando la realización de experimentos con muestreos de números pseudoaleatorios en una computadora.

En general, se puede definir a la simulación de Monte Carlo como un conjunto de técnicas que hacen uso de la generación de números aleatorios para dar solución a ciertos problemas estocásticos o deterministas. La simulación de Monte Carlo ha sido de mucha ayuda para obtener resultados numéricos de problemas complejos, los cuales no se pueden resolver mediante enfoques analíticos.

### 2.3.3 APLICACIÓN DE LA SIMULACIÓN DE MONTE CARLO AL PROBLEMA DE RUTEO DE VEHÍCULOS.

Como hemos visto, el problema de ruteo de vehículos es uno de los más estudiados y tratados en literatura, pero muy pocos trabajos se han

desarrollado en donde se integre la simulación de Monte Carlo para dar solución a estos problemas, ver Tabla 2.1.

Propuesta	Autores	Bases	Comentarios
	Buxey	CWS + MCS	Itera la lista de ahorros de forma totalmente aleatoria.
	Fdez. de Córdoba et al.	NNI + MCS	Asigna probabilidades a los nodos dependiendo de la distancia entre ellos y la ruta en formación.
ALGACEA-1	Faulin & Juan	CWS + MCS	Utiliza una función de entropía para asignar una distribución de probabilidad para el proceso de selección de nodos.
SR-1	Faulin et. al	CWS + MSC	Asigna una distribución probabilística a las distancias entre dos nodos consecutivos en una ruta.
SR-GCWS	Juan et. al	CWS + MCS	Asigna una probabilidad de selección a cada par de nodos en la lista de ahorros.
Binary MCS-CWS	Takes & Kusters	CWS + MCS	Asigna una probabilidad dependiendo de la factibilidad del par de nodos en la lista de ahorros.
SR-GCWS-CS	Juan et al.	CWS + MCS + C + S	Agregas las técnicas de caché y división a las rutas generadas,

Tabla 2.1 Simulación de Monte Carlo aplicada al VRP

### Notación:

CWS: Heurística de los ahorros (Clarke and Wright Savings heuristic)

MCS: Simulación de Monte Carlo (Monte Carlo Simulation)

NNI: Heurística de la inserción más cercana (Nearest Neighbour Insertion)

C: Técnicas de caché (Cache techniques)

S: Técnicas de división (Splitting techniques)

El primero en sugerir el uso de la simulación de Monte Carlo para la resolución del problema de ruteo de vehículos fue Buxey (1979). En este método, se combina la simulación de Monte Carlo con la heurística de los ahorros de Clarke y Wright (1964) en su forma secuencial. Aquí, cualquier programación de ruta evolucionará mediante la elección aleatoria de un par de nodos de la lista ordenada.

Poco más de 20 años después, Fernández de Córdoba et al. (2000) resolvieron con éxito un problema de distribución de comida en la ciudad de Valencia, España. Este algoritmo simula el viaje de un vehículo de un nodo a otro dependiendo de cierta probabilidad. Dicha probabilidad depende de cuantas rutas serán designadas. El algoritmo se corre durante un cierto número de iteraciones (rutas) y se selecciona la mejor solución encontrada.

Años después, Faulin y Juan (2008) desarrollaron una heurística basada en Entropía ALGACEA-1 (acrónimo de algoritmo de ahorros con entropía limitada – primera entrega) para dar solución al CVRP. Básicamente, ALGACEA-1 es un método en donde se utiliza una función de entropía para guiar la selección aleatoria de los nodos mediante una probabilidad previamente establecida. Entropía es un concepto matemático usado para construir distribuciones de probabilidades discretas para el proceso de selección de nodos.

En el mismo año, Faulin et al. (2008) desarrollaron el SR-1 (acrónimo de SimoRoute algorithm, first release). Este es un algoritmo que utiliza a la simulación de Monte Carlo para generar rutas y soluciones al problema. Este algoritmo, primeramente utiliza una solución “inicial buena”, la cual la obtenemos mediante el algoritmo de los ahorros (CWS), de esa solución se obtienen los datos necesarios para entrar a la siguiente fase. Estos datos son ajustados a una distribución estadística la cual caracteriza las distancias entre nodos en la solución. Entonces, estas distribuciones se utilizan para generar un gran número de soluciones aleatorias de tamaño similar. En ese punto, se realiza una búsqueda local aleatoria, y se obtiene una lista con las mejores soluciones. El método sigue iterando y al final arroja la mejor solución encontrada.

Otro método que involucra a la simulación de Monte Carlo para dar solución al problema de ruteo de vehículos capacitado es el SR-GCWS (Juan et al. 2010). El SR-GCWS (Simulation in Routing via the Generalized Clarke and Wright Savings heuristic) es un algoritmo híbrido que combina la versión paralela de la heurística clásica de los ahorros de Clarke y Wright (CWS) con la simulación Monte Carlo (SMC) y generadores de números aleatorios modernos para producir un conjunto de soluciones alternativas para una instancia dada del CVRP. Este enfoque asigna una probabilidad de selección a cada arista en la lista de ahorros en lugar de escoger a aquel que represente el mayor ahorro (como lo haría el CWS clásico). Además, esta probabilidad debe de ser coherente con el valor del ahorro asociado con cada arista. Finalmente, este proceso de selección debería de hacerse sin introducir demasiados parámetros en la metodología.

Takes y Kusters (Takes y Kusters, 2010) trabajaron con la simulación de Monte Carlo (MCS) y el algoritmo de los ahorros (CWS) en conjunto, desarrollando el *BinaryMCS-CWS*. En esencia, itera sobre la



lista de ahorros, ordenándola de forma descendiente por el valor del ahorro generado. Si un ahorro es factible, se realizan  $r$  simulaciones aleatorias para el estado actual y el estado donde el ahorro será procesado. En cada simulación aleatoria, la lista de ahorros es procesada desde el valor más alto al más bajo. Cada par es procesado o ignorado de acuerdo a una cierta probabilidad.

Finalmente, Juan et al. (2011) desarrollaron el SR–GCWS–CS (*Simulation in Routing via the Generalized Clarke and Wright Savings heuristic with Cache and Splitting techniques*). Este método es una extensión del SR–GCWS (explicado anteriormente), pero con la variante de que se agregó un sistema de memoria para que guarde las mejores soluciones encontradas además que se aplicaron técnicas de *splitting* para mejorar aún más las rutas encontradas.

Los algoritmos antes mencionados toman como supuesto que todos los vehículos tienen la misma capacidad, y en algunos casos la flota de vehículos se asume infinita, es decir, que no tienen restricción alguna en el número de vehículos a utilizar.

## 2.4 APORTACIONES EN LITERATURA AL PROBLEMA DE RUTEO DE VEHÍCULOS HETEROGÉNEO

El VRP pertenece a la clase *NP–Hard* en el sentido fuerte ya que es una generalización del ya conocido problema del agente viajero (TSP por sus siglas en inglés). Debido a esto, este problema ha sido de gran interés entre los investigadores en el campo de investigación operativa.

Una extensión más realista del VRP considera que la flotilla de vehículos es heterogénea. Esta variante ha tenido menos atención ya que es más difícil de resolver, y hasta el momento no hay un algoritmo que lo

pueda resolver en tiempo polinomial. La mayoría de los artículos publicados tratan sobre el problema de vehículos con flota mixta (VFMP por sus siglas en inglés).

En el VFMP la cantidad de vehículos de cada tipo es ilimitada y cada tipo de vehículo tiene asociado un costo fijo u un costo por unidad de distancia recorrida, y su objetivo es minimizar la suma de los costos implicados. Osman y Salhi (1996) no consideran los costos variables, y Gendreau et al. (1999) solo toman en cuenta los costos variables. Los costos variables son aquellos, que por ejemplo, están en función al número de unidades a transportar.

En el VFMP se acepta que el número de vehículos sea ilimitado solo cuando se quiere establecer el tamaño de la flotilla de vehículos en un futuro, pero no es realista, ya que involucra inversiones a mediano plazo y a corto plazo.

En el problema de ruteo de vehículos con flota heterogénea (HVRP, por sus siglas en inglés) el número de vehículos de cada tipo es fijo, lo que lo hace más realista que el VFMP ya que hace un mejor uso de una flotilla existente desde un punto de vista operacional.

El primero en abordar el HVRP tomando en cuenta ambos tipos de costos (fijos y variables) fué publicado por Taillard (1999). Este problema también es conocido como problema de ruteo de vehículos con flota mixta, y el primero en considerarlo de forma estructurada fueron Golden et al. (1984).

En la literatura, se observa que este problema se nombra de manera diferente, pero se muestra cierta homogeneidad al hablar de los problemas de ruteo de vehículos heterogéneo, donde el número de

vehículos es limitado, y el problema de ruteo de vehículos con el tamaño y composición de la flotilla mixto, donde los vehículos son ilimitados.

Baldacci et al. (2007) realizaron una recopilación de las diferentes variantes del HVRP y de quienes las han abordado hasta esa fecha, ver Tabla 3.1.

Nombre del problema	Tamaño de la flotilla	Costo fijo	Costo de ruteo	Referencias
HVRPFD	Limitado	Considerado	Dependiente	Li et al (2006)
HVRPD	Limitado	No Considerado	Dependiente	Taillard (1999), Gendreau et al. (1999), Prins (2002), Tarantilis et al. (2003), Tarantilis et al. (2004), Li et al. (2006)
SDVRP	Limitado	No Considerado	Dependiente del Lugar	Nag et al. (1988), Chao et al. (1999), Cordeau y Laporte (2001), Pisinger y Ropke (2007)
FSMFD	Ilimitado	Considerado	Dependiente	Ferland y Michelon (1988), Teodorovic et al. (1995), Choi y Tcha (2007)
FSMD	Ilimitado	No Considerado	Dependiente	Taillard (1999), Gendreau et al. (1999), Wassan y Osman (2002), Choi y Tcha (2007)
FSMF	Ilimitado	Considerado	Independiente	Gheysens et al. (1984), Golden et al. (1984), Gheysens et al. (1986), Desrochers y Verhoog (1991), Salhi y Rand (1993), Osman y Salhi (1996), Taillard (1999), Ochi et al. (1998a), Ochi et al. (1998b), Gendreau et al. (1999), Liu y Shen (1999), Wassan y Osman (2002), Dullaert et al. (2002), Renaud y Boctor (2002), Choi y Tcha (2007), Yaman (2006), Dell' Amico et al. (2006)

Tabla 3.1 Variantes del HVRP presentadas en literatura

En la tabla anterior se utilizaron acrónimos, los cuales se definen en seguida:

HVRPFD Problema de ruteo de vehículos heterogéneo con costos fijos

y costos de ruteo dependientes del vehículo.

HVRPD	Problema de ruteo de vehículos heterogéneo con costos de ruteo dependientes del vehículo.
SDVRP	Problema de ruteo de vehículos con costos de ruteo dependiente del lugar.
FSMFD	Problema de ruteo de vehículos con el tamaño y composición de la flotilla mixta y con costos fijos y costos de ruteo dependientes del vehículo.
FSMD	Problema de ruteo de vehículos con el tamaño y composición de la flotilla mixta y con costos de ruteo dependientes del vehículo.
FSMF	Problema de ruteo de vehículos con el tamaño y composición de la flotilla mixta y con costos fijos.

En la tabla 3.2 se enlistan los métodos más relevantes desarrollados para el problema de ruteo de vehículos heterogéneo y más adelante se habla un poco de ellos.

Método	Autores	Año	Basado en
PeGA	Ochi et al.	1998	Algoritmo genético paralelo Búsqueda dispersa Criterio de descomposición en pétalos
PGA–PED	Ochi et al.	1998	Algoritmo genético Búsqueda dispersa Criterio de descomposición en pétalos
	Gendreau et al.	1999	Búsqueda tabú GENIUS Memoria adaptativa
	Tarantilis, C.D., Kiranoudis, C.T.	2001	Algoritmo BATA
MER	Prins, C.	2002	Heurística de los ahorros
	Lima et al.	2004	Algoritmo memético

			GENIUS $\lambda$ -Intercambio
	Pessoa et al.	2007	Algoritmo de ramificación-corte-y-valor (price)
	Li et al.	2007	Método record-to-record
	Brandão, J.	2009	Búsqueda tabú
VNS	Imran et al.	2009	Algoritmo de Dijkstra Algoritmo de barrido Movimiento 2-opt
	Prins, C.	2009	Algoritmo memético Algoritmo genético Búsqueda local
	Vaquerizo, B.	2009	Algoritmo memético
MAMP	Li et al	2010	Memoria adaptativa Encadenamiento de trayectorias Búsqueda tabú
	Baldacci et al.	2010	
TSAM	Euchi, J., Chabchoub, H.	2010	Búsqueda tabú Memoria adaptativa
	Brandão, J.	2011	Búsqueda tabú

Tabla 3.2 Métodos relevantes de solución para el HVRP

El algoritmo PeGA (Petal Genetic Algorithm) propuesto por Ochi et al (1998) es un híbrido que combina un algoritmo genético, la búsqueda dispersa y el criterio de descomposición en pétalos para construir cromosomas. Básicamente construye las rutas utilizando un criterio de descomposición en pétalos. Genera una población inicial de cromosomas, las reproduce y evalúa. Al final del proceso, utiliza las técnicas de diversificación.

En el mismo año, los mismos autores presentaron PGA–PED (Parallel Genetic Algorithm with Petal Decomposition), el cual básicamente sigue los mismos pasos anteriores, solo que utiliza la forma paralela de un algoritmo genético.

Posteriormente, Gendreau et al. (1999) aplicaron la búsqueda tabú para dar solución al problema de ruteo de vehículos heterogéneo. El algoritmo que propusieron es complejo. Primeramente hace uso del algoritmo GENIUS que tiene una fase de construcción de ruta (GENI – Generalized Insertion) y una fase de mejora (US – Unstringing Stringing). Además, el algoritmo tiene integrado un procedimiento de memoria adaptativa.

Tarantilis y Kiranoudis (2001) desarrollaron un heurístico para resolver el problema de ruteo de vehículos heterogéneo con tamaño de la flota fijo. BATA (Backtracking Adaptative Threshold Accepting algorithm) es un algoritmo de retroceso adaptativo basado en umbral de aceptación. La innovación de BATA radica en que el umbral no solo se reduce, sino que también crece o decrece cuando el criterio de aceptación no se satisface.

Prins (2002) desarrolló la heurística de unión MER (Merge heuristic), la cual basa sus principios en el algoritmo de los ahorros de Clarke y Wright (1964). Cada vez que MER forma una ruta, esta es mejorada mediante un procedimiento de búsqueda local 2–opt.

Lima et al. (2004) desarrollaron un algoritmo genético híbrido (también conocido como algoritmo memético). Este algoritmo combina la heurística GENIUS (antes mencionada) y el operador  $\lambda$ –Intercambio.

Pessoa et al. (2007) presentaron un algoritmo de ramificación–corte–y–valor (price) para el problema de ruteo de vehículos heterogéneo.

En donde las columnas se asocian con  $q$ -rutas, y la relajación de las capacidades de los camiones en las rutas hace que el sub-problema de valoración pueda ser resuelto en tiempo pseudo-polinomial. Además se propusieron nuevas familias de corte, las cuales no incrementan la complejidad del sub-problema de valoración.

Li et al. (2007) realizaron una revisión de los métodos HCG (Heuristic Column Generation) propuesto por Taillard (1999), LBTA (List-Based Threshold Accepting algorithm) y BATA (Backtracking Adaptative Threshold Accepting algorithm) propuestos por Tarantilis et al. (2003, 2004). Así mismo, desarrollaron el método HRTR (Heterogeneous Record-To-Record algorithm).

Brandão (2009) desarrolló un algoritmo basado en la búsqueda tabú para dar solución al problema de ruteo de vehículos con tamaño y composición de la flota mixta.

Imran et al. (2009) abordaron este problema mediante una búsqueda de vecindarios variable VNS (Variable Neighborhood Search). La solución inicial es obtenida mediante la implementación del algoritmo de Dijkstra el cual tiene como bases al algoritmo de barrido y al movimiento 2-opt.

Prins (2009) presentó dos algoritmos meméticos (también conocidos como algoritmos genéticos híbridos) los cuales resuelven con éxito tanto al problema de ruteo de vehículos heterogéneo como al problema de ruteo de vehículos con flota mixta.

Vaquerizo (2009) desarrolló un algoritmo memético para el problema de ruteo de vehículos heterogéneo para cualquier problema de ruteo entre muchos orígenes (depósitos) y muchos destinos (clientes).

Xiangyong et al. (2010) propusieron un algoritmo de memoria adaptativa multiarranque (MAMP por sus siglas en inglés, Multistart Adaptive Memory Programming) y un algoritmo de encadenamiento de trayectorias.

Baldacci et al. (2010) realizaron una recopilación de los algoritmos que han causado impacto en el estado del arte correspondiente a los algoritmos exactos para el problema de ruteo de vehículos bajo restricciones de capacidad, enfocándose en el problema de ruteo de vehículos capacitado y el problema de ruteo de vehículos heterogéneo.

Euchi y Chabchoub (2010) desarrollaron un procedimiento de búsqueda tabú integrado con memoria adaptativa TSAM (Tabu Search embedded with Adaptive Memory) para darle solución al problema de ruteo de vehículos heterogéneo con flota fija.

Brandão (2011) desarrolló otro enfoque basado en búsqueda tabú para darle solución al problema de ruteo de vehículos heterogéneo con flota fija.



### CAPÍTULO 3

# DESCRIPCIÓN Y MODELACIÓN DEL PROBLEMA DE RUTEO DE VEHÍCULOS HETEROGÉNEO

En este capítulo se describirá el problema de ruteo de vehículos con flota heterogénea, así como su modelo matemático. El modelo matemático utilizado para las experimentaciones de esta tesis es el propuesto por Baldacci et al. (2007).

El problema de ruteo de vehículos heterogéneo (HVRP por sus siglas en inglés) consiste en diseñar un conjunto de rutas empleando una flotilla de vehículos de diferente capacidad hacia un conjunto de clientes distribuidos geográficamente.

Formalmente el HVRP se define como:

Sea  $G=(V, A)$  un grafo dirigido,  $V = \{0, 1, \dots, n\}$  es el conjunto de  $n+1$  nodos y  $A$  es el conjunto de arcos. El nodo 0 representa al depósito, mientras que el subconjunto  $V' = V \setminus \{0\}$  restante corresponde a los  $n$  clientes. Cada cliente  $i \in V'$  tiene asociada una demanda  $d_i$ . La flotilla de vehículos está compuesta por  $m$  tipos de vehículos diferentes, con  $M = \{1, \dots, m\}$ . Para cada tipo  $k \in M$ ,  $m_k$  vehículos están disponibles en el depósito, cada uno con capacidad  $Q_k$ . A cada tipo de vehículo se le asigna

un costo fijo no negativo  $F_k$ . Además, por cada arco  $(i, j) \in A$  y cada vehículo  $k \in M$  se tiene un costo de ruteo no negativo  $c_{ij}^k$  asociado.

Entonces, una ruta se define como el par  $(R, k)$ , donde  $R = (i_1, i_2, \dots, i_{|R|})$ , con  $i_1 = i_{|R|} = 0$  y  $\{i_2, \dots, i_{|R|-1}\} \subseteq V'$ , es un circuito simple en  $G$  el cual contiene al depósito, y  $k$  es el tipo de vehículo asociado a la ruta. Una ruta  $(R, k)$  es factible si la demanda total de los clientes visitados por la ruta no excede la capacidad del vehículo  $Q_k$ . El costo de la ruta corresponde a la suma de los costos de los arcos que conforman la ruta, más el costo fijo asociado a cada vehículo.

La versión más general del HVRP consiste en diseñar un conjunto de rutas factibles de mínimo costo, tales que:

- i. cada cliente sea visitado exactamente por una ruta;
- ii. el número de rutas realizadas por los vehículos de tipo  $k \in M$  no sea más grande que  $m_k$ .

### 3.1 PARÁMETROS Y CONJUNTOS

El problema de ruteo de vehículos heterogéneo tiene los siguientes parámetros:

$n$ : Número de clientes a servir.

$V = \{0, 1, \dots, n\}$ : Conjunto de índices de los nodos en la red, donde el 0 corresponde al depósito y el subconjunto  $V' = V \setminus \{0\}$  es el conjunto de los índices correspondientes a los clientes.

$d_i$ : Demanda del cliente  $i$ .

$c_{ij}^k$ : Costo por utilizar el arco  $(i, j)$  con el vehículo tipo  $k$ .

$M = \{1, \dots, m\}$ : Conjunto de tipos de vehículos disponibles en el depósito

$m_k$ : Número de vehículos del tipo  $k$  disponibles, donde  $k \in M$ .

$Q_k$ : Capacidad del vehículo del tipo  $k$ .

$F_k$ : Costo fijo asociado a cada vehículo de tipo  $k$ .

## 3.2 VARIABLES DEL PROBLEMA

Para la modelación de un problema de ruteo de vehículos con flota heterogénea se necesitan dos tipos de variables. Las variables de decisión que nos dicen que arco toman los vehículos en la solución final y las variables auxiliares, que estas miden cuanta carga lleva el vehículo.

### 3.2.1 VARIABLES DE DECISIÓN

La variable  $x_{ij}^k$  es una variable binaria que indica si el vehículo  $k$  viaja directamente del cliente  $i$  al cliente  $j$ , tomando los siguientes valores:

$$x_{ij}^k = \begin{cases} 1, & \text{si el arco } (i, j) \text{ es visitado por el vehículo } k \\ 0. & \text{en caso contrario} \end{cases}$$

### 3.2.2 VARIABLES AUXILIARES

$y_{ij}$ : Variable auxiliar que especifica la cantidad de bienes que lleva un vehículo cuando deja al cliente  $i$  para ir a servir al cliente  $j$ .

### 3.3 OBJETIVO DEL PROBLEMA

Como objetivo principal nos hemos planteado es minimizar el costo de la solución final,

$$\text{Min } z = \sum_{k \in M} \sum_{j \in V'} F_k x_{0j}^k + \sum_{k \in M} \sum_{\substack{i, j \in V \\ i \neq j}} c_{ij}^k x_{ij}^k$$

### 3.4 RESTRICCIONES

1. Cada cliente es visitado exactamente una vez.

$$\sum_{k \in M} \sum_{i \in V} x_{ij}^k = 1 \quad \forall j \in V' \quad (3.1)$$

2. Si en la ruta del vehículo  $k$  se visita al cliente  $p$ , ese mismo vehículo tiene que dejarlo.

$$\sum_{i \in V} x_{ip}^k - \sum_{j \in V} x_{pj}^k = 0 \quad \forall p \in V', \forall k \in M \quad (3.2)$$

3. En esta restricción se impone el número máximo de vehículos del tipo  $k$  que están disponibles para salir del depósito.

$$\sum_{j \in V'} x_{0j}^k \leq m_k \quad \forall k \in M \quad (3.3)$$

4. Este conjunto de restricciones son las de flujo de mercancía, esto es, que especifican que la diferencia entre la cantidad de bienes que lleva el vehículo antes de visitar a un cliente  $j$  y después de dejarlo sea igual a la demanda de ese mismo cliente  $j$ .

$$\sum_{i \in V} y_{ij} - \sum_{i \in V} y_{ji} = d_j \quad \forall j \in V' \quad (3.4)$$

5. Y por ultimo esta restricci3n asegura que la capacidad del veh3culo no sea excedida y evita la formaci3n de subtours.

$$d_j x_{ij}^k \leq y_{ij} \leq (Q_k - d_i) x_{ij}^k \quad \forall i, j \in V, i \neq j, \forall k \in M \quad (3.5)$$

Finalmente el modelo completo se muestra a continuaci3n:

$$\text{Min } z = \sum_{k \in M} \sum_{j \in V'} F_k x_{0j}^k + \sum_{k \in M} \sum_{\substack{i, j \in V \\ i \neq j}} c_{ij} x_{ij}^k$$

Sujeto a:

$$\sum_{k \in M} \sum_{i \in V} x_{ij}^k = 1 \quad \forall j \in V' \quad (3.1)$$

$$\sum_{i \in V} x_{ip}^k - \sum_{j \in V} x_{pj}^k = 0 \quad \forall p \in V', \forall k \in M \quad (3.2)$$

$$\sum_{j \in V'} x_{oj}^k \leq m_k \quad \forall k \in M \quad (3.3)$$

$$\sum_{i \in V} y_{ij} - \sum_{i \in V} y_{ji} = d_j \quad \forall j \in V' \quad (3.4)$$

$$d_j x_{ij}^k \leq y_{ij} \leq (Q_k - d_i) x_{ij}^k \quad \forall i, j \in V, i \neq j, \forall k \in M \quad (3.5)$$

## CAPÍTULO 4

# METODOLOGÍA PROPUESTA PARA EL PROBLEMA DE RUTEO DE VEHÍCULOS HETEROGÉNEO

Se propondrá un nuevo método para dar solución al problema de ruteo de vehículos heterogéneo.

En esencia, estamos proponiendo un método de aproximación sucesiva (SAM, por sus siglas en inglés), el cual utiliza como sub-rutina el SR-GCWS-CS propuesto por Juan et al. (2010). El SR-GCWS-CS, como se mencionó anteriormente, combina el método de los ahorros con la simulación de Monte Carlo utilizando las técnicas de memoria cache y de *splitting* para dar solución al problema de ruteo de vehículos capacitado.

### 4.1 METODOLOGÍA

SAM recibe como entradas:

- los nodos a ser servidos,
- el conjunto de restricciones,
- la matriz de costos, y
- el número de iteraciones a correr.

Formalmente el proceso de SAM es el siguiente:

- 
- // Repetir hasta que todos los nodos sean servidos //*
1. Ordenar los vehículos de forma no decreciente por medio del método *SortVehiclesByCap()*
  - // Resolver un CVRP asociado con cada capacidad //*
  2. **Por** cada capacidad de vehículos *cap* **hacer**
  3. Obtener *subSol* mediante el procedimiento *solveCVRP(vCap)*;  
*// Guardar las rutas de acuerdo al número de vehículos disponibles con esa capacidad cap //*
  4. De los *n* vehículos, se obtiene la cantidad de vehículos disponibles *nVehicles* con capacidad *cap* con el procedimiento *getVehiclesCountByCap(cap)*;
  5. Actualizar *subSol* con el número de rutas obtenidas a partir de *nVehicles* con el método *subSol.extracRoutes(nVehicles)*;
  6. Agregar las rutas guardadas a la solución con el procedimiento *newSol.addNewRoutes(subSol)*;
  7. Eliminar los nodos servidos en la *subSol*;
  8. **Si** todos los nodos han sido servidos **Parar**
  9. **Fin Si.**
  10. **Fin Por.**
- 

Nota: La programación del algoritmo SAM se muestra en el Apéndice II.

Básicamente trabaja de la siguiente forma: resuelve un problema de ruteo de vehículos capacitado como si todos los vehículos fueran de la mayor capacidad que se tiene disponible en la flotilla. Después ordena las rutas de mayor a menor demanda y el algoritmo guarda el número de rutas dependiendo del número de vehículos de mayor capacidad que se tiene disponible. Esto es por ejemplo, si en la solución parcial nos quedan 6 rutas, de las cuales las demandas en cada ruta son 48, 50, 38, 29, 27 y

25 respectivamente, y tenemos una flotilla de vehículos con las siguientes capacidades: 2 con capacidad 50, 1 con capacidad 40 y 2 con capacidad 30, entonces el algoritmo guarda las dos primeras rutas, que son las que tienen mayor demanda. En este punto, obtenemos un sub-problema eliminando los vehículos utilizados y los nodos servidos. Entonces, resolvemos este sub-problema de la misma forma solo que esta vez trabajamos sobre los nodos restantes. Al final, cuando todos los nodos pertenecen a alguna ruta, obtenemos la solución final dada por todas las rutas guardadas. En la Figura 4.1 se muestra el proceso de forma gráfica:



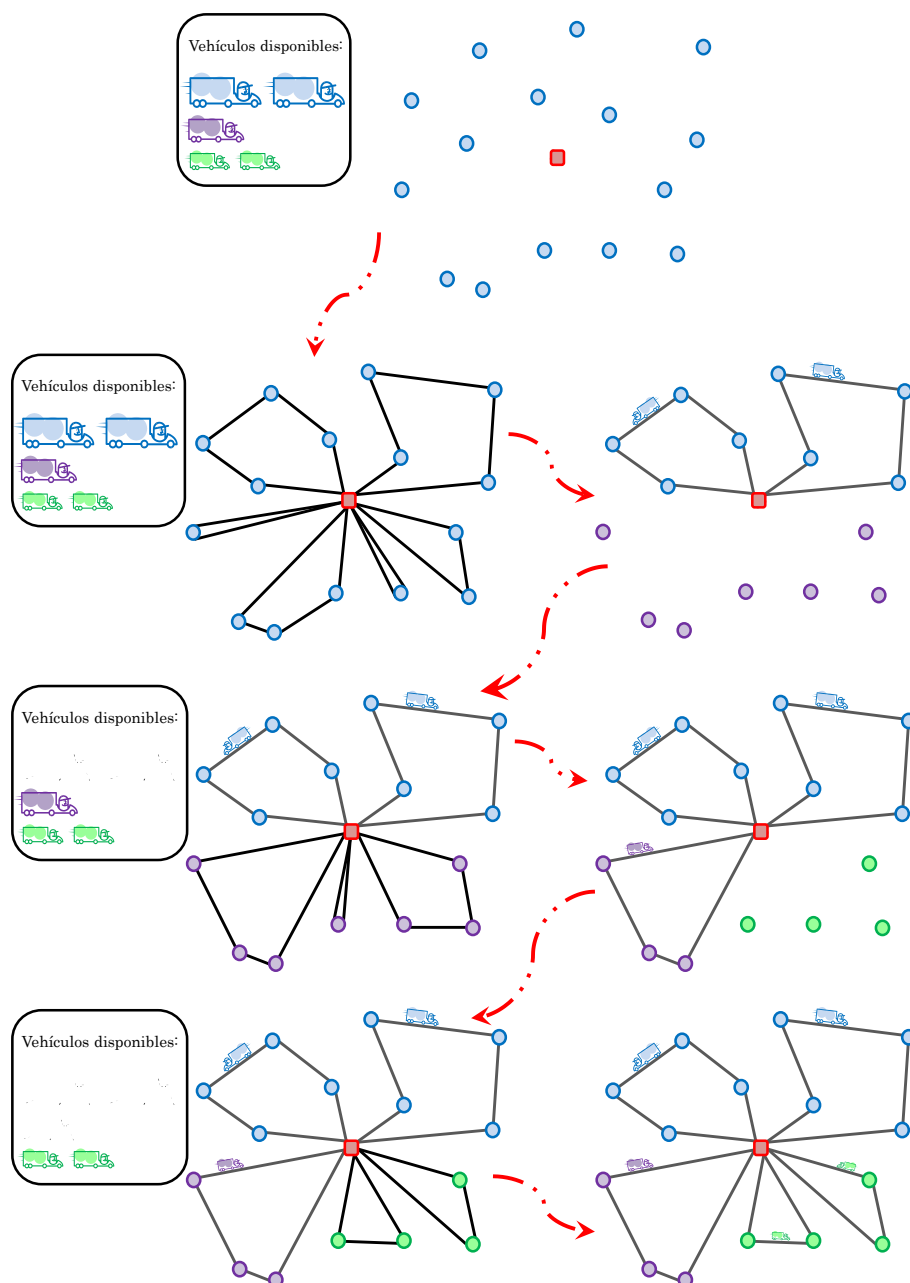


Figura 4.1 Ejemplificación del método SAM

#### 4.1.1 MÉTODO DE SOLUCIÓN PARA EL PROBLEMA DE RUTEO DE VEHÍCULOS CAPACITADO (SOLVECVRP(vCAP))

El procedimiento *solveCVRP(vCap)* es la base del algoritmo y este primeramente calcula la matriz de ahorros, y se construye la lista de ahorros y la ordena.

Después, se obtiene una solución inicial aplicando la heurística de los ahorros en su versión paralela (el algoritmo de los ahorros se explica en el apartado A) Heurísticas constructivas del Capítulo 2). El costo asociado a esta solución será usado como un límite superior para aquellos costos que consideraremos como una buena solución. Ahora da comienzo el proceso iterativo para generar las nuevas soluciones.

Una solución nueva es construida usando la heurística de los ahorros aleatorizado, luego esta solución se procesa por el procedimiento de cache (explicado más adelante). Si esta solución resultante supera a la solución de la heurística de los ahorros, es considerada prometedora se guarda en la memoria cache y se empieza una nueva iteración. Y se repite el proceso hasta que se satisface un criterio de parada dado.

El algoritmo es el siguiente:

- 
1. Inicializar  $cwsSol = \infty$ ,  $vrpSol = \infty$  y  $bestSols = \infty$ , e introducir los parámetros *Node*, *Constraints*, *rng* (generador de números aleatorios), *nIter* (número de iteraciones), *costMatrix* y *nSol* (número de soluciones a guardar).
  2. Se crea la lista de ahorros *savingList* con los parámetros *Node* y *costMatrix*.
  3. Construcción de *cwsSol* con los parámetros *Node*, *costMatrix*, *savingList*, y *Constraints*.
  4. **Mientras** no se satisfaga el criterio de terminación *nIter*, **hacer**
  5. Construir *hvrpSol* mediante el procedimiento *construcRandomSol* con los parámetros *Node*, *costMatrix*, *savingList*, *Constraints* y *rng*.

6. Mejorar *hvrpSol* con el método *improveSolUsingRoutesCache* usando la *hvrpSol* y la *costMatrix*.
  7. **Si** *hvrpSol* es mejor que *cwsSol*, entonces
  8. Actualizar *bestSols*.
  9. **Fin Si**.
  10. **Fin Mientras**.
  11. Devolver *bestSols*.
  12. Fin
- 

#### CONSTRUCCIÓN DE LA SOLUCIÓN DE REFERENCIA (CWSOL)

Esta solución es la que obtenemos por medio del algoritmo clásico de los ahorros de Clarke y Wright (1964) en su forma paralela.

Entonces *cwsSol* se construye a partir de las coordenadas  $(x_i, y_i)$  de los  $n$  nodos o clientes y de las coordenadas del depósito  $(x_0, y_0)$  y de la matriz de costos *costMatrix*, además de tomar en cuenta la demanda  $d_i$  de cada cliente y la capacidad del vehículo  $k$  con mayor capacidad.

Primero que nada, se asigna una ruta a cada cliente y posteriormente se construye la lista de ahorros *savingList*. Esta lista se calcula de la siguiente forma: dadas dos rutas parciales factibles se calcula el ahorro  $s_{ij} = c_{i0} + c_{0j} - c_{ij}$ , donde  $s_{ij}$  es un valor mayor a cero.

Ahora, las uniones de las rutas se basan en esta lista y se van formando empezando siempre por las uniones que generan un ahorro más grande y finalizando con las que generan un ahorro menor. Entonces el algoritmo sigue haciendo uniones de rutas mientras no se sobrepasen las capacidades de los vehículos.

Por ejemplo, en la Figura 4.2, del lado izquierdo se muestran tres rutas actuales generadas en una iteración del algoritmo, y supongamos que el algoritmo al final de su ejecución nos arroja la solución mostrada en el lado derecho de la figura, en donde vemos como se han unido las rutas, quedándonos dos rutas en la solución final.

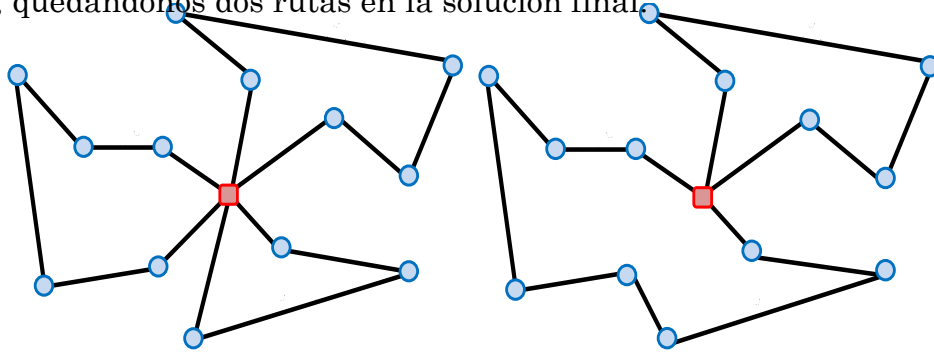


Figura 4.2 Unión de rutas

#### MÉTODO DE CONSTRUCCIÓN DE LA SOLUCIÓN ALEATORIZADA (HVRPSOL)

Una de las principales ideas de este algoritmo es introducir un comportamiento aleatorio parcial dentro de la heurística de los ahorros para realizar un proceso de búsqueda dentro del espacio de soluciones factibles. Cada solución factible es un conjunto de rutas las cuales tienen como origen y destino el depósito, de tal forma que se satisfagan todas las demandas de los clientes, los cuales son visitados solo una vez.

Ahora bien, la obtención de la solución aleatorizada *vrpSol* sigue básicamente los mismos pasos que la obtención de la *cwsSol*, solo que cuando selecciona qué unión hacer, no lo hace eligiendo siempre la de mayor ahorro (comportamiento *greedy*) si no que esta vez se asigna una probabilidad de selección a cada unión potencial en la *savingList*. Dicha probabilidad es coherente con el ahorro generado con dicha unión, esto es, que las uniones que generen un mayor ahorro son más propensas a ser elegidas que las que generen un ahorro menor.

Entonces, durante el proceso de construcción de la *vrpSol* cada vez que se selecciona una unión de la lista de uniones disponibles, se selecciona aleatoriamente un valor  $\alpha$  de una distribución uniforme en  $(a,b)$ , donde  $0 < a \leq b < 1$ . Este parámetro  $\alpha$  define a la distribución geométrica que será utilizada para asignar las probabilidades decrecientes de manera exponencial a cada unión elegible de acuerdo a su posición en la *savingList*. Como se mencionó, las uniones con mayor ahorro son más propensas a ser elegidas, pero la probabilidad exacta asignada a ellas va a depender de la distribución en concreto que se seleccionó en cada paso.

#### MÉTODO DE MEJORA BASADO EN LA MEMORIA CACHE (BESTSOLS)

Este proceso iterativo anterior ha sido enriquecido con un simple mecanismo de aprendizaje, en cual logra una mayor velocidad inicial de convergencia a soluciones óptimas locales.

La idea básica de este mecanismo de aprendizaje se describe como: para cada ruta generada, el mejor orden conocido para viajar entre los nodos que conforman esa ruta se guarda en la memoria cache.

La cache es actualizada constantemente cuando se encuentra un orden mejor con un coste menor para un conjunto de nodos dado. Al mismo tiempo, durante el proceso de construcción las rutas contenidas en la cache son reutilizadas cuando sea posible mejorar soluciones nuevas generadas.

Entonces, como la memoria cache guarda el mejor orden para visitar a un conjunto de nodos con un solo vehículo, la mayoría de las nuevas soluciones se pueden beneficiar con esta información.

## CAPÍTULO 5

# EXPERIMENTACIÓN COMPUTACIONAL

En este capítulo presentaremos algunos de los resultados obtenidos mediante la implementación de la metodología propuesta.

En el apartado 5.1 se describirán las instancias utilizadas en los experimentos. En las secciones 5.2 y 5.3 se presentan los resultados más relevantes de la experimentación del método propuesto. Y finalmente, en el apartado 5.4 se presentan las comparaciones entre SAM y otros métodos presentados en literatura.

## 5.1 DESCRIPCIÓN DE LAS INSTANCIAS

La experimentación computacional se llevó a cabo con dos tipos de instancias, aquellas que fueron diseñadas especialmente para el problema de ruteo de vehículos heterogéneo, y aquellas que se diseñaron para el problema de ruteo de vehículos capacitado y se modificaron para adaptarse a nuestro problema.

### 5.1.1 INSTANCIAS QUE SE ADECUARON AL PROBLEMA DE RUTEO DE VEHÍCULOS HETEROGÉNEO

Como existen muy pocas instancias diseñadas en específico para el HVRP, se tomaron las instancias del grupo A y del grupo B (Augerat et

al., 1995) y se han adecuado para poder trabajar con ellas, la fuente de donde se tomaron las instancias está en la bibliografía.

Antes de empezar a modificarlas, obtuvimos la información que nos proporcionan ambos tipos de instancias, la cual es el número de nodos y su demanda asociada, las coordenadas en X y las coordenadas en Y para cada nodo, la capacidad de los vehículos (que en este caso es la misma para todos) y el número mínimo de vehículos para resolverlo, así como la mejor solución encontrada hasta el momento.

#### INSTANCIAS DEL GRUPO A

Este conjunto de instancias se caracteriza por que tanto la ubicación de los clientes como su demanda fueron generadas aleatoriamente siguiendo una distribución uniforme, y el depósito se encuentra localizado en el origen. El tamaño de estas instancias varía entre los 31 a los 79 clientes. La capacidad de los vehículos es de 100.

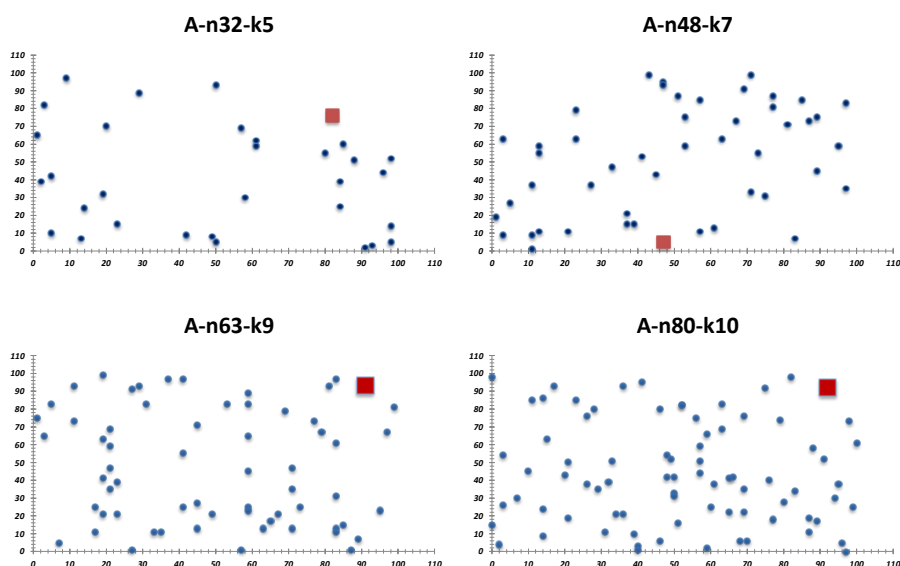


Figura 5.1 Algunas distribuciones de nodos del conjunto A

## INSTANCIAS DEL GRUPO B

La principal característica de este conjunto de instancias es que los clientes se encuentran agrupados en zonas localizadas, el tamaño varía entre los 30 a los 77 clientes y el depósito se localiza en la parte inferior al centro. La capacidad de los vehículos es de 100.

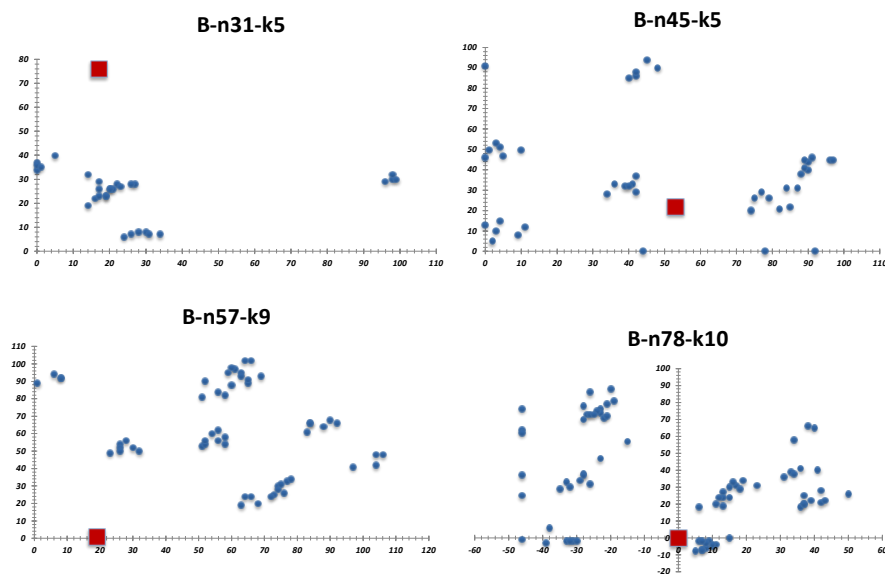


Figura 5.2 Algunas distribuciones de nodos del conjunto B

## MODIFICACIÓN DE INSTANCIAS

Ahora bien, la modificación a las instancias fue de la siguiente forma: primero resolvimos cada instancia utilizando el algoritmo de los ahorros. Una vez que obtuvimos las rutas correspondientes, de acuerdo a la demanda en cada una de ellas, se fijó la cantidad de vehículos de cada tipo que utilizaríamos más tarde.

Por ejemplo, para la instancia A-n32-k5, el algoritmo de los ahorros arrojó 5 rutas, lo cual nos da el tamaño de la flotilla. La configuración o el número de vehículos de cada tipo lo obtenemos mediante la demanda en cada ruta. Para esta instancia en específico obtenemos que nuestra



flotilla de vehículos es de tamaño 5, con tres tipos diferentes de capacidad, esto es, tenemos un vehículo con capacidad 50, dos con capacidad 90 y tres con capacidad 100, en la Tabla 5.1 se ejemplifica lo anterior:

Número de ruta	Demanda	Capacidad ajustada del vehículo
1	90	90
2	99	100
3	92	100
4	38	50
5	91	100

Tabla 5.1 Ejemplo de modificación de una instancia

De esta forma se modificaron un total de 27 instancias para el grupo A y 23 para el grupo B. En estos dos grupos de instancias el costo fijo no se tomó en cuenta, es decir, que se fijó en cero.

### 5.1.2 INSTANCIAS ESPECIALES PARA EL PROBLEMA DE RUTEO DE VEHÍCULOS HETEROGÉNEO

Estas instancias pertenecen al grupo T, diseñadas por Taillard (1999). La información que nos proporcionan es: el número de nodos y su demanda correspondiente, las coordenadas en X y en Y de cada nodo, el tamaño de la flota, cuantos vehículos de cada tipo hay y sus capacidades, y el costo fijo asociado al tipo de los vehículos.

En general hay tres tipos de instancias, de acuerdo al número de nodos que las componen, con 51 nodos tenemos de la T13 a la T16, con 76 nodos están la T17 y la T18, y con 101 nodos están la T19 y la T20. Cada

una de ellas con un tamaño de flotilla de vehículos diferente, a excepción de la T15 y T16, en las cuales su flotilla de vehículos está compuesta por nueve vehículos, pero la diferencia entre una y otra son las capacidades de los mismos.

Cabe mencionar que la diferencia entre los siguientes pares de instancias: T13 – T14, T15 – T16, T17 – T18, y T19 – T20, está en las características de la flotilla de vehículos, ya que los nodos tienen la misma configuración, tanto en coordenadas  $(x, y)$  como en la demanda de los nodos.

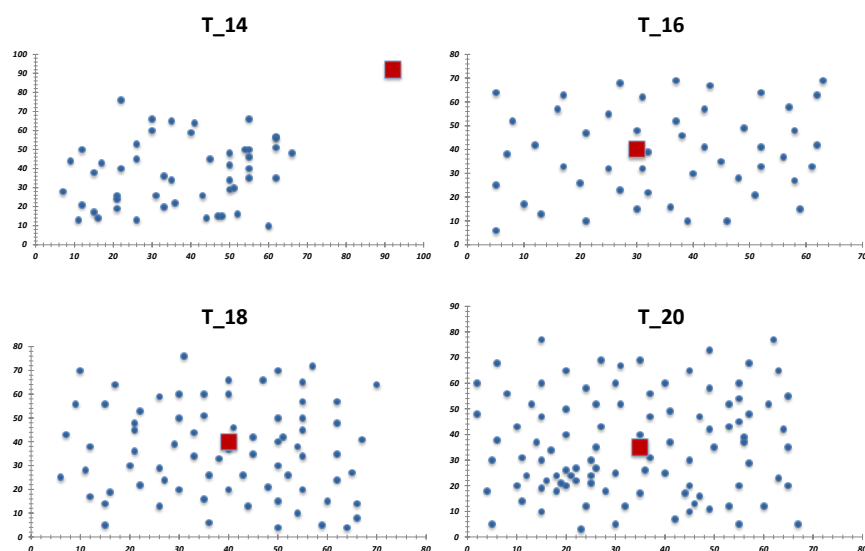


Figura 5.3 Algunas distribuciones de nodos del conjunto T

Se corrieron un total de 58 instancias, de las cuales, las cuales se describen a continuación. Cabe mencionar que para las instancias de los grupos A y B el costo fijo es cero.

### 5.1.3 INSTANCIAS DEL GRUPO A

En la Tabla 5.2 se muestran la configuración final de las instancias del grupo A, donde  $Q_x$  representa la capacidad del vehículo del tipo  $x$ ,  $m_x$  representa la cantidad de vehículos del tipo  $x$  disponibles, donde  $x = A, B, C, D, E$ , y  $n$  representa el número de nodos en el grafo.

Instancia	n	Tipo de vehículo									
		A		B		C		D		E	
		$Q_A$	$m_A$	$Q_B$	$m_B$	$Q_C$	$m_C$	$Q_D$	$m_D$	$Q_E$	$m_E$
A-n32-k5	32	50	1	90	2	100	2				
A-n33-k5	33	80	1	90	2	100	2				
A-n33-k6	33	40	1	80	1	90	3	100	2		
A-n34-k5	34	70	1	50	2	90	1	100	2		
A-n36-k5	36	60	1	90	1	100	3				
A-n37-k5	37	50	1	80	1	90	1	100	2		
A-n37-k6	37	90	4	100	2						
A-n38-k5	38	60	1	79	1	90	2	100	2		
A-n39-k5	39	90	2	100	3						
A-n39-k6	39	80	2	90	1	90	1				
A-n44-k7	44	50	1	70	1	80	1	90	2	100	2
A-n45-k6	45	60	1	80	1	90	2	100	3		
A-n45-k7	45	80	2	90	1	100	4				
A-n46-k7	46	70	1	80	2	90	2	100	2		
A-n48-k7	48	70	1	80	1	90	2	100	3		
A-n53-k7	53	90	3	100	4						
A-n54-k7	54	90	3	100	4						
A-n55-k9	55	80	1	90	3	100	5				
A-n60-k9	60	70	1	80	1	90	2	100	5		
A-n61-k9	61	40	1	80	1	90	3	100	5		
A-n62-k8	62	70	1	80	1	90	1	100	5		
A-n63-k9	63	40	1	80	1	90	3	100	5		
A-n63-k10	63	80	2	90	3	100	5				
A-n64-k9	64	50	1	70	1	80	2	90	3	100	3
A-n65-k9	65	50	1	70	1	80	1	90	2	100	5
A-n69-k9	69	80	1	90	3	100	5				
A-n80-k10	80	70	1	90	2	100	7				

Tabla 5.2 Descripción de instancias del grupo A

## 5.1.4 INSTANCIAS DEL GRUPO B

En la Tabla 5.3 se muestran la configuración final de las instancias del grupo B, donde  $Q_x$  representa la capacidad del vehículo del tipo  $x$ ,  $m_x$  representa la cantidad de vehículos del tipo  $x$  disponibles, donde  $x = A, B, C, D, E$ , y  $n$  representa el número de nodos en el grafo.

Instancia	n	Tipo de vehículo									
		A		B		C		D		E	
		$Q_A$	$m_A$	$Q_B$	$m_B$	$Q_C$	$m_C$	$Q_D$	$m_D$	$Q_E$	$m_E$
B-n31-k5	31	60	1	90	2	100	2				
B-n34-k5	34	80	1	90	2	100	2				
B-n35-k5	35	60	1	90	2	100	2				
B-n38-k6	38	60	1	80	2	90	1	100	2		
B-n39-k5	39	70	1	80	1	90	1	100	2		
B-n41-k6	41	60	2	70	1	80	1	100	3		
B-n43-k6	43	60	1	80	1	90	1	100	3		
B-n44-k7	44	70	1	90	2	100	4				
B-n45-k5	45	90	1	100	4						
B-n45-k6	45	60	1	80	2	90	2	100	2		
B-n50-k7	50	50	1	80	1	90	2	100	3		
B-n50-k8	50	70	1	90	3	100	4				
B-n51-k7	51	70	2	80	1	90	3	100	2		
B-n52-k7	52	50	1	80	1	90	2	100	3		
B-n56-k7	56	60	1	80	1	90	2	100	3		
B-n57-k7	57	60	1	70	1	80	1	90	1	100	4
B-n57-k9	57	60	1	80	1	90	3	100	4		
B-n63-k10	63	70	1	90	4	100	5				
B-n64-k9	64	70	2	80	2	90	2	100	4		
B-n66-k9	66	70	1	80	2	90	5	100	2		
B-n67-k10	67	50	1	60	2	80	2	90	2	100	4
B-n68-k9	68	70	1	80	1	90	1	100	6		
B-n78-k10	78	70	1	90	3	100	6				

Tabla 5.3 Descripción de instancias del grupo B

### 5.1.5 INSTANCIAS DEL GRUPO T

En la Tabla 5.4 se muestran las características de las instancias del grupo T, donde  $Q_x$  representa la capacidad del vehículo del tipo  $x$ ,  $f_x$  representa el costo fijo del vehículo del tipo  $x$ ,  $v_x$  representa el costo variable del vehículo del tipo  $x$ ,  $m_x$  representa la cantidad de vehículos del tipo  $x$  disponibles, donde  $x = A, B, C, D, E$ , y  $n$  representa el número de nodos en el grafo.

Instancia	n	Tipo de vehículo																							
		A				B				C				D				E				F			
		$Q_A$	$f_A$	$v_A$	$m_A$	$Q_B$	$f_B$	$v_B$	$m_B$	$Q_C$	$f_C$	$v_C$	$m_C$	$Q_D$	$f_D$	$v_D$	$m_D$	$Q_E$	$f_E$	$v_E$	$m_E$	$Q_F$	$f_F$	$v_F$	$m_F$
T_13	51	20	20	1.0	4	30	35	1.1	2	40	50	1.2	4	70	120	1.7	4	120	225	2.5	2	200	400	3.2	1
T_14	51	120	100	1.0	4	160	1500	1.1	2	300	3500	1.4	1												
T_15	51	50	100	1.0	4	100	250	1.6	3	160	450	2.0	2												
T_16	51	40	100	1.0	2	80	200	1.6	4	140	400	2.1	3												
T_17	76	50	25	1.0	4	120	80	1.2	4	200	150	1.5	2	350	320	1.8	1								
T_18	76	20	10	1.0	4	50	35	1.3	4	100	100	1.9	2	150	180	2.4	2	250	400	2.9	1	400	800	3.2	1
T_19	101	100	500	1.0	4	200	1200	1.4	3	300	2100	1.7	3												
T_20	101	60	100	1.0	6	140	300	1.7	4	200	500	2.0	3												

Tabla 5.4 Descripción de instancias del grupo T

## 5.2 RESULTADOS Y COMPARACIÓN DE METODOLOGÍAS

En este apartado mostraremos algunos de los resultados, haciendo comparaciones entre el método SAM y el algoritmo de los ahorros (CWS) para las instancias del conjunto A y B.

Para el conjunto de instancias del conjunto T se hacen comparaciones entre el método SAM y el algoritmo de los ahorros (CWS), así como entre el método SAM y la mejor solución en literatura (BS). Además se hacen otras dos comparaciones, ya que se reportan resultados donde solo se toman en cuenta los costos fijos (FC) y en otro caso solo se toman en cuenta los costos variables (VC).

Un criterio simple para medir la eficiencia y la calidad de nuestro método fue la comparación con las instancias que se reportan en literatura, y como se observa en las siguientes tablas, en la mayoría de las instancias se logro una buena aproximación e incluso en algunas se obtuvo un mejor resultado que el reportado en la literatura.

## RESULTADOS SAM VS CWS DE LAS INSTANCIAS DEL CONJUNTO A

En la Figura 5.4 se muestran los resultados obtenidos mediante la implementación del algoritmo de los ahorros, así como los obtenidos con el método SAM. Ambas flotas son heterogéneas. Podemos observar que en todas las instancias se mejoró el resultado obtenido por el método de los ahorros.

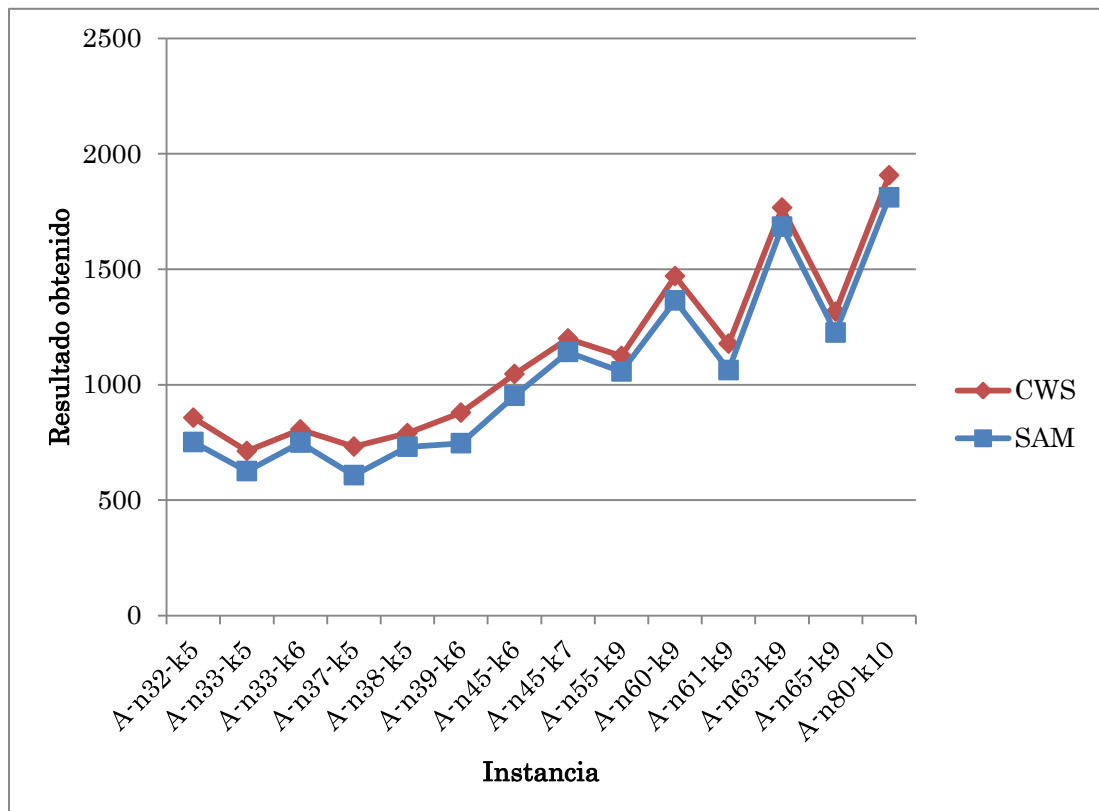


Figura 5.4 Comparación de resultados: SAM vs CWS del conjunto A

## RESULTADOS SAM VS CWS DE LAS INSTANCIAS DEL CONJUNTO B

De igual forma en la Figura 5.5 se muestran los resultados del CWS contra el método SAM, en donde ambas flotillas son heterogéneas, y en cada instancia se mejoro la solución encontrada por el algoritmo de los ahorros.

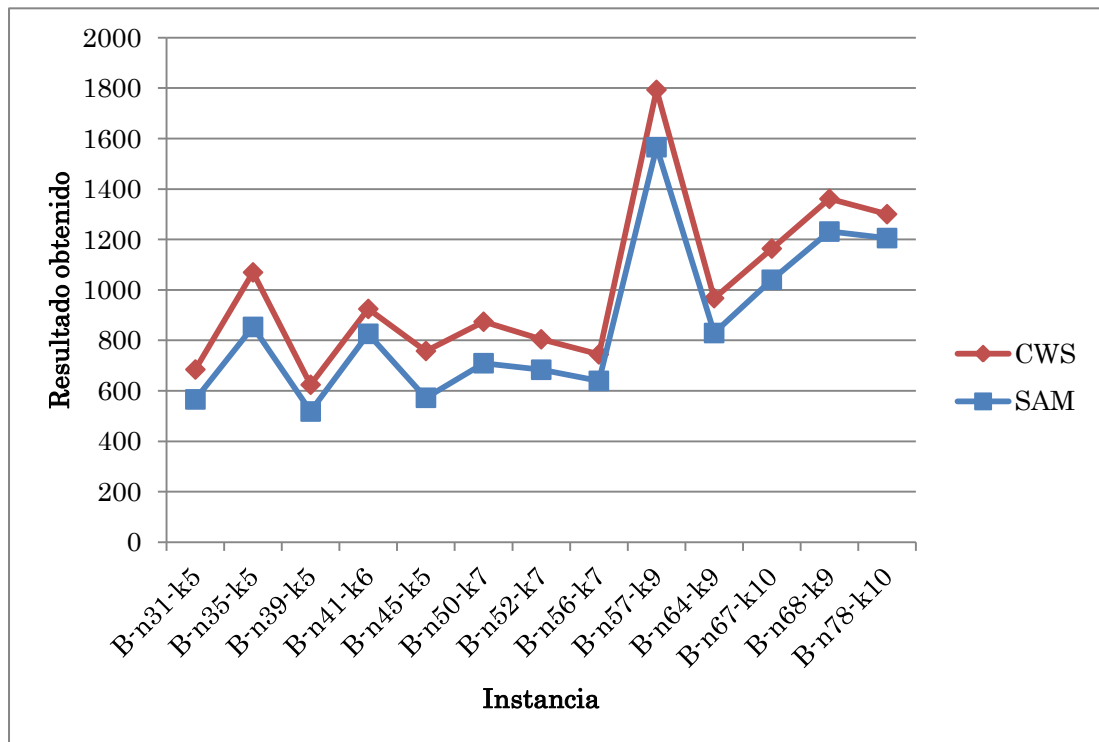


Figura 5.5 Comparación de resultados: SAM vs CWS del conjunto B



## RESULTADOS SAM VS CWS DE LAS INSTANCIAS DEL CONJUNTO T

Como se menciono antes, el conjunto de instancias T fueron diseñadas especialmente para el problema de ruteo de vehículos con flota heterogénea. De igual forma tenemos dos comparaciones, la primera, en la Figura 5.6, hecha tomando en cuenta los costos fijos (Fixed costs, FC) y comparándola con el algoritmo de los ahorros, donde observamos que, a excepción de la instancia T<sub>18</sub>, se mejoraron todos los resultados obtenidos por el CWS. Se compararon nuestros resultados con los reportados por Laporte et al. (1999) y Tarantilis et al. (2004).

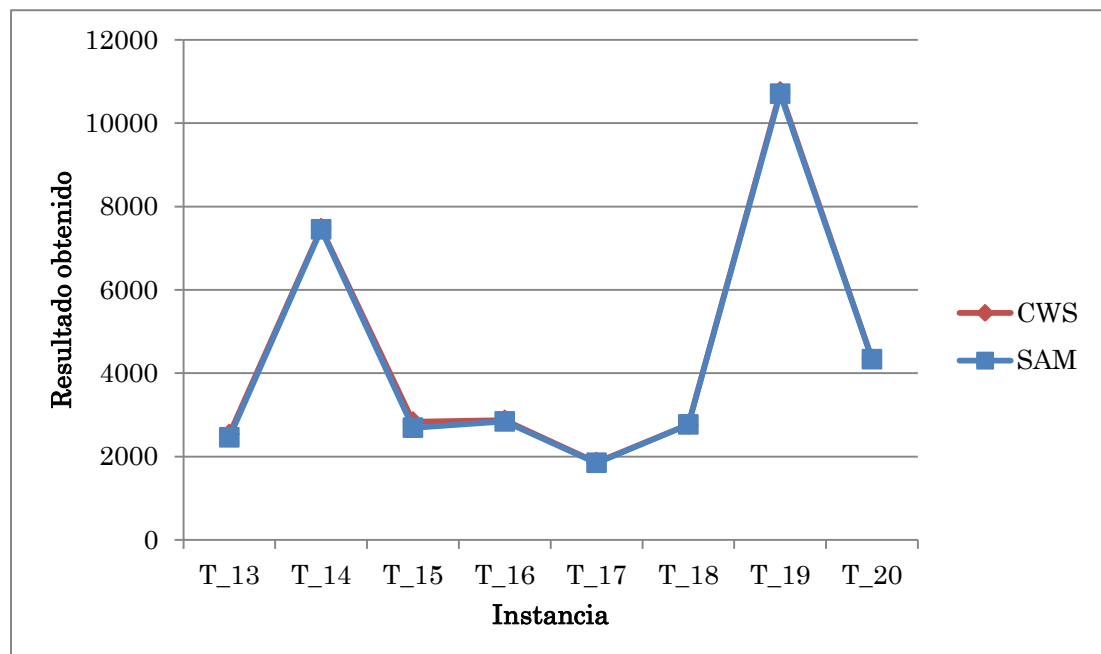


Figura 5.6 Comparación de resultados: SAM vs CWS del conjunto T (FC)

De igual forma se compararon los dos métodos tomando en cuenta los costos variables (Variable costs, VC), y en la Figura 5.7 se muestran los resultados obtenidos.

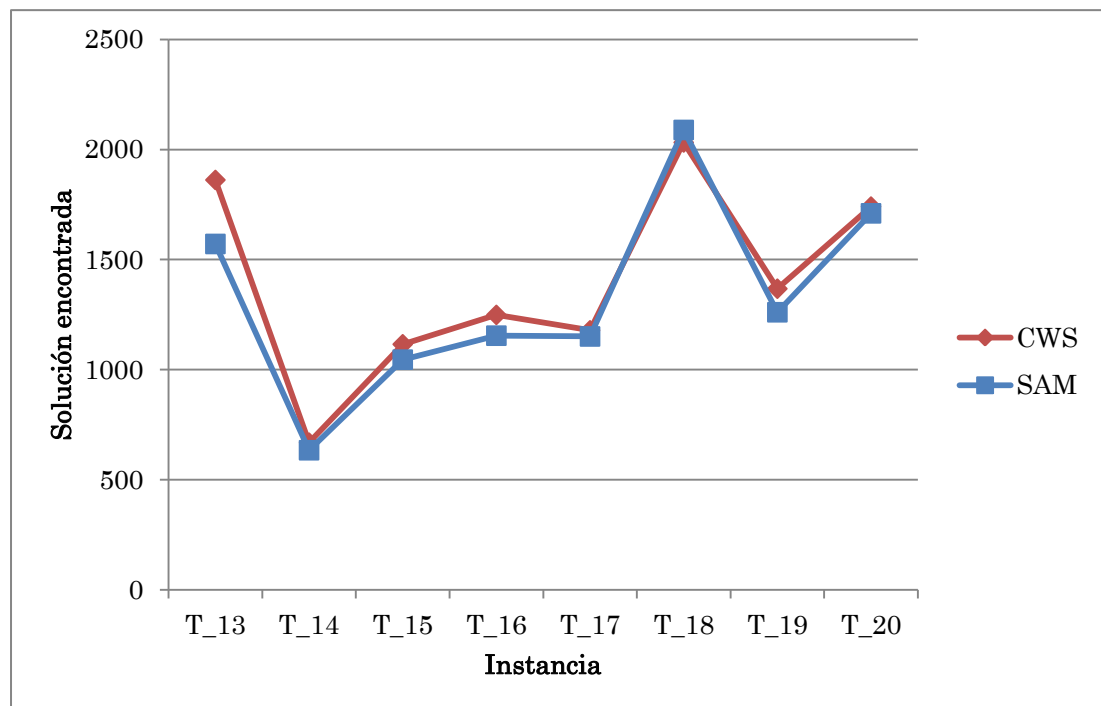


Figura 5.7 Comparación de resultados: SAM vs CWS del conjunto T (VC)

## RESULTADOS SAM VS BS DE LAS INSTANCIAS DEL CONJUNTO T

En la Figura 5.8 se muestran los resultados de método SAM tomando en cuenta los costos fijos así como la mejor solución en literatura. En donde podemos notar, que si bien solo en una instancia se mejoro el resultado encontrado en literatura, encontramos soluciones que consideramos competitivas. Así mismo, como en el caso anterior, se compararon nuestros resultados con los reportados por Laporte et al. (1999) y Tarantilis et al. (2004).

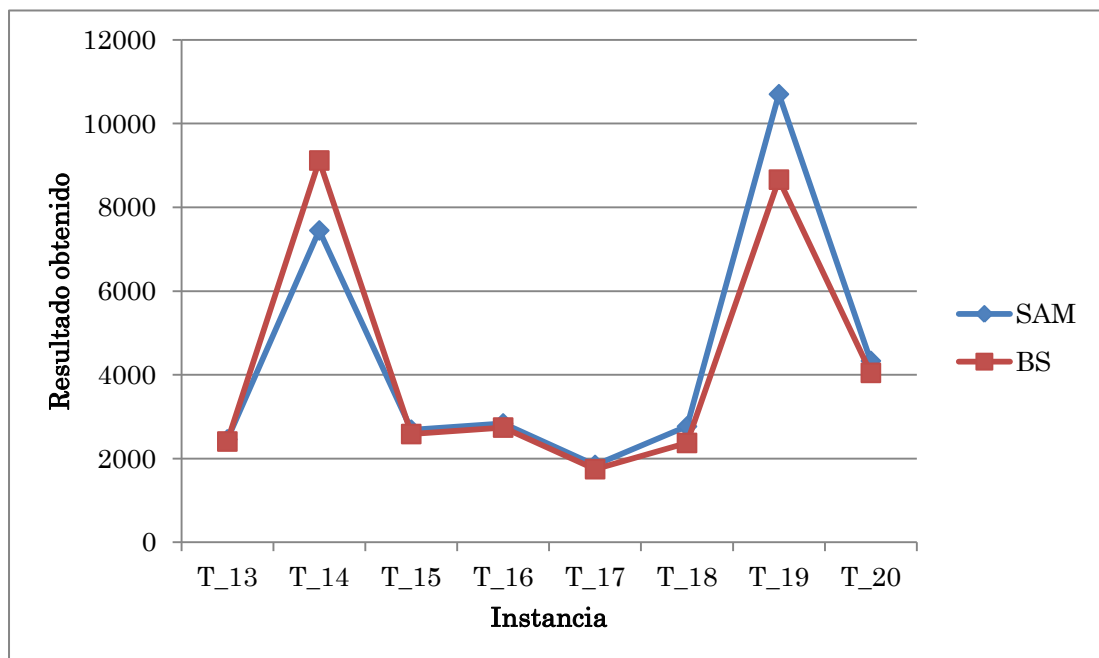


Figura 5.8 Comparación de resultados: SAM vs BS del conjunto T (FC)

Y por último, en la Figura 5.9 se muestra la comparación entre los resultados obtenidos con el método SAM y las soluciones de la literatura tomando en cuenta solo los costos variables, y de la misma forma, no se lograron mejorar los resultados pero se encontraron soluciones competitivas.

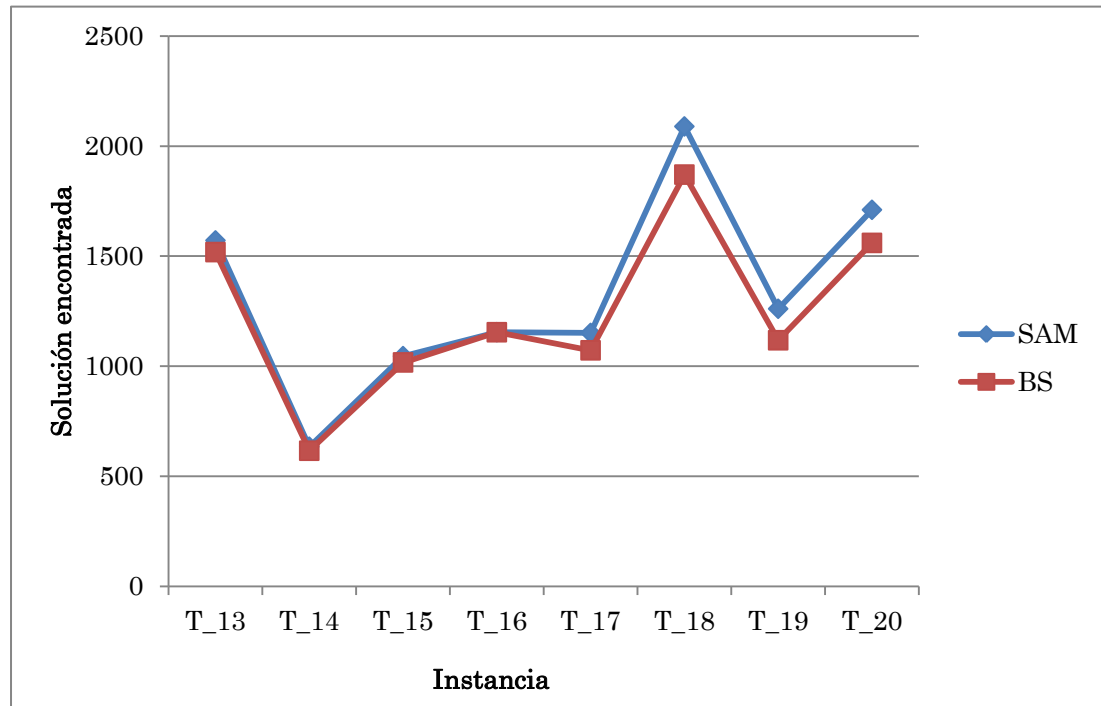


Figura 5.9 Comparación de resultados: SAM vs BS del conjunto T (VC)

## COMPARACIÓN DE TIEMPO DE EJECUCIÓN

Al ser el tiempo de ejecución de SAM un parámetro ajustable, podemos notar que en menos de 5 minutos (300 segundos) obtenemos buenas soluciones

- Tiempos de ejecución del conjunto A

Como podemos ver en la Figura 5.10, hay una diferencia significativa entre los tiempos de ejecución del método SAM y el método de los ahorros, pero como se observó en el apartado anterior, las soluciones encontradas por SAM son mejores que las reportadas por el algoritmo de los ahorros.

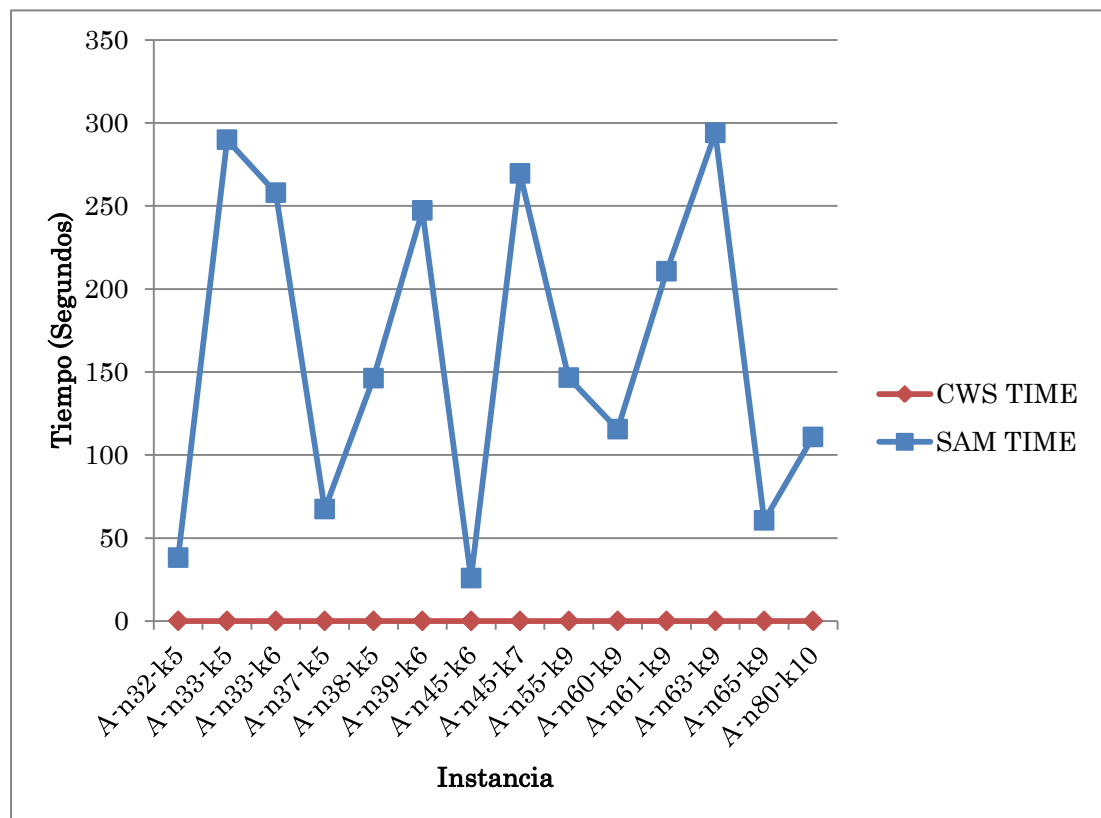


Figura 5.10 Comparación de tiempos de ejecución: SAM vs CWS del conjunto A

- Tiempos de ejecución del conjunto B

El caso de la Figura 5.11 es similar al de la figura 5.10, en donde los tiempos de ejecución del algoritmo de los ahorros es menor al reportado por SAM en la mayoría de las instancias, pero ese tiempo se compensa porque SAM en todas las instancias reporto una mejor solución que la obtenida por el algoritmo de los ahorros.

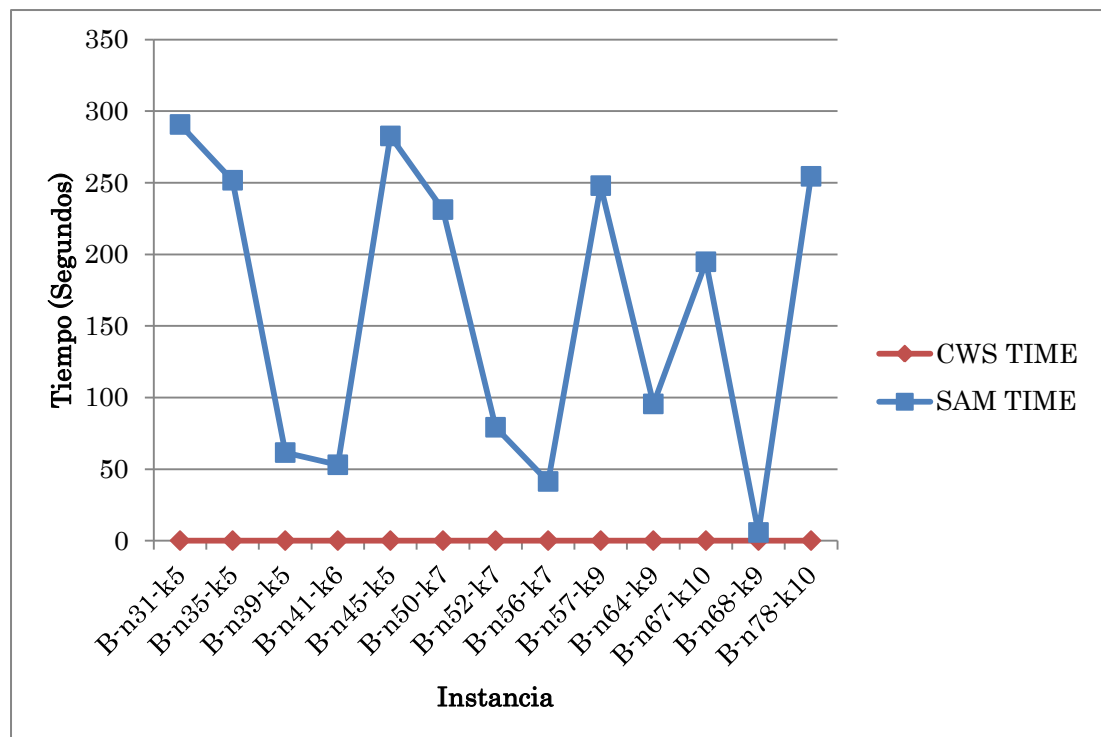


Figura 5.11 Comparación de tiempos de ejecución: SAM vs CWS del conjunto B

- Tiempos de ejecución del conjunto T

La Figura 5.12 muestra el tiempo de ejecución de SAM contra el CWS, en donde tomamos en cuenta solo los costos fijos (FC). Podemos observar que las soluciones encontradas por SAM son de la misma calidad que las encontradas por el CWS.

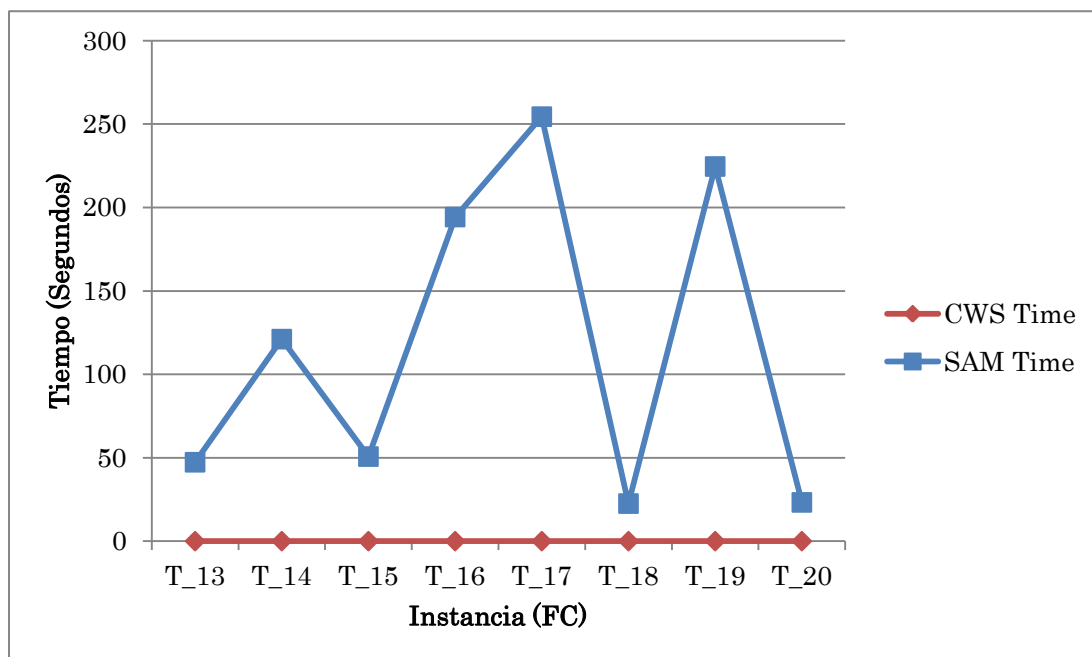


Figura 5.12 Comparación de tiempos de ejecución: SAM vs CWS del conjunto T(FC)

Podemos observar en la Figura 5.13 que SAM encuentra soluciones competitivas en un menor tiempo que las reportadas en literatura, incluso mejoró el resultado en la instancia T\_14.

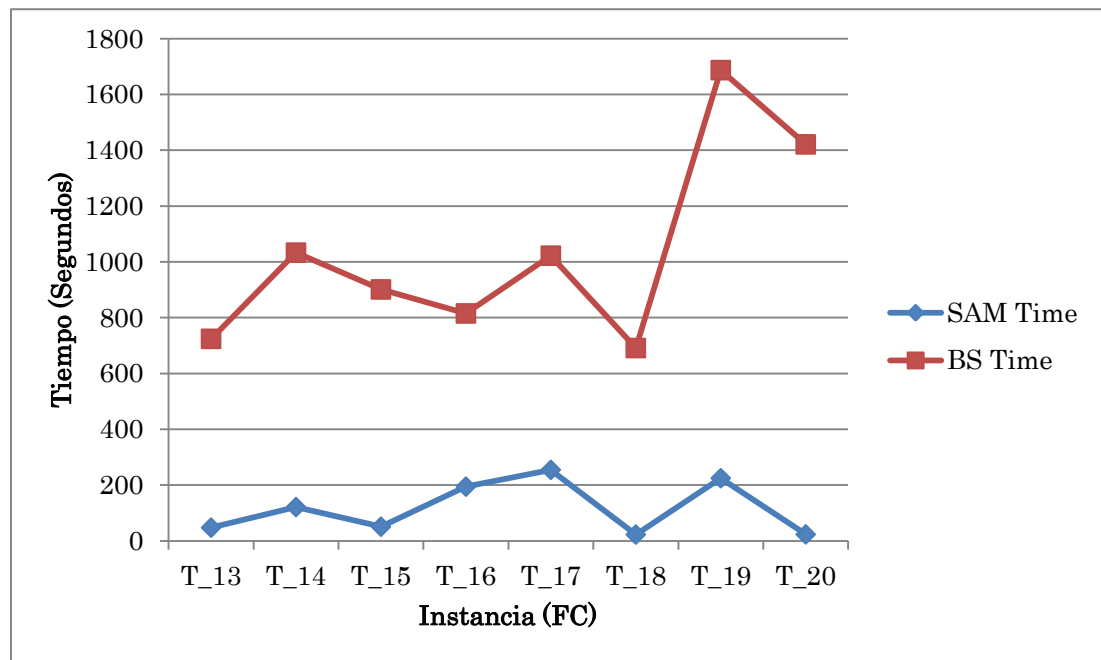


Figura 5.13 Comparación de tiempos de ejecución: SAM vs BS del conjunto T(FC)

Si bien podemos observar en la Figura 5.14 que los tiempos reportados por el CWS es menor que el de SAM, en la Figura 5.10 podemos observar que SAM mejoro la solución encontrada en casi la totalidad de las instancias.



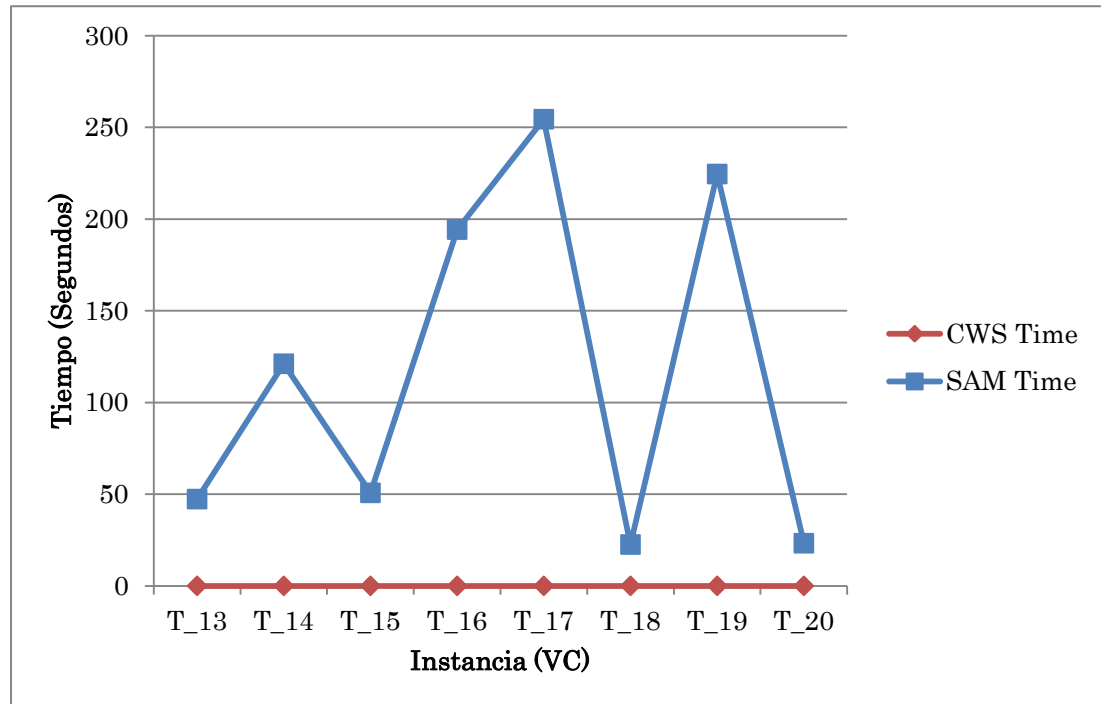


Figura 5.14 Comparación de tiempos de ejecución: SAM vs CWS del conjunto T(VC)

En la Figura 5.15 podemos observar que los tiempos de ejecución de SAM son menores que los reportados en literatura. Así mismo, como se observa en la Figura 5.9 vemos que las soluciones encontradas por SAM son competitivas.

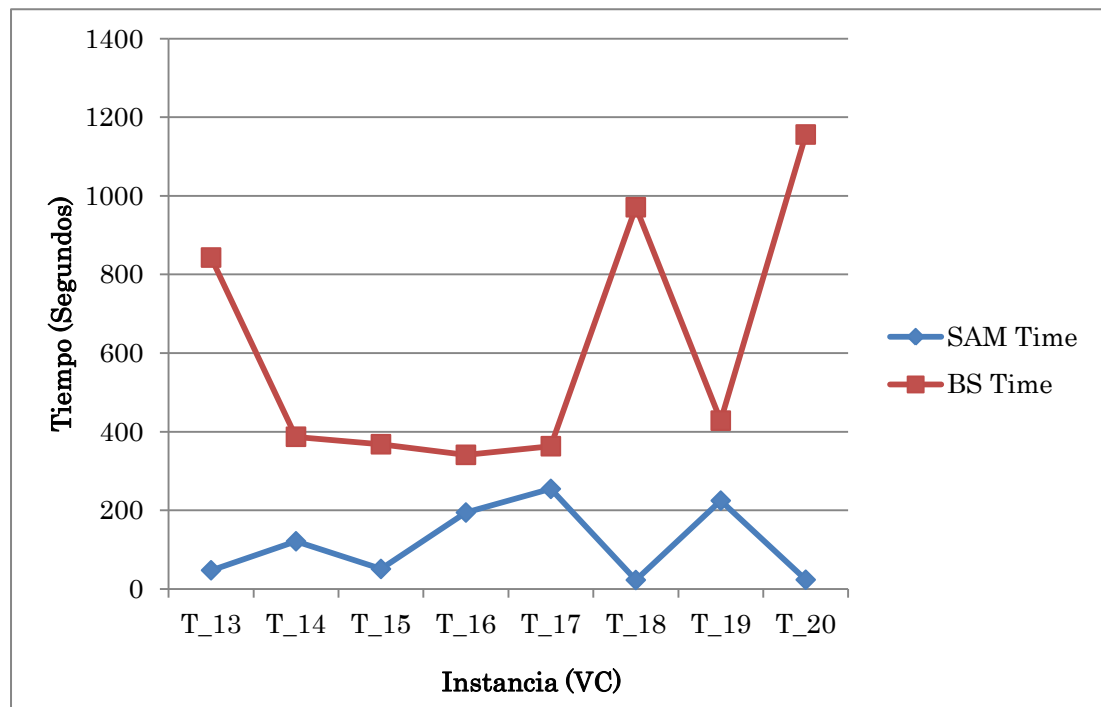


Figura 5.15 Comparación de tiempos de ejecución: SAM vs BS del conjunto T(VC)

Podemos concluir que SAM obtiene buenas soluciones en un tiempo no mayor a los 300 segundos (5 minutos).

## CAPÍTULO 6

# CONCLUSIONES Y TRABAJO FUTURO

### 6.1 CONCLUSIONES

En esta tesis se abordó el problema de ruteo de vehículos con flota heterogénea. Esta variante del problema de ruteo de vehículos original es un tanto más realista, ya que hoy en día es muy común que la flota de vehículos de las empresas sea de diferentes capacidades, esto es, por las especificaciones de empaque del material que están transportando u otros factores.

Para dar solución al problema de ruteo de vehículos heterogéneo se desarrolló una metodología simple y eficiente. Dicho algoritmo es una generalización del algoritmo de los ahorros, en donde se utilizan las técnicas de Monte Carlo como una herramienta para obtener mejores soluciones.

Se realizaron experimentos computacionales con varios tamaños y tipos de instancias tomadas de la literatura para dar validez a nuestro método.

Los resultados se mostraron en el capítulo 5 y pudimos concluir que nuestro método genera buenas soluciones, las cuales compiten, esto es, que el costo incurrido fue menor o muy similar al reportado en la literatura.

## 6.2 TRABAJO FUTURO

Como trabajo a futuro esta integrar al algoritmo las técnicas de *splitting* para optimizar las soluciones generadas. Así probar el algoritmo con un problema real y hacer las adecuaciones correspondientes.

## APÉNDICE I

## DATOS DE INSTANCIAS UTILIZADAS

En este apéndice se muestra el formato de las instancias utilizadas para dar validez a nuestro método.

## CONJUNTO DE INSTANCIAS DEL GRUPO A

NAME : A-n32-k5

COMMENT : (Augerat et al, Min no of trucks: 5, Optimal value: 784)

TYPE : CVRP

DIMENSION : 32

EDGE\_WEIGHT\_TYPE : EUC\_2D

CAPACITY : 100

NODE\_COORD\_SECTION

1 82 76

2 96 44

3 50 5

.

.

.

30 20 70

31 85 60

32 98 5

DEMAND\_SECTION

1 0

2 19

3 21

.

.

.

30 2

31 14

32 9

DEPOT\_SECTION

1

-1

EOF

CONJUNTO DE INSTANCIAS DEL GRUPO B

NAME : B-n31-k5

COMMENT : (Augerat et al, Min no of trucks: 5, Optimal value: 672)

TYPE : CVRP

DIMENSION : 31

EDGE\_WEIGHT\_TYPE : EUC\_2D

CAPACITY : 100

NODE\_COORD\_SECTION

1 17 76

2 24 6

3 96 29

.

.

.

29 26 28

30 17 29

31 20 26

DEMAND\_SECTION

1 0

2 25

3 3

.

.

.

29 10

30 13

31 19

DEPOT\_SECTION

1

-1

EOF

CONJUNTO DE INSTANCIAS DEL GRUPO T

NAME : 13°E076-08s

COMMENT : Gillett and Miller, 1974

TYPE : HVRP

DIMENSION : 51

EDGE\_WEIGHT\_TYPE : EUC\_2D

CAPACITY : 180

VEHICLES : 8

VEHICLE\_KINDS : 6

NODE\_COORD\_SECTION

1 40 40

2 22 22

3 36 26

.

.

.

49 48 21

50 12 38

51 15 56

DEMAND\_SECTION

1 0

2 18

3 26

.

.

.

49 20

50 5

51 22

DEPOT\_SECTION

1

-1

CAPACITIES

20 30 40 70 120 200

FIXED\_COSTS

20 35 50 120 225 400

COST\_COEFFICIENTS

1.0 1.1 1.2 1.7 2.5 3.2

VEHICLES\_AVAILABLE

4 2 4 4 2 1

EOF



## APÉNDICE II

# ALGORITMO

```

public class MultiStartTester
{
    final static String inputFolder = "inputs";// + File.separator + "TSP";
    final static String outputFolder = "outputs";// + File.separator + "TSP";
    final static String fileNameTest = "test2run.txt";//TSP.txt";

    final static String testFolder = "tests";
    final static String sufixFileNodes = "_input_nodes.txt";
    final static String sufixFileVehicules = "_input_vehicules.txt";
    final static String sufixFileOutput = "_outputs.txt";

    public static void main(String[] args)
    {
        System.out.println("**** WELCOME TO THIS PROGRAM ****");
        long programStart = ElapsedTime.systemTime();

        /*****
        * 1. GET THE LIST OF TESTS TO RUN FORM "test2run.txt"
        * aTest = instanceName + testParameters
        *****/

        String testsFilePath = testFolder + File.separator + fileNameTest;
        ArrayList<Test> testsList =
TestsManager.getTestsList(testsFilePath);

        /*****
        * 2. FOR EACH TEST (instanceName + testParameters) IN THE LIST...

```

```

*****/

int nTests = testsList.size();
//System.out.println("nombre de test "+nTests);
for( int k = 0; k < nTests; k++ )
{ Test aTest = testsList.get(k);
  //System.out.println("\n# STARTING TEST " + (k + 1) + " OF " +
nTests);
  long testStart = ElapsedTime.systemTime();

  // 2.1 GET THE INSTANCE INPUTS (DATA ON NODES AND
VEHICLES)
  // "instanceName_input_nodes.txt" contains data on nodes
String inputNodesPath = inputFolder + File.separator +
  aTest.getInstanceName() + suffixFileNodes;
  // "instanceName_input_vehicles.txt" contains data on vehicles
String inputVehPath = inputFolder + File.separator +
  aTest.getInstanceName() + suffixFileVehicules;

  // Read inputs files (nodes + vehicles) and construct the inputs
object
  Inputs inputs = InputsManager.getInputs(inputNodesPath,
inputVehPath);

  // 2.2. USE THE MULTI-START ALGORITHM TO SOLVE THE
INSTANCE
  //MultiStart algorithm = new MultiStart(aTest, inputs);
  Outputs output = MultiStart.solve(aTest, inputs);

  // 2.3. PRINT OUT THE RESULTS TO FILE
"instanceName_seed_outputs.txt"
  String outputsFilePath = outputFolder + File.separator +

```

```

        aTest.getInstanceName() + "_" + aTest.getSeed() +
sufixFileOutput;
        output.sendFile(outputsFilePath);

// 2.4. END OF CURRENT TEST
        long testEnd = ElapsedTime.systemTime();
//      System.out.println("\nElapsed time for this test = "
//      + ElapsedTime.calcElapsedHMS(testStart, testEnd));
    }

/*****
* 3. END OF PROGRAM
*****/

        System.out.println("\n**** END OF PROGRAM, CHECK
OUTPUTS FILES ****");
        long programEnd = ElapsedTime.systemTime();
        System.out.println("Total elapsed time = "
        + ElapsedTime.calcElapsedHMS(programStart, programEnd));
    }
}

```

## BIBLIOGRAFÍA

- ALFA, A. S., HERAGU, S. S., CHEN, M. 1991. A 3-opt Based Simulated Annealing Algorithm for Vehicle Routing Problems. *Computers & Industrial Engineering*, 21, pp. 635 – 639.
- P. AUGERAT, J.M. BELENGUER, E. BENAVENT, A. CORBERÁN, D. NADDEF G. RINALDI. 1995. Computational Results with a Branch and Cut Code for the Capacitated Vehicle Routing Problem. Technical Report RR 949–M, Université Joseph Fourier Grenoble.
- BALDACCI, R., BATTARRA, M., VIGO, D. 2007. Routing a Heterogeneous Fleet of Vehicles. Technical Report DEIS OR.INGCE.
- BALDACCI, R., TOTH, P., VIGO, D. 2007. Recent Advances in Vehicle Routing Exact Algorithms. *4OR*, 5(4), pp. 269 – 298.
- BALDACCI, R., TOTH, P., VIGO, D. 2010. Exact Algorithms for Routing Problems under Vehicle Capacity Constraints. *Annals of Operation Research*, 175(1), pp. 213 – 245.
- BARBAROSOGLU, G., OZGUR, D. 1999. A Tabu Search Algorithm for the Vehicle Routing Problem. *Computer and Operations Research*, 26, pp. 255 – 270.
- BEASLEY, J. E. 1983. Route–First Cluster–Second Methods for Vehicle Routing. *Omega*, 11, pp. 403 – 408.

- BORGULYA, I. 2008. An Algorithm for the Capacitated Vehicle Routing Problem with Route Balancing. *Central European Journal of Operations Research*, 16(4), pp. 331 – 343.
- BUXEY, G. M. 1979. The Vehicle Scheduling Problem and Monte Carlo Simulation. *The Journal of the Operational Research Society*, 30(6), pp. 563 – 573.
- CAMPOS, V., MOTA, E. 2000. Heuristic Procedures for the Capacitated Vehicle Routing Problem. *Computational Optimization and Applications*, 16(3), pp. 265 – 277.
- CHAO, I. M., GOLDEN, B. L., AND WASIL, E. 1999. A computational study of a new heuristic for the site-dependent vehicle routing problem. *INFOR*, 37(3), pp. 319 – 336.
- CHOI, E., TCHA, D. 2007. A Column Generation Approach to the Heterogeneous Fleet Vehicle Routing Problem. *Computers & Operations Research*, 34(7), pp. 2080 – 2095.
- CHRISTOFIDES, N., MINGOZZI, A., TOTH, P. 1979. The Vehicle Routing Problem. In N. Christofides, A. Mingozzi, P. Toth, and C. Sandi, editors, *Combinatorial Optimization*, Wiley, Chichester, UK, pp. 315 – 338.
- CLARKE, G., WRIGHT, J. T. 1964. Scheduling of Vehicles from a Central Depot to a Number of Delivery Points. *Operations Research*, 12(4), pp. 568 – 581.
- COLORNI, A., DORIGO, M., MANIEZZO, V. 1991. Distributed Optimization by Ant Colonies. In F. Varela and P. Bourguine, editors, *Proceedings of the European Conference on Artificial Life*, Elsevier, Amsterdam.

- CORDEAU, J. F., AND LAPORTE, G. 2001. A tabu search algorithm for the site dependent vehicle routing problem with time windows. *INFOR*, 39, pp. 292 – 298.
- DANTZIG, G. B., RAMSER, J. H. 1959. The Truck Dispatching Problem. *Management Science*, 6(1), pp. 80 – 91.
- DELL'AMICO, M., MONACI, M., PAGANI, C., VIGO, D. 2006. Heuristic Approaches for the Fleet Size and Mix Vehicle Routing Problem with Time Windows. Technical Report, DISMI, University of Modena and Reggio Emilia, Italy.
- DESROCHERS, M., VERHOOG, T. W. 1991. A New Heuristic for the Fleet Size and Mix Vehicle Routing Problem. *Computers & Operations Research*, 18(3), pp. 263 – 274.
- DUECK, G. 1993. New Optimization Heuristics: The Great Deluge Algorithm and the Record-to-Record Travel. *Journal of Computational Physics*, 104, pp. 86–92.
- DUECK, G., SCHEURER, T. 1990. Threshold Accepting: A General Purpose Optimization Algorithm. *Journal of Computational Physics*, 90, pp. 161 – 175.
- DULLAERT, W., JANSSENS, G. K., SÖRENSEN, K., VERNIMMEN, B. 2002. New Heuristics for the Fleet Size and Mix Vehicle Routing Problem with Time Windows. *Journal of the Operational Research Society*, 53(11), pp. 1232 – 1238.
- FAULIN, J. AND ANGEL, J. 2008. The ALGACEA-1 Method for the Capacitated Vehicle Routing Problem. *International Transactions in Operational Research*, 15(5), pp. 599 – 621.

- FAULIN, J., GILIBERT, M., JUAN, A., RUIZ, R., VILAJOSANA, X. 2008. SR-1: A Simulation-Based Algorithm for the Capacitated Vehicle Routing Problem. In: Mason JH, Hill R, Moench L, and Rose O (eds). *Proceedings of the 2008 Winter Simulation Conference*, Miami, Florida, USA, December 7 – 10, pp. 2708 – 2716.
- FERLAND, J. A., MICHELON, P. 1988. The Vehicle Scheduling Problem with Multiple Vehicles Types. *Journal of the Operational Research Society*, 39(6), pp. 577 – 583.
- FERNÁNDEZ DE CÓRDOBA, P., GARCÍA-RAFFI, L. M., MAYADO, A., AND SANCHIS, J. M. 2000. A Real Delivery Problem Dealt with Monte Carlo Techniques. *TOP*, 8(1), pp. 57 – 71.
- GENDREAU, M., HERTZ, A., LAPORTE, G. 1992. New Insertion and Postoptimization Procedures for the Traveling Salesman Problem. *Operations Reserarch*, 40, pp. 1086 – 1094.
- GENDREAU, M., LAPORTE, G., MUSARAGANYI, C., TAILLARD, E. D. 1999. A Tabu Search Heuristic for the Heterogeneous Fleet Vehicle Routing Problem. *Computers & Operations Research*, 26(12), pp. 1153 – 1173.
- GHEYSENS, F. G., GOLDEN, B. L., ASSAD, A. A. 1984. A Comparision of Techniques for Solving the Fleet Size and Mix Vehicle Routing Problem. *OR Spectrum*, 6(4), pp. 207 – 216.
- GHEYSENS, F. G., GOLDEN, B. L., ASSAD, A. A. 1986. A New Heuristic for Determining Fleet Size and Composition. *Mathematical Programming Studies*, 26, pp. 233 – 236.

GLOVER, F., LAGUNA, M. 1993. Tabu Search. In C. R. Reeves, editor, Modern Heuristic Techniques for Combinatorial Problems, Blackwell, Oxford, U. K., pp. 70 – 150.

GLOVER, F., LAGUNA, M. 1993. Tabu Search. Kluwer, Boston, MA.

GOLDEN B., ASSAD, A., LEVY, L., GHEYSENS, F. 1984. The Fleet Size and Mix Vehicle Routing Problem. Computers & Operations Research, 11(1), pp. 49 – 66.

GOLDEN, B. L., WASIL, E. A., KELLY, J. P., CHAO, I. M. 1998. Metaheuristics in Vehicle Routing. In T. G. Crainic and G. Laporte, editors, Fleet Management and Logistics, Kluwer, Boston, MA, pp. 33 – 56.

HERTZ, A., TAILLARD, E. D., DE WERRA, D. Tabu Search. In E. H. L. Aarts and J. K. Lenstra, editors, Local Search in Combinatorial Optimization, Wiley, Chichester, UK, pp. 121 – 136.

HILLIER AND LIBERMAN. 2001. Introduction to Operations Research. Seventh Edition. McGraw Hill.

HOLMES, R. A., PARKER, R. G. 1976. A Vehicle Scheduling Procedure Based Upon Savings and a Solution Perturbation Scheme. Operational Research Quarterly, 27(1), pp. 83 – 92.

Instancias para el problema de ruteo de vehículos capacitado:  
<http://osiris.tuwien.ac.at/~wgarn/VehicleRouting/neo/Problem%20Instancias/CVRPinstances.html> (Descarga: 02/09/11)

Instancias para el problema de ruteo de vehículos heterogéneo:  
<http://apice54.ingce.unibo.it/hvrp/> (Descarga: 02/09/11)



- JUAN, A., FAULIN, J., JORBA, J., RIERA, D., MASIP, D. AND BARRIOS, B. 2011. On the use of Monte Carlo Techniques to Improve the Clarke and Wright Savings Heuristics. *Journal of Operational Research Society*, 62, pp. 1085 – 1087.
- JUAN, A., FAULIN, J., RUÍZ, R., BARRIOS, B., CABALLÉ, S. 2010. The SR–GCWS Hybrid Algorithm for Solving the Capacitated Vehicle Routing Problem. *Applied Soft Computing*, 10(1), pp. 215 – 224.
- LI, F., GOLDEN, B. L., WASIL, E. A. 2007. A Record–to–Record Travel Algorithm for Solving the Heterogeneous Fleet Vehicle Routing Problem. *Computers & Operations Research*, 34(9), pp. 2734 – 2742.
- LIN, S. 1965. Computer Solutions of the Traveling Salesman Problem. *Bell System Technical Journal*, 44, pp. 2245 – 2269.
- LIONG CHOONG, Y., WAN ROSMANIRA, I., KHAIRUDDIN, O., MOURAD, Z. 2008. Vehicle Routing Problem: Models and Solutions. *Journal of Quality Measurement and Analysis*, 4(1), pp. 205 – 218.
- LIU, F. H., SHEN, S. Y. 1999. The Fleet Size and Mix Vehicle Routing Problem with Time Windows. *Journal of Operational Research Society*, 50(7), pp. 721 – 732.
- MOLE, R. H., JAMESON, S. R. 1976. A Secuential Route – Buiding Algorithm Employing a Generalised Savings Criterion. *Operation Research Quartely*, 27, pp. 503 – 511.
- NAG, B., GOLDEN, B. L., AND ASSAD, A. 1988. Vehicle routing with side dependencies. In B. L. Golden and A. Assad, editors, *Vehicle Routing: Methods and Studies*, pp. 149 – 159. Elsevier, Amsterdam, Holland.

- OCHI, L. S., VIANNA, D. S., DRUMMOND, M. A., VICTOR, A. O.1998a. An Evolutionary Hybrid Metaheuristic for Solving the Vehicle Routing Problem with Heterogeneous Fleet. *Lecture Notes in Computer Science*, 1391, pp. 187 – 195.
- OCHI, L. S., VIANNA, D. S., DRUMMOND, M. A., VICTOR, A. O.1998b. A Parallel Evolutionary Algorithm for the Vehicle Routing Problem with Heterogeneous Fleet. *Parallel and Distributed Processing*, 1388, pp. 216 – 224.
- OLIVERA, A. 2004. *Heurísticas para Problemas de Ruteo de Vehículos*. Instituto de Computación, Facultad de Ingeniería. Universidad de la República, Montevideo, Uruguay.
- OSMAN, I. H. 1993. Metastrategy Simulated Annealing and Tabu Search Algorithms for the Vehicle Routing Problem. *Annals of Operations Research*, 41, pp. 421 – 451.
- OSMAN, I. H., SALHI, S. 1996. Local Search Strategies for the Vehicle Fleet Mix Problem. In V. J. Rayward-Smith, I. H. Osman, C. R. Reeves, and G. D. Smith, editors, *Modern Heuristic Search Methods*, pp. 131 – 153. Wiley Chichester.
- PISINGER, D., AND ROPKE, S. 2007. A general heuristic for vehicle routing problems. *Comput. Oper. Res.*, 34(8), pp. 2403 – 2435.
- PRINS, C. 2002. Efficient Heuristics for the Heterogeneous Fleet Multitrip VRP with Application to a Large-Scale Real Case. *Journal of Mathematical Modelling and Algorithms*, 1(2), pp. 135 – 150.

- PUREZA V. M., FRANÇA, P. M. Vehicle Routing Problems via Tabu Search Metaheuristic. Technical Report CRT-347, Centre for Research on Transportation, Montreal, Canada.
- RAND, G. K. 2009. The Life and Times of the Savings Method for Vehicle Routing Problems. *ORiON*, 25(2), pp. 125 – 145.
- REGO, C. 1998. A Subpath Ejection Method for the Vehicle Routing Problem. *Management Science*, 44, pp. 1447 – 1459.
- REGO, C., ROUCAIROL, C. 1996. A Parallel Tabu Search Algorithm Using Ejection Chains for the Vehicle Routing Problem. In I. H. Osman and J. P. Kelly, editors, *Meta-Heuristics: Theory and Applications*, Kluwer, Boston, MA, pp. 661 – 675.
- RENAULD, J., BOCTOR, F. F. 2002. A Sweep-Based Algorithm for the Fleet Size and Mix Vehicle Routing Problem. *European Journal of Operational Research*, 140(3), pp. 618 – 628.
- ROBUSTÉ, F., DAGANZO, C. F., SOULEYRETTE, R. 1990. Implementing Vehicle Routing Models. *Transportation Research B*, 24, pp. 263 – 286.
- ROCHAT, Y., TAILLARD, E. D. 1995. Probabilistic Diversification and Intensification in Local Search for Vehicle Routing. *Journal of Heuristics*, 1, pp. 147 – 167.
- SALAZAR, J. J. 2010. Comments on: Routing Problems With Loading Constraints. *TOP*, 18(1), pp. 36 – 38.

- SALHI, S., RAND, G. K. 1993. Incorporating Vehicle Routing into the Vehicle Fleet Composition Problem. *European Journal of Operational Research*, 66(3), pp. 313 – 330.
- TAILLARD, E. D. 1993. Parallel Iterative Search Methods for Vehicle Routing Problems. *Networks*, 23, pp. 661 – 673.
- TAILLARD, E. D. 1999. A Heuristic Column Generation Method for the Heterogeneous Fleet VRP. *RAIRO Recherche Opérationnelle*, 33(1), pp. 1 – 14.
- TAILLARD, E. D., GAMBARDELLA, L. M., GENDREAU, M., POTVIN, J. -Y. 1998. *Adaptative Memory Programming: A Unified View of Metaheuristics*. Research Report IDSIA/19–98, IDSIA, Lugano, Switzerland.
- TAKES, F. W. AND KOSTERS, W. A. 2010. Applying Monte Carlo Techniques to the Capacitated Vehicle Routing Problem. Master Thesis. Leiden Institute of Advanced Computer Science, Leiden University, The Netherlands.
- TARANTILIS, C. D., KIRANOUDIS, C. T., VASSILIADIS, V. S. 2003. A List Based Threshold accepting Metaheuristic for the Heterogeneous Fixed Fleet Vehicle Routing Problem. *Journal of the Operational Research Society*, 54(1), pp. 65 – 71.
- TARANTILIS, C. D., KIRANOUDIS, C. T., VASSILIADIS, V. S. 2004. A Threshold Accepting Metaheuristic for the Heterogeneous Fixed Fleet Vehicle Routing Problem. *European Journal of Operational Research*, 152(1), pp. 148 – 158.
- TOTH, P., VIGO, D. 2002. *The Vehicle Routing Problem*. Editorial Siam, Monographs on Discrete Mathematics and Applications.

- TOTH, P., VIGO, D. 2003. The Granular Tabu Search and its Application to the Vehicle Routing Problem. *INFORMS Journal on Computing*, 15, pp. 333 – 346.
- VAQUERIZO, M. B. 2009. HAIS 2009, LNAI 5572, Springer-Verlag Berlin Heidelberg, pp. 42 – 49.
- WASSAN, N. A., OSMAN, I. H. 2002. Tabu Search Variants for the Mix Fleet Vehicle Routing Problem. *Journal of the Operational Research Society*, 53(7), pp. 768 – 782.
- WILLARD, J. A. G. 1989. Vehicle Routing Using  $r$ -optimal Tabu Search. M.sc. Dissertation, The Management School, Imperial College, London.
- XU, J., KELLY, J. P. 1996. A Network Flow-Based Tabu Search Heuristic for the Vehicle Routing Problem. *Transportation Science*, 30, pp. 379 – 393.
- YAMAN, H. D. 2006. Formulations and Valid Inequalities for the Heterogeneous Vehicle Routing Problem. *Mathematical Programming*, 106(2), pp. 365 – 390.