

División de estudios de posgrado e investigación



“Análisis de factores que afectan el desempeño de  
metaheurísticos aplicados a VRPTW”

Para obtener el grado de:  
Maestro en Ciencias de la Computación

Presenta:  
I.S.C. Juan José Martínez Ponce

Asesor:  
Dra. Claudia G. Gómez Santillán

"2013, Año de la Lealtad Institucional y Centenario del Ejército Mexicano"

Cd. Madero, Tamps; a **25de Octubre de 2013.**

OFICIO No.: U5.269/13  
AREA: DIVISIÓN DE ESTUDIOS  
DE POSGRADO E INVESTIGACIÓN  
ASUNTO: AUTORIZACIÓN DE IMPRESIÓN DE TESIS

**ING. JUAN JOSÉ MARTÍNEZ PONCE**  
**NO. DE CONTROL G06070450**  
**PRESENTE**

Me es grato comunicarle que después de la revisión realizada por el Jurado designado para su examen de grado de Maestría en Ciencias de la Computación, el cual está integrado por los siguientes catedráticos:

PRESIDENTE :	DRA. LAURA CRUZ REYES
SECRETARIO :	DR. JOSÉ ANTONIO MARTÍNEZ FLORES
VOCAL :	DRA. CLAUDIA GUADALUPE GÓMEZ SANTILLÁN
SUPLENTE	DR. JUAN JAVIER GONZÁLEZ BARBOSA
DIRECTOR DE TESIS :	DRA. CLAUDIA GUADALUPE GÓMEZ SANTILLÁN
CO-DIRECTOR:	DR. JUAN JAVIER GONZÁLEZ BARBOSA

Se acordó autorizar la impresión de su tesis titulada:

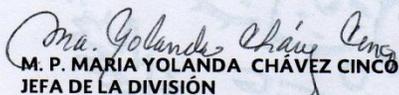
**"ANÁLISIS DE FACTORES QUE AFECTAN EL DESEMPEÑO DE METAHEURÍSTICOS APLICADOS A VRPTW"**

Es muy satisfactorio para la División de Estudios de Posgrado e Investigación compartir con Usted el logro de esta meta.

Espero que continúe con éxito su desarrollo profesional y dedique su experiencia e inteligencia en beneficio de México.

**ATENTAMENTE**

"Por mi patria y por mi bien"®

  
**M. P. MARIA YOLANDA CHÁVEZ CINCO**  
**JEFA DE LA DIVISIÓN**

c.c.p.- Minuta  
Archivo  
MYCHC 'NICO'.jar

  
S.E.P.  
DIVISION DE ESTUDIOS  
DE POSGRADO E  
INVESTIGACION  
ITCM



Ave. 1° de Mayo y Sor Juana I. de la Cruz, Col. Los Mangos, CP. 89440 Cd. Madero, Tam.  
Tel. (833) 357 48 20, Fax, Ext. 1002, e-mail: itcm@itcm.edu.mx

[www.itcm.edu.mx](http://www.itcm.edu.mx)



## **DEDICATORIA**

Entiendo que por todo lo que me ama mi familia es fácil mantenerlos orgullosos de mí, por lo que el esfuerzo que apliqué a este trabajo y que aplico en otros aspectos de mi vida, es siempre pensando en lo gratificante que resulta para mí el superar sus expectativas. Tengan por seguro que seguiré actuando lo más responsablemente posible, para que pensar en mí les traiga la satisfacción tanto de haber hecho buen trabajo criándome, como de saber que estaré bien por mis propios medios.

## **AGRADECIMIENTOS**

Aprecio profundamente la guía que Dra. Claudia y mis asesores me brindaron, cada uno con su estilo propio supo aconsejarme y al mismo tiempo confiar en mi capacidad académica.

Siempre estaré orgulloso de haber pertenecido al Instituto Tecnológico de Ciudad Madero, porque creo fielmente en su prestigio, no sólo por comentarios externos, sino por haber tenido la posibilidad de constatar de primera mano el compromiso que este instituto tiene con el futuro de nuestro país.

De igual manera considero necesario mencionar, el gran aporte que el Consejo Nacional de Ciencias y Tecnología hizo en mí, y en otros que como yo desean engrandecer a México por medio de la ciencia.

## **Declaración de Originalidad**

Declaro y prometo que este documento de tesis es producto de mi trabajo original y que no infringe los derechos de terceros, tales como derecho de publicación, derechos de autor, patentes y similares.

Además, declaro que en las citas textuales que he incluido y en los resúmenes que he realizado de publicaciones ajenas, indico explícitamente los datos de los autores y las publicaciones.

Además, en caso de infraccionar los derechos de terceros derivados de este documento de tesis, acepto la responsabilidad de la infracción y relevo de ésta a mi director y codirector de tesis, así como al Instituto Tecnológico de Cd. Madero y sus autoridades.

Noviembre de 2013, Cd. Madero, Tamaulipas

---

ISC. Juan José Martínez Ponce

## RESUMEN

En el área de las Ciencias Computacionales, el *Problema de Rutas de Vehículos con Ventanas de Tiempo* (VRPTW, por sus siglas en inglés) se define como el problema de diseñar rutas con el menor costo de viaje desde un almacén central, a un conjunto de clientes geográficamente dispersos, con demandas conocidas y con horario de atención establecido (ventanas de tiempo). Debido a su relevancia en los ámbitos científicos y sectores productivos y de servicio, el VRPTW ha recibido atención en las últimas décadas.

En este trabajo se aborda el VRPTW, para su solución se empleó el algoritmo de múltiples sistemas de colonia de hormigas (MACS-VRPTW, por sus siglas en inglés). Este emula el verdadero comportamiento de las hormigas, donde depositan en el suelo una sustancia química llamada feromona cada vez que recorren el camino desde el hormiguero hasta la fuente de comida. Lo anterior implica que aquellos tramos donde existe mayor cantidad de dicha feromona, son los predilectos para ser recorridos por las hormigas.

Para mejorar el desempeño de MACS-VRPTW fueron analizadas las heurísticas que proporcionaban información al sistema de colonia de hormigas, tal como las listas restringidas diseñadas para extraer datos relevantes de aspecto geográfico y de tiempo de disponibilidad de los clientes. Se diseñaron e implementaron nuevos algoritmos para incrementar el beneficio de la información suministrada al sistema de colonia de hormigas.

## SUMMARY

In the field of Computer Science, the *Vehicle Routing Problem with Time Windows* (VRPTW) is defined as a problem of designing least cost routes from a depot to a set of geographically scattered customers with known demand and service times (time windows). Its importance in computer science, operations research and transportation logistics has lead VRPTW to receive plenty of attention in recent decades.

This work approaches VRPTW; a multiple ant colony system (MACS-VRPTW) was applied to solve it. This reproduces actual ants' behavior; they release and leave on the ground a chemical substance called "*pheromone*" every time they walk through a part of the road, starting from their anthill to the food source. Information above implies those tracts with a higher amount of the "pheromone" will be preferred by the ants to be walked through.

In order to improve MACS-VRPTW performance, heuristics responsible for supplying information to multiple ant colony system were analyzed, such as restricted lists designed to extract relevant data related to location and availability of clients. New algorithms were designed and implemented intending to increase the benefit from information provided to the multiple ant colony system.

## TABLA DE CONTENIDO

<b>1</b>	<b>Introducción</b>	1
1.1	Antecedentes	2
1.2	Descripción del problema de investigación	3
1.3	Objetivos	4
1.3.1	Objetivo general	4
1.3.2	Objetivos específicos	4
1.4	Alcances y limitaciones	5
1.5	Justificación	5
1.6	Organización del documento	6
<b>2</b>	<b>Marco teórico</b>	7
2.1	Complejidad de los problemas	7
2.1.1	Problemas de decisión y análisis de algoritmos	7
2.1.2	Clase P	8
2.1.3	Problemas intratables	8
2.1.4	Clase NP	8
2.1.5	Relación entre P y NP	8
2.1.6	Transformación polinomial	8
2.1.7	Problemas NP-completos y NP-duros	9
2.2	Problema de rutas de vehículos y su variante con ventanas de tiempo VRPTW	9
2.3	Complejidad del VRPTW	11
2.4	Enfoques de solución	12
2.4.1	Métodos exactos	12
2.4.2	Métodos heurísticos	13
2.5	Estrategias de solución	14
2.5.1	Algoritmos metaheurísticos	14
2.5.2	Búsqueda local	17
2.6	Análisis de datos	18
2.6.1	Modelos estadísticos	19
2.6.2	Correlación	21
2.6.3	Análisis de varianza (ANOVA) y factorial	22

<b>3 Estado del arte.....</b>	<b>23</b>
3.1 Trabajos que experimentan con instancias de Solomon.....	23
3.2 Trabajos relacionados con VRPTW en el ITCM.....	24
3.3 Análisis del estado del arte.....	25
<b>4 Metodología.....</b>	<b>27</b>
4.1 Análisis experimental.....	27
4.2 Caracterización del comportamiento del algoritmo.....	28
4.2.1 Búsqueda inicial.....	31
4.2.2 Heurísticas de caracterización.....	31
4.2.3 Búsqueda local.....	35
4.2.4 Aprendizaje por niveles.....	36
4.2.5 Sistemas de colonia de hormigas.....	37
<b>5 Propuesta de solución.....</b>	<b>39</b>
5.1 Lista restringida extendida LRE.....	39
5.1.1 Caracterización por intervalos.....	40
5.1.2 Activación por intervalos.....	41
5.1.3 Inicio de ventana.....	42
5.1.4 LRE Adaptativo.....	43
5.2 Lista restringida adaptativa.....	44
5.2.1 Agrupación semi-jerárquica.....	44
5.2.2 Agrupación alterna.....	45
5.3 Búsqueda inicial y local.....	48
5.4 Aprendizaje por niveles.....	49
<b>6 Experimentación y resultados.....</b>	<b>50</b>
6.1 Ranking de estrategias.....	50
6.2 Revisión de condición de activación de LRE por valores estadísticos.....	52
6.3 Propuesta de condición de activación de LRE por intervalos no vacíos.....	53
6.4 Revisión de LRA y propuesta de algoritmos de agrupación.....	59
6.5 Algoritmo genético para ajuste de estrategias.....	65
6.6 Resultados.....	69
<b>7 Conclusiones y trabajos futuros.....</b>	<b>70</b>
7.1 Conclusiones.....	70
7.2 Aportaciones de la investigación.....	71
7.3 Publicaciones.....	71
7.4 Trabajos futuros.....	72
7.5 Productos científicos.....	72
<b>8 Referencias bibliográficas.....</b>	<b>73</b>

## ÍNDICE DE TABLAS

2.1 Caracterización de las instancias de Solomon por su distribución geográfica.....	11
2.2 Clasificación de los modelos estadísticos.....	21
3.1 Abreviatura de la tabla 3.2 y 3.3 mostrando el nombre del autor, metodología y año de publicación de la investigación correspondiente.....	25
3.2 Mejores resultados logrados en el estado del arte que experimentan con Solomon por cada tipo de instancia.....	26
3.3 Mejores resultados logrados en el ITCM por cada tipo de las instancias de Solomon.....	26
3.4 Análisis de las estrategias utilizadas en trabajos del estado del arte que experimentan con Solomon .....	26
3.5 Análisis de las estrategias utilizadas en trabajos del ITCM que experimentan con Solomon .....	26
5.1 Modificaciones realizadas al algoritmo MACS-VRPTW.....	39
5.2 Caracterización por intervalos.....	40
6.1 Ranking de estrategias.....	51
6.2 Cantidad de grupos generados por agrupación jerárquica.....	61
6.3 Tabla de coeficientes de los algoritmos de agrupación.....	64
6.4 Relación de valores que pueden adquirir del octavo al décimo segundo gen y el algoritmo que se ejecuta de acuerdo a tal valor.....	65
6.5 Parámetros para un algoritmo genético que selecciona estrategias.....	67
6.6 Parámetros utilizados en algoritmo MACS-VRPTW por Cámara.....	69
6.7 Promedios de vehículos y distancia recorrida por tipo de instancia.....	69

## ÍNDICE DE FIGURAS

1.1	Algoritmo general del problema de investigación.....	3
2.1	Pseudocódigo de algoritmo metaheurístico de ACS general.....	15
2.2	Procedimiento de búsqueda local.....	17
4.1	Esquema de la metodología propuesta.....	27
4.2	Esquema de solución de Cámara.....	28
4.3	Esquema de Módulo APN.....	29
4.4	Esquema de Módulo de construcción de solución.....	29
4.5	Caracterización de las instancias de Solomon en base a su ventana de tiempo.....	29 30
4.6	Caracterización de las instancias escalonadas.....	30
4.7	Esquema 3-opt.....	35
4.8	Esquema cross-exchange.....	36
5.1	Gráfica de ventanas de tiempo de instancia ejemplo dividida en intervalos.....	40
5.2	Módulo de construcción de solución parcial de esquema adaptativo.....	43
5.3	Algoritmo de agrupación semi-jerárquica.....	44
5.4	Árbol resultante de agrupación semi-jerárquica.....	45
5.5	Algoritmo de agrupación alterna.....	45
5.6	Proceso de agrupación alterna.....	48
5.7	Ejemplo de permutaciones en movimiento para un entero k de valor 2.....	49
6.1	Efectividad de LRE con valores estadísticos como condición de activación.....	52
6.2	Instancias identificadas como lineales bajo la caracterización por intervalos (parte 1).....	53
6.3	Instancias identificadas como lineales bajo la caracterización por intervalos (parte 2).....	54
6.4	Instancias identificadas como curvas bajo la caracterización por intervalos (parte 1).....	54
6.5	Instancias identificadas como curvas bajo la caracterización por intervalos (parte 2).....	55
6.6	Instancias identificadas como curvas bajo la caracterización por intervalos (parte 3).....	56
6.7	Instancias identificadas como escalonadas bajo la caracterización por intervalos (parte 1).....	57
6.8	Instancias identificadas como escalonadas bajo la caracterización por intervalos (parte 2).....	58
6.9	Árbol de expansión mínimo para instancias C1 y C2.....	59
6.10	Árbol de expansión mínimo para instancias R1 y R2.....	59
6.11	Árbol de expansión mínimo para instancias RC1 y RC2.....	60

6.12 Agrupación jerárquica para instancias C1 y C2.....	60
6.13 Agrupación jerárquica para instancias R1 y R2.....	61
6.14 Agrupación jerárquica para instancias RC1 y RC2.....	61
6.15 Agrupación semi-jerárquica para instancias C1 y C2.....	62
6.16 Agrupación semi-jerárquica para instancias R1 y R2.....	62
6.17 Agrupación semi-jerárquica para instancias RC1 y RC2.....	62
6.18 Agrupación alterna para instancias C1 y C2.....	63
6.19 Agrupación alterna para instancias R1 y R2.....	63
6.20 Agrupación alterna para instancias RC1 y RC2.....	63
6.21 Individuo en población de genético de estrategias.....	65
6.22 Elementos de función objetivo para un individuo en población de genético de estrategias.....	66
6.23 Esquema de solución de algoritmo genético.....	68

## ÍNDICE DE FÓRMULAS

2.1	Expresión de VRPTW.....	10
2.2	Fórmula de correlación.....	22
4.1	Porcentaje de variabilidad de costos asociados a los arcos de MST.....	32
4.2	Valor inicial de umbral de agrupación.....	32
4.3	Cálculo de ventana de tiempo.....	33
4.4	Cálculo de media de ventanas de tiempo de instancia.....	33
4.5	Cálculo de desviación estándar de ventanas de tiempo de instancia.....	33
4.6	Condición de activación de heurística LRE utilizando media.....	34
4.7	Disyunción de importancia de clientes según su ventana de tiempo.....	34
4.8	Disyunción de valor para tabla de prioridad utilizando importancia de clientes.....	34
4.9	Primer nivel de actualización de la tabla de feromonas.....	36
4.10	Segundo nivel de actualización de la tabla de feromonas.....	36
4.11	Actualización de copia de tabla de feromonas.....	37
4.12	Regla de selección de candidatos.....	37
4.13	Probabilidad de hormiga de ir de cliente $i$ a cliente $j$ .....	37
4.14	Cálculo de información heurística.....	38
5.1	Cálculo de intervalo de ventanas.....	41
5.2	Cálculo de número de clientes en intervalo $k$ .....	41
5.3	Disyunción de pertenencia a intervalo $k$ .....	41
5.4	Cálculo de intervalos no vacíos.....	41
5.5	Disyunción de intervalo vacío.....	41
5.6	Disyunción de valor para tabla de prioridad.....	42
5.7	Cálculo de intervalo de inicio de ventanas.....	43
5.8	Disyunción de valor para tabla de prioridad utilizando inicio de ventana de tiempo.....	43

# Capítulo 1

## INTRODUCCIÓN

La mayoría de los trabajos relacionados con el análisis del desempeño de algoritmos metaheurísticos están enfocados en validar los resultados experimentales a partir de comparaciones. Desde hace algunos años ha venido tomando fuerza la necesidad e importancia de aprender y explicar el desempeño observado, esta tarea consiste en analizar, identificar y comprender los factores que afectan directamente o indirectamente el desempeño obtenido por un algoritmo [Pérez, 2007].

El análisis individual de los factores significantes revela interacciones entre ellos e indicadores de buen o mal desempeño. Las interacciones determinan si la unión de dos o más factores repercute en el rendimiento del algoritmo.

Existen métodos para la caracterización de algoritmos metaheurísticos cuya finalidad es entender como su desempeño depende de una serie de factores que lo influyen y así conducir a mejores predicciones del desempeño en nuevas situaciones y al descubrimiento de algoritmos mejorados. Lo anterior pretende encausar el desarrollo de heurísticas que mejoren el rendimiento de los algoritmos a través de estrategias tales como: búsqueda local, aprendizaje por niveles, búsqueda inicial, y lista restringida de candidatos.

## 1.1 ANTECEDENTES

Para explicar y generalizar el comportamiento presentado por los algoritmos; desde hace algunos años en el Instituto Tecnológico de Ciudad Madero se ha incursionado en el desarrollo y aplicación de estrategias que proporcionen explicaciones confiables del comportamiento observado, con ellas es posible obtener mayor conocimiento de los sistemas y por lo tanto brindar retroalimentación acerca de cómo mejorar su funcionamiento. Otro aspecto a considerar es tener un alto conocimiento del problema a enfrentar, lo cual lleva a explicaciones mucho más precisas.

El problema abordado en este trabajo en su forma más general se le conoce como Problema de Rutas de Vehículos (Vehicle Routing Problem, VRP por sus siglas en inglés). Éste se relaciona con el problema de gestión de transporte y ha sido muy trabajado por la comunidad científica. VRP tiene sus orígenes en el problema de envío de camiones estudiado por [Dantzing, 1959] en el cual se determinan las rutas óptimas para satisfacer las demandas de un número determinado de clientes utilizando una flotilla de vehículos, respetando restricciones de vehículos, clientes, etc. Con la particularidad de que las rutas inician y terminan en el almacén central y solamente se asocia un solo vehículo por ruta.

La variante de VRP con la que se ha trabajado en el Instituto Tecnológico de Ciudad Madero es el Problema de Rutas de Vehículos con Ventanas de Tiempo (Vehicle Routing Problem with Time Windows, VRPTW por sus siglas en inglés). El algoritmo utilizado hasta el momento es el conocido como MACS-VRPTW (Multiple Ant Colony System for Vehicle Routing Problem with Time Windows). Herrera en [Herrera, 2006] propuso una heurística integral para solucionar el problema de asignación de rutas y horarios, agregó caracterización topológica de clientes realizando agrupación en base a ella, añadió de igual forma la búsqueda inicial de vecinos cercanos y una búsqueda local de Cross-Exchange. Enseguida Arrañaga en [Arrañaga, 2007] usó una técnica llamada Aprendizaje por Niveles la cual colaboraba con el aprendizaje de las hormigas. Finalmente Cámara en [Cámara, 2011] incorporó la caracterización de ventanas de tiempo.

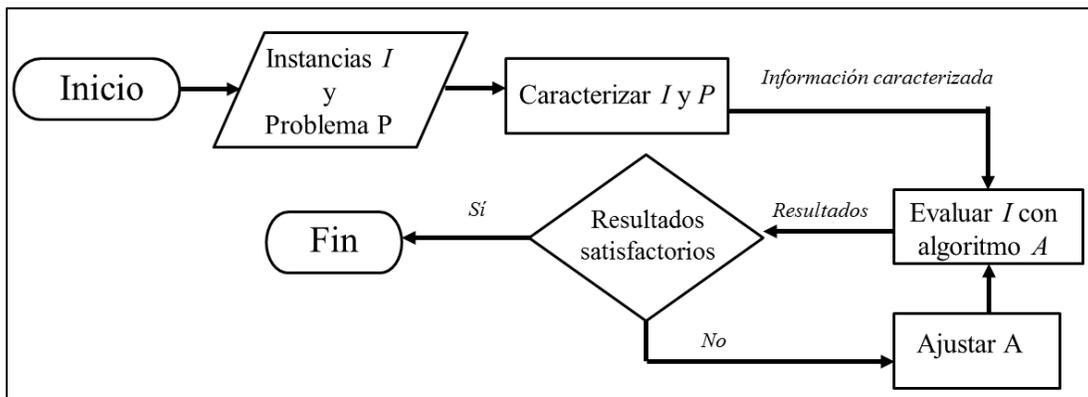
## 1.2 DESCRIPCIÓN DEL PROBLEMA DE INVESTIGACIÓN

Los parámetros de un algoritmo necesitan ajustarse para obtener resultados de buena calidad. Dichos parámetros pueden ser incluidos como variables en funciones matemáticas, siendo esto una práctica de gran utilidad en muchas disciplinas del conocimiento, por ejemplo: las industriales, científicas y tecnológicas.

Los investigadores frecuentemente invierten mucho tiempo en tratar de ajustar los parámetros del algoritmo. En ocasiones los investigadores consideran adecuado el configurar la activación de las heurísticas del algoritmo para una instancia específica invocando a otro metaheurístico, a esto se le llama meta-evolución [Gómez, 2009].

Dado un conjunto de instancias  $I$  de la literatura, y un problema  $P$  de optimización como el VRPTW, mediante el algoritmo  $A$  de múltiples sistemas de colonia de hormigas MACS-VRPTW y un conjunto de heurísticas, se debe encontrar una  $\langle I_i, P_i, A_i \rangle$  para el cual  $A$  tiene mejor rendimiento.

Figura 1.1 Algoritmo general del problema de investigación.



## **1.3 OBJETIVOS**

Aquí se describen los objetivos generales y específicos planteados para esta investigación.

### *1.3.1 OBJETIVO GENERAL*

Proponer estrategias de solución para el problema VRPTW a través de una metodología basada en análisis estadístico sobre las instancias seleccionadas, y así mejorar el desempeño del algoritmo MACS-VRPTW.

### *1.3.2 OBJETIVOS ESPECÍFICOS*

Para alcanzar el objetivo principal de este trabajo es necesario llevar a cabo una serie de etapas que, en conjunto, contribuirán en la obtención de los resultados requeridos. De esta forma se han definido los siguientes objetivos específicos:

- Seleccionar un algoritmo metaheurísticos del estado del arte, que muestre un nivel de competencia internacional y pueda ser evaluado en condiciones similares.
- Preparar conjuntos de instancias de prueba, relevantes y representativas de problemas reconocidos por la comunidad científica.
- Identificar los factores que afectan el comportamiento del algoritmo seleccionado para mejorarlo.
- Aplicar análisis estadístico, para proponer estrategias heurísticas innovadoras que pudieran ser incorporadas en el algoritmo metaheurístico MACS-VRPTW para incrementar su desempeño.

## **1.4 ALCANCES Y LIMITACIONES**

- La caracterización del proceso de optimización será aplicado únicamente al problema de ruteo de vehículos VRPTW.
- En este trabajo se emplearon únicamente las instancias de [Solomon, 1987].

## **1.5 JUSTIFICACIÓN**

La complejidad de los problemas reales ha hecho necesario el uso de estrategias heurísticas inteligentes. El VRPTW ha sido clasificado como un problema del tipo NP-Duro siendo estudiado por la comunidad científica por lo menos los últimos veinte años [Kallehauge, 2006]. Con el propósito de desarrollar mejores estrategias, los investigadores han intentado explicar el comportamiento de este tipo de algoritmos ante diferentes instancias de un problema, partiendo de análisis experimentales y analizando estadísticamente los datos obtenidos [Quiroz ,2009].

Los algoritmos de aprendizaje automático para clasificación y predicción son muy utilizados en la minería de datos para extracción de conclusiones sobre diferentes casos de estudio, sin embargo parecen ser dependientes de los datos a partir de los cuales las conclusiones son generadas, esto disminuye la aplicabilidad de dichas estrategias al análisis de algoritmos ya que por sí solas no permiten generalizar las conclusiones obtenidas [Pérez, 2007] [Menzies, 2009].

## **1.6 ORGANIZACIÓN DEL DOCUMENTO**

Este documento se estructuró de la siguiente forma, en el capítulo dos se presenta la teoría necesaria para la llevar a cabo este trabajo, en ella se incluye lo relacionado con el problema a resolver y su complejidad, la estructura básica del algoritmo y heurísticas empleadas. En el capítulo tres se exhibe tanto el estado del arte como los trabajos realizados dentro del Instituto Tecnológico de Ciudad Madero (ITCM). Para ambos se presenta un listado por año de publicación, autores y estrategias empleadas por estos. El capítulo cuatro muestra la metodología utilizada, tanto las caracterizaciones como la descripción de las estrategias utilizadas en el ITCM que fueron incorporadas en este trabajo. En el capítulo cinco se incluyen las descripciones de las estrategias que fueron modificadas y de aquellas estrategias que fueron propuestas, en el capítulo seis se muestran graficas de comportamiento de las estrategias propuestas sobre las instancias de [Solomon, 1987] y resultados del algoritmo, mientras que en el capítulo siete son mostradas las conclusiones generadas, trabajos futuros, aportaciones y publicaciones realizadas.

# Capítulo 2

## MARCO TEÓRICO

En este capítulo se describe la terminología necesaria para el desarrollo de este trabajo de tesis.

### 2.1 COMPLEJIDAD DE LOS PROBLEMAS

En los problemas de tipo combinatorio el método tradicional para encontrar una solución óptima es realizar una búsqueda exhaustiva en el conjunto de soluciones posibles, es decir generar todas las configuraciones factibles, calcular su costo y elegir aquella que ofrezca mejores resultados [Herrera, 2006]. Esta metodología no es eficiente debido al crecimiento exponencial del tiempo de cálculo en función de varios factores como la complejidad de la instancia del problema. Stephen Cook en 1971 [Garey, 1979] propuso los fundamentos para lo que hoy se conoce como la teoría NP-Complejidad.

#### 2.1.1 PROBLEMAS DE DECISIÓN Y ANÁLISIS DE ALGORITMOS

Un problema de decisión  $\pi = (D, Y)$  es una pareja formada por un conjunto de instancias  $D$ , las cuales se obtienen a partir de una instancia genérica que se especifica en términos de varios componentes: conjuntos, funciones, números, etc.; y un subconjunto de instancias-sí  $Y \subseteq D$ . Una instancia  $i \in Y$ , si y sólo si la respuesta a la cuestión del problema es sí para esa instancia.

Un problema de decisión se asocia con un lenguaje formal usando algún mecanismo de codificación y un algoritmo con una máquina de Turing. Un algoritmo se dice que resuelve un problema de decisión si y sólo si el lenguaje aceptado por la máquina de Turing es el subconjunto de todas las cadenas asociadas con las instancias-sí del problema. En las siguientes definiciones, si no se señala otra cosa, la palabra problema se usa como sinónimo de problema de decisión.

### 2.1.2 CLASE $P$

Es el conjunto de todos los problemas de decisión que pueden ser resueltos en tiempo polinomial por un algoritmo determinista. A los problemas que pertenecen a esta clase se les denomina tratables.

### 2.1.3 PROBLEMAS INTRATABLES

Son todos los problemas de decisión para los que no existe un algoritmo determinista de tiempo polinomial que los resuelva, es decir, son todos los problemas comprendidos en  $P^c$ .

### 2.1.4 CLASE $NP$

La clase  $NP$  es el conjunto de todos los problemas de decisión que se pueden verificar en tiempo polinomial con un algoritmo no determinista.

### 2.1.5 RELACIÓN ENTRE $P$ Y $NP$

Como toda máquina determinista es un caso particular de una máquina no determinista, se tiene entonces que  $P \subseteq NP$ .

### 2.1.6 TRANSFORMACIÓN POLINOMIAL

Se dice que un problema de decisión  $\pi_1 = (D_1, Y_1)$  se puede transformar polinomialmente en el problema de decisión  $\pi_2 = (D_2, Y_2)$ , si y sólo si existe una función  $f: D_1 \rightarrow D_2$  que satisface las siguientes dos condiciones:

1.  $f$  es computable con un algoritmo determinista de tiempo polinomial.
2. Para toda instancia  $i \in D_1$ ,  $i \in Y_1$ , si y sólo si  $f(i) \in Y_2$ .

En tal caso se dice que  $\pi_1 \alpha_p \pi_2$ .

### 2.1.7 PROBLEMAS NP-COMPLETOS Y NP-DUROS

Un problema  $\pi$  es *NP-Completo* si y sólo si,  $\pi \in NP$  y  $\forall \phi \in NP, \phi \leq_p \pi$ . Al conjunto de todos los problemas *NP-Completo* se les denota por *NPC*. Un problema de optimización se dice que es *NP-Duro*, si y sólo si, su versión de decisión es *NP-Completo*. Para probar que  $\pi \in NPC$ , es suficiente y necesario probar que  $\pi \in NP$  y que existe  $\pi^* \in NPC$  tal que  $\pi^* \alpha_p \pi$ . Una propiedad importante de los problemas *NP-Completo* es que si  $\pi \in NPC$ , entonces  $\pi \in P$ , si y sólo si  $P=NP$ .

## 2.2 PROBLEMA DE RUTAS DE VEHÍCULOS Y SU VARIANTE CON VENTANAS DE TIEMPO

El ruteo de vehículos ha sido de gran interés por la comunidad científica en los últimos cincuenta años. Sin embargo, aún quedan preguntas por responder debido a su complejidad [Thangiah, 2003]. VRP, definido por [Dantzig, 1959], es un clásico problema de optimización combinatoria. Consiste en satisfacer un conjunto de clientes usando una flota de vehículos, respetando restricciones de vehículos, clientes, etc. Dentro de las variables más importantes del VRP se encuentra el VRPTW [Kallehauge, 2006]. Este problema consiste en encontrar la ruta óptima con una flota de vehículos desde un almacén a un número de clientes que deben ser visitados dentro de un intervalo de tiempo determinado, llamado ventana de tiempo. Un dígrafo  $G = (V, A, c, t, a, b, d, q)$  restringido por tiempo y capacidad es definido por un conjunto de nodos  $V = V_* \cup \{0, n+1\}$ , donde  $V_* = \{1, \dots, n\}$  es un conjunto de nodos cliente, y los nodos 0 y  $n+1$  son el almacén inicial y el almacén final respectivamente.

El conjunto de arcos  $A = A_* \cup \delta^+(0) \cup \delta^-(n+1)$ , donde  $A_* = A(V_*)$  es el conjunto de arcos tal que  $(i, j) \in V_*$ ,  $\delta^+(0) = \{(0, i) \mid i \in V_*\}$  es el conjunto de arcos dejando el almacén inicial, y  $\delta^-(n+1) = \{(i, n+1) \mid i \in V_*\}$  es el conjunto de arcos que entran al almacén final. Cada arco  $(i, j)$  tiene un costo  $c_{ij} \in \mathbb{Z}^A$ , tal que  $c_{ij} \leq c_{ik} + c_{kj}$  y tiene una duración de  $t_{ij} \in \mathbb{N}^A$ , tal que  $t_{ij} \leq t_{ik} + t_{kj}$ , para  $i, j, k \in V$ .

Los tiempos de inicio y final de las ventanas de tiempo en los nodos son denotados por  $a, b \in \{\mathbb{Z}_+ \cup \{+\infty\}\}^V$ , donde  $a_0 = a_{n+1} = 0$ ,  $b_0 = b_{n+1} = +\infty$ ,  $a_i \geq t_{0i}$ , y  $b_i \geq a_i$  para  $i \in V^*$  y  $b_j \geq a_i + t_{ij}$  para  $(i, j) \in A^*$ . Cada nodo  $i$  tiene una demanda  $d_i \in \mathbb{Z}_+^V$ , donde  $d_0 = d_{n+1} = 0$ , y una capacidad de carga  $q \in \mathbb{Z}_+$  donde  $q \geq d_i$  para  $i \in V^*$  y  $q \geq d_i + d_j$  para  $(i, j) \in A^*$ .

Para cada camino  $P = (v_1, \dots, v_k)$  en  $G$ , los tiempos de llegada del conjunto de nodos  $V(P)$  del camino se representan por el vector  $s \in \mathbb{Z}_+^{V(P)}$  definido por:  $s_{v_1} = a_{v_1}$ ,  $s_{v_i} = \max_{i=2, \dots, k} \{s_{v_{i-1}} + t_{v_{i-1}v_i}, a_{v_i}\}$ . La demanda del camino se representa por  $d(V(P))$  y su costo por  $c(A(P))$ .

Se dice que un camino  $P = (v_1, \dots, v_k)$  en  $G$  es factible si  $s_{v_i} \leq b_{v_i}$  para  $i \in V(P)$  y  $d(V(P)) \leq q$ . Una ruta factible  $R$  desde 0 a  $n+1$  en  $G$  es definida como  $R(0, v_2, \dots, v_{k-1}, n+1)$ . Se denota por  $R$  el conjunto de todas la rutas factibles desde 0 a  $n+1$  en  $G$ . Dado un dígrafo  $G$  restringido por tiempo y capacidad, encontrar una ruta con mínimo costo. VRPTW se expresa por la fórmula 2.1:

$$\min_{R \in R} \{c(A(R))\} \quad (2.1)$$

VRPTW es clasificado como un problema multiobjetivo, donde primeramente se intenta minimizar el número de vehículos utilizados y después la distancia recorrida y tiempos de espera requeridos para satisfacer a todos los clientes.

Un grupo de instancias reconocido por la comunidad científica para evaluar los algoritmos que abordan el VRPTW es el benchmark de [Solomon, 1987]. Este benchmark contiene instancias de tamaño 25,50 y 100 clientes, además de caracterizarse por su distribución geográfica según la tabla 2.1.

Tabla 2.1. Caracterización de las instancias de Solomon por su distribución geográfica.

Prefijo	Posfijo	Clase	Descripción
C	1	C1	Clientes distribuidos en cúmulos con ventanas de tiempo cortas y horizonte de calendarización corto.
C	2	C2	Clientes distribuidos en cúmulos con ventanas de tiempo amplias y horizonte de calendarización largo.
R	1	R1	Clientes generados aleatoriamente uniformemente distribuidos con ventanas de tiempo cortas y horizonte de calendarización corto.
R	2	R2	Clientes generados aleatoriamente uniformemente distribuidos con ventanas de tiempo amplias y horizonte de calendarización largo
RC	1	RC1	Clientes semi-agrupados (combinación de cúmulos y distribución aleatoria) con ventanas de tiempo cortas y horizonte de calendarización corto.
RC	2	RC2	Clientes semi-agrupados (combinación de cúmulos y distribución aleatoria) con ventanas de tiempo amplias y horizonte de calendarización largo.

### 2.3 COMPLEJIDAD DEL VRPTW

Siendo el VRP una generalización del problema TSP [Gómez, 2012], puede decirse que el primero posee la dificultad combinatoria del segundo [Kallehauge, 2006]. Suponiendo que la capacidad del vehículo es infinita, un VRP se simplifica al hecho de encontrar un circuito hamiltoniano, lo cual es la definición del TSP. Por otro lado, si la capacidad de los vehículos es finita, de tal forma que es imposible satisfacer a todos los clientes en un solo recorrido, el VRP puede representarse como un TSP Múltiple (MTSP, Multiple Traveling Salesman Problem), una instancia de MTSP puede ser transformada en su equivalente TSP añadiendo al grafo  $k-1$  (siendo  $k$  el número de rutas) copias adicionales del nodo centro de embarque y sus arcos incidentes (no existiendo arcos entre los  $k-1$  nodos y el centro de embarque) [Gutin, 2002].

Dado que TSP es NP-Duro [Garey, 1979], por extensión VRP también pertenece a la clase NP-Duro. Si el VRP clásico es NP-Duro, se induce que toda variante del VRP también lo es, igual complejidad tendría el VRPTW [Kallehauge, 2006]. Por lo tanto, se determina que el VRPTW pertenece a la clase NP-Duro.

## 2.4 ENFOQUES DE SOLUCIÓN

Existen básicamente dos tipos de enfoque en la solución de problemas de optimización: los métodos exactos y los heurísticos. A continuación se brinda una breve explicación de cada uno.

### 2.4.1 MÉTODOS EXACTOS

Los algoritmos exactos de solución para problemas de optimización más utilizados son la búsqueda exhaustiva, el método simplex de Dantzing , el método elipsoidal de Khachian y el método Karmarkar; el primero se basa en la creación y evaluación de todas las soluciones factibles, haciéndolo sumamente costoso, en términos computacionales, para problemas complejos, por lo que su aplicación se limita a instancias relativamente simples; el segundo realiza una búsqueda en los puntos adyacentes del cerco convexo del espacio de soluciones factibles del problema; el tercero crea un elipsoide que modifica en cuanto a volumen y posición basándose en las restricciones quebrantadas hasta que el centro del elipsoide converge a la región de soluciones factibles, en ese punto el centro elipsoidal constituye la solución óptima; el último método, considerado actualmente como el más eficiente de los cuatro, inicia en el centro del espacio de soluciones factibles y se mueve por los puntos internos de dicha región hacia la solución óptima.

A pesar de que todos estos métodos obtienen el resultado óptimo, su desempeño resulta inefectivo en problemas del mundo real. Debido a lo anterior se emplea una alternativa como los métodos heurísticos que si bien no siempre encuentran la mejor solución, son de gran ventaja por su bajo costo computacional.

## 2.4.2 MÉTODOS HEURÍSTICOS

Una heurística es un procedimiento simple, generalmente basado en el sentido común, que permite de manera perspicaz abordar un problema [Díaz, 1996]. Los algoritmos aproximados o heurísticos se pueden clasificar en dos tipos principales: algoritmos constructivos y algoritmos de búsqueda local. Los primeros se basan en generar soluciones desde cero añadiendo componentes a cada solución paso a paso. Un ejemplo bien conocido son las heurísticas voraces. Su gran ventaja es la velocidad: normalmente son muy rápidas y, además, a menudo devuelven soluciones razonablemente buenas. Sin embargo, no puede garantizarse que dichas soluciones sean óptimas con respecto a pequeños cambios a nivel local.

En consecuencia, una mejora típica es refinar la solución obtenida por la heurística voraz utilizando una búsqueda local. Los algoritmos de búsqueda local intentan repetidamente mejorar la solución actual con movimientos a soluciones vecinas. El caso más simple son los algoritmos de mejora iterativos.

Desafortunadamente, los algoritmos de mejora iterativos pueden estancarse en soluciones de baja calidad. Para permitir una mejora adicional en la calidad de las soluciones, la investigación en este campo en las últimas dos décadas ha centrado su atención en el diseño de técnicas de propósito general para guiar la construcción de soluciones y la búsqueda local mediante distintas heurísticas. Estas técnicas se llaman comúnmente metaheurísticas y consisten en conceptos generales empleados para definir métodos heurísticos. Dicho de otra manera, una metaheurística puede verse como un marco de trabajo general referido a algoritmos que puede aplicarse a diversos problemas de optimización combinatoria con pocos cambios significativos. De hecho, las metaheurísticas son ampliamente reconocidas como una de las mejores aproximaciones para abordar problemas de optimización combinatoria.

## 2.5 ESTRATEGIAS DE SOLUCIÓN

### 2.5.1 ALGORITMOS METAHEURÍSTICOS

#### SISTEMA DE COLONIA DE HORMIGAS (ACS)

En [Dorigo, 1991] se introduce por primera vez el metaheurístico de optimización basado en Colonia de Hormigas (ACO, Ant Colony Optimization), éste se inspira en el funcionamiento de las hormigas reales, las cuales encuentran el camino de su hormiguero a la comida por medio de una sustancia química llamada “feromona”. Las hormigas se mueven aplicando decisiones locales estocásticas que utilizan los rastros de la feromona (se denota por la letra  $\tau$ ) e información heurística (se denota por la letra  $\eta$ ).

Los rastros de feromona  $\tau$  son actualizados cuando una hormiga ha construido la solución, depositando cantidades de sustancia química por el camino que eligió, sin embargo dicho rastro también puede actualizarse después de haber recorrido el trayecto de un nodo a otro. Por otro lado, la información heurística  $\eta$  mide la predilección por recorrer el camino comprendido entre  $v_i$  y  $v_j$ . Ambos componentes dirigirán la búsqueda de futuras hormigas.

La figura 2.1 muestra el pseudocódigo de ACS general, en donde además de las actividades propias de las hormigas de recorrer los caminos; el ACS incluye dos procedimientos adicionales: evaporación y acciones demonio. La evaporación consiste en que la feromona depositada disminuya con el tiempo. Esto se dirige a evitar una rápida convergencia y favorecer la exploración de nuevas áreas del espacio de solución. Mientras que las acciones demonio son opcionales, siendo un ejemplo de ellas el procedimiento de búsqueda local [Glover, 2003].

Figura 2.1. Pseudocódigo de algoritmo metaheurístico de ACS general [Glover, 2003]

1	Procedimiento de metaheurístico ACS
2	Agenda de Hormigas
3	Administrar_actividad_hormigas( $\tau, \eta, \beta$ )
4	Evaporar_feromona ( $\tau$ )
5	Acciones_demonio ( ) { opcionales}
6	Fin Agenda de Hormigas.
7	Fin Procedimiento de metaheurístico ACS

En [Dorigo, 1996] se presenta una mejora para ACO, llamada Sistema de Colonia de Hormigas (ACS, Ant Colony System), ver figura 2.1, cuyas principales diferencias con su antecesor son:

- se modifica la regla de selección de clientes de tal manera que la importancia relativa de la información heurística denotada por  $\beta$  proporciona una manera directa de balance entre la exploración de nuevos arcos y la explotación a priori del conocimiento acumulado del problema.
- la regla de actualización global se aplica de forma general, también para aquellos arcos que pertenecen al mejor *tour* de la hormiga.
- mientras que las hormigas construyen una solución, es aplicada una regla de actualización local de la feromona.

## ALGORITMOS GENÉTICOS

Método de búsqueda que imita la teoría de la evolución biológica de Darwin para la resolución de problemas. Para ello, se parte de una población inicial de la cual se seleccionan los individuos más capacitados para luego reproducirlos y mutarlos para finalmente obtener la siguiente generación de individuos que estarán más adaptados que la anterior generación [Arranz, 2010].

Entre los operadores para este tipo de algoritmos se encuentran:

- Selección: elegir los individuos más capacitados para que éstos sean los que se reproduzcan de acuerdo con la teoría de Darwin, en la cual los más capacitados son los que deben sobrevivir y crear una nueva descendencia más facultada. Por lo tanto, una vez evaluado cada cromosoma y obtenida su puntuación, se crea la nueva población teniendo en cuenta los rasgos de los mejores individuos de la población actual.
  
- Cruza: consiste en el intercambio de material genético entre dos cromosomas. El objetivo del cruce es conseguir que el descendiente mejore la aptitud de sus padres. Algunos ejemplos de este operador son:
  - De un punto: Se copia la información genética de uno de los padres desde el inicio hasta el punto de cruce y el resto se copia del otro progenitor.
  
  - Uniforme: cada gen de la descendencia se crea copiando el correspondiente gen de uno de los dos padres, escogido de acuerdo a una "máscara de cruce" generada aleatoriamente. Cuando existe un 1 en la "máscara de cruce", el gen es copiado del primer padre, mientras que cuando exista un 0 en la "máscara de cruce", el gen se copia del segundo padre.

- **Mutación:** consiste en modificar ciertos genes de forma aleatoria con una probabilidad determinada.

### 2.5.2 BÚSQUEDA LOCAL

[Herrera, 2006] sostiene que a pesar de lo sofisticadas y aparentemente adecuadas que pueden resultar algunas metaheurísticas constructivas como ACS o los algoritmos genéticos; es altamente efectivo alternar una heurística de búsqueda local a partir de la última o mejor solución producida, en vez de ejecutar iterativamente sólo el método constructivo; por lo tanto, la posibilidad de añadir una búsqueda local resulta una alternativa prometedora. Tal tipo de mecanismos forman una clase general de heurísticas aproximadas o subóptimas, basadas en el concepto de mejora iterativa mediante la exploración de vecindades. La generación de un vecino se logra a través de un esquema  $\kappa$  que cambia un atributo o combina características de la solución actual.

La búsqueda local efectúa una exploración miope, pues sólo acepta soluciones secuenciales originadas por su diseño; la Figura 2.2 generaliza este concepto, sea  $\kappa$  el esquema de vecindad,  $s$  una solución factible y  $\chi s$  la función de aptitud de la solución.

Figura 2.2. Procedimiento de Búsqueda Local

Procedimiento <i>Búsqueda Local</i> ( )	
1	Crear solución factible $s$
2	Sea $K$ el esquema de vecindad.
3	<b>Repeat</b>
4	Buscar en el vecindario, $K(s)$
5	<b>if</b> se encuentra $s'$ con $f(s') < f(s)$ <b>then</b>
6	$s \leftarrow s'$
7	<b>endif</b>
8	<b>until</b> no se encuentren soluciones $s'$ factibles con menor costo

## BÚSQUEDA $K$ -OPT

Para introducir mayor flexibilidad al modificar un ciclo hamiltoniano, se puede considerar dividirlo en  $k$  partes, en lugar de dos, y combinar los caminos resultantes de la mejor manera posible. Se le llama movimiento  $k$ -opt a tal modificación.

Al aumentar  $k$  de igual manera aumenta el tamaño del entorno y el número de posibilidades a examinar en el movimiento, tanto por las posibles combinaciones para eliminar las aristas del ciclo, como por la reconstrucción posterior [Martí, 2006].

## 2.6 ANÁLISIS DE DATOS

Existen dos enfoques principales para el estudio visual de los datos y el análisis de la información obtenida. Esto se lleva a cabo a través de las técnicas exploratorias o bien con técnicas confirmatorias [Parra, 2002].

[Vilar, 2005] El objetivo en el primer tipo de técnicas es la organización, representación y descripción de los mismos, así como los pasos necesarios para extraer la información que contienen. Las técnicas más utilizadas son las siguientes:

- Diagrama de tallo y hojas (método semigráfico).
- Tablas resumen de 3 y 5 números (índices descriptivos).
- Gráfico de caja (método gráfico).

Ante un conjunto de datos, el enfoque exploratorio recomienda iniciar con gráficos que permitan visualizar su estructura. Para datos cuantitativos es aconsejable comenzar con un histograma [Sánchez, 1990]. Las técnicas de carácter confirmatorio utilizan estadísticos numéricos de resumen generados a partir del empleo de un modelo, definido *a priori*, para confirmar o no una hipótesis.

### 2.6.1 *MODELOS ESTADÍSTICOS*

Un modelo estadístico es una distribución de probabilidad construida para hacer inferencias y tomar decisiones a partir de datos [Davison, 2003]. Del mismo modo puede definirse como una formalización de las relaciones entre variables en forma de ecuaciones matemáticas [Ader, 2008]. Un modelo estadístico es un caso especial del modelo estocástico donde los parámetros son desconocidos, dichos parámetros se estiman a partir de datos empíricos obtenidos a través de procedimientos estadísticos tales como diseño de experimentos o estudios observacionales [Merli,2008].

#### CLASIFICACIÓN DE LOS MODELOS ESTADÍSTICOS

En la literatura estudiada se presentan diversas clasificaciones de los modelos estadísticos. Por lo tanto aquí son mostradas las siguientes clasificaciones, con el objetivo de identificar las características de dichos modelos.

- Tipos de efectos: La distinción entre efectos está relacionada con la naturaleza de los parámetros desconocidos del modelo.
  - Fijos: El modelo de efectos fijos se aplica a situaciones en las que el experimentador ha sometido al grupo o material analizado a varios factores, cada uno de los cuales le afecta sólo a la media, permaneciendo la "variable respuesta" con una distribución normal. Este modelo se utiliza cuando el investigador se interesa únicamente por los niveles del factor presentes en el experimento, por lo que cualquier variación observada en las puntuaciones se deberá al error experimental.

- Aleatorios: estos modelos se usan para describir situaciones en que ocurren diferencias incomparables en el material o grupo experimental. El ejemplo más simple es el de estimar la media desconocida de una población compuesta de individuos diferentes y en el que esas diferencias se mezclan con los errores del instrumento de medición. Este modelo se supone cuando el investigador está interesado en una población de niveles, teóricamente infinitos, del factor de estudio, de los que únicamente una muestra al azar ( $t$  niveles) están presentes en el experimento.
  - Mixtos: algunos efectos son fijos y otros aleatorios.
- Número de variables de respuesta: como su nombre lo dice, la cantidad de variables que son utilizadas para interpretar el resultado del modelo.
- Univariados: Muestran aisladamente característica por característica, exhiben una única respuesta.
  - Multivariados: Presentan las relaciones entre conjuntos de variables dependientes y los individuos para los cuales se han medido dichas variables. A partir de una muestra de variables observadas en una muestra de elementos de una población, se busca:
    - Resumir la información en pocas variables
    - Analizar dependencias e independencias
    - Agrupar y clasificar

➤ Tipo de modelo obtenido

- Descriptivos: el objetivo de estos modelos es la generación de hipótesis. Los modelos exploratorios son un ejemplo de ellos.
- Analíticos: tienen como finalidad la confirmación de hipótesis. Los modelos predictivos y explicativos (causales) son ejemplos de modelos analíticos.

Como se observa en tabla 2.2, el análisis causal es una alternativa que brinda explicaciones para dar solución a un problema determinado, éste hace uso de herramientas como diagramas y reglas de asociación para dar solución al problema.

*Tabla 2.2. Clasificación de multivariados por método obtenido.*

<b>Técnica</b>	<b>Objetivo</b>	<b>Carácter</b>
Componentes principales Factores comunes Agrupación Correlación	Reducir Reducir Reducir Reducir	Descriptivo
Regresión lineal Regresión logística Análisis discriminante Análisis causal	Relacionar-Predecir Relacionar-Predecir Clasificar Explicar	Analítico

### 2.6.2 CORRELACIÓN

Número que indica el grado de relación que existe entre dos variables. En términos de probabilidad se expresa que una de las variables se produzca (o no) cuando ocurra (o no ocurra) la otra. La correlación no indica que una variable cause la otra. Dos variables presentan correlación si una de ellas se produce de modo más (o menos) frecuente al producirse la otra. La correlación es positiva si ambas variables tienden a aumentar juntas o a disminuir juntas; es negativa si una variable tiende a aumentar cuando la otra disminuye, y es cero si no hay una conexión evidente [Stassen, 2007].

En el mismo sentido, [Moore, 2005] la expresa como el coeficiente que mide la fuerza y la dirección lineal entre dos variables cuantitativas. Se simboliza con la letra  $r$ . Suponer que se tienen datos de dos variables  $x$  e  $y$  para  $n$  individuos. Los valores del primer individuo se denotan por  $x_1$  e  $y_1$ , para el segundo por  $x_2$  e  $y_2$ . Las medias y desviaciones estándar de las dos variables son  $\bar{x}$  y  $s_x$  para los valores de  $x$ , e  $\bar{y}$  y  $s_y$  para los valores de  $y$ . La correlación entre  $x$  e  $y$  es:

$$r = \frac{1}{n-1} \sum \left( \frac{x_i - \bar{x}}{s_x} \right) \left( \frac{y_i - \bar{y}}{s_y} \right) \quad (2.2)$$

### 2.6.3 ANÁLISIS DE VARIANZA Y FACTORIAL

Desarrollada por R. Fisher, el análisis de varianza es una técnica que permite se realicen inferencias sobre las medias poblacionales, cuando éstas se ven afectadas por diversos factores. El interés al aplicar el análisis de varianza es explicar los valores de la variable dependiente en términos de las condiciones experimentales, considerando un rango de error. Se pretende determinar bajo qué condición experimental difieren las medias de la variable dependiente [Rutherford, 2012].

Por otro lado, el análisis factorial es una técnica matemática y estadística basada en correlación. En ella se evalúa un conjunto inicial de atributos de entrada con el objetivo de identificar relaciones entre ellos y para conseguir un número reducido de factores en términos de los valores originales que mejor representen a los datos [Nettleton, 2012].

En este tipo de análisis, toda combinación posible entre los niveles de las variables es evaluada en cada réplica del experimento. Cuando las variables son estructuradas en diseño factorial, se les llama variables cruzadas, mientras que el efecto de una variable se define como el cambio en respuesta por la modificación del nivel de dicha variable [Montgomery, 2008].

# Capítulo 3

## ESTADO DEL ARTE

Este capítulo incluye los trabajos relacionados que han abordado algoritmos de solución que resuelven el Problema de Rutas de Vehículos con Ventanas de Tiempo.

### 3.1 TRABAJOS QUE EXPERIMENTAN CON INSTANCIAS DE SOLOMON

- [Thangiah, 2010] describe una heurística llamada GIDEON. Esta consiste en un método de agrupamiento global de clientes y un método local de post-optimización. El método de agrupamiento global utiliza una estrategia de búsqueda adaptativa basada en poblaciones genéticas, con el fin de asignar los vehículos a los clientes. La mejor solución obtenida del método de agrupamiento es mejorada por el método local de post-optimización.
- [Prescott, 2009] presentan una Búsqueda en Gran Vecindad donde la heurística con el método de Ramificación y Hojas (*Branch and Price*) para la búsqueda en la exploración de soluciones vecinas.
- [Hashimoto, 2008] sugieren una enfoque de solución de re-vinculación de rutas. Inicialmente, se construye un conjunto de soluciones generadas al azar, mientras que una búsqueda local iterativa mejorada hace uso de los vecindarios de 2-Opt, CROSS-Exchange y Or-Opt.
- [Hoshino, 2007] desarrolla una búsqueda local heurística controlado por el principio dinámico explosivo caótico de las redes neuronales. La búsqueda caótica utiliza movimientos de Relocalización e Intercambio local. Para controlar estos movimientos locales un conjunto de neuronas es definida. A cada neurona se le asigna una ganancia, un reflejo refractario y un efecto mutuo.

- [Pisinger, 2006] proponen un enfoque adaptativo que extiende el concepto de Búsqueda de Vecindarios Grandes (LNS), propuesto por [Shaw, 1998]. Este algoritmo adaptativo hace uso de un conjunto de vecindarios destructivos/constructivos parciales que compiten para modificar la solución actual de manera iterativa.
- [Mester, 2006] propone una Estrategia Evolutiva (ES) de múltiples parámetros. Al principio, cinco soluciones son generados mediante un algoritmo de ahorro heurístico híbrido en paralelo, los mejores resultados forman la población inicial. Durante este proceso evolutivo, una nueva solución es generada mediante el operador en mutación de remover-insertar de múltiples parámetros y el padre es remplazado si alguna mejora es observada.

### **3.2 TRABAJOS RELACIONADOS CON VRPTW EN EL ITCM**

[Cámara, 2011] emplea una variante del Sistema de Colonia de Hormigas (ACS) que busca optimizar ambos objetivos del VRPTW al coordinar las actividades de dos colonias de hormigas (MACS-VRPTW). El objetivo de la primera colonia, ACS-VEI disminuye el número de vehículos usados, mientras una segunda colonia ACS-TIME optimiza la distancia recorrida encontrada por ACS-VEI. Ambas colonias usan rastros de feromona independiente pero colaboran al compartir una variable de solución global administrada por el MACS-VRPTW.

[Arrañaga, 2007] emplea un ACS en el cual las hormigas construyen incrementalmente una solución, moviéndose en forma simultánea y de manera asíncrona a través de los estados adyacentes del problema, dicho movimiento es dirigido por una regla de transición pseudoaleatoria. Para la Búsqueda Local obtiene mejores resultados con la utilización de la estrategia de aprendizaje por niveles en combinación con el operador de intercambio de ejes 3-Opt [Bock, 1958], *Cross-Exchange* [Taillard, 1997] y Ramificación y Poda (*Branch and Bound*).

[Herrera, 2006] emplea un ACS usando la técnica de Lista Restringida en conjunto de una Búsqueda Local con los operadores de intercambio de ejes: *3-Opt* y *Cross-Exchange* operando sobre una y dos rutas respectivamente.

### 3.3 ANÁLISIS DEL ESTADO DEL ARTE DE VRPTW

El problema VRPTW ha sido ampliamente estudiado por la comunidad científica debido a la importancia que tiene como problema de aplicación [Cámara, 2011]. La tabla 3.1 muestra abreviaturas para la tabla 3.2 y 3.3, además de mostrar el algoritmo utilizado. En la tabla 3.2 se muestra una comparación de resultados de los algoritmos propuestos en el estado del arte, mientras que la tabla 3.3 exhibe los resultados obtenidos en los trabajos realizados en el ITCM. Asimismo se incluye la tabla 3.4 la cual muestra las estrategias utilizadas por dichos algoritmos.

*Tabla 3.1: Abreviatura para tabla 3.2 y 3.3 mostrando el nombre del autor, algoritmo utilizado y año de publicación de la investigación correspondiente.*

<b>Abreviatura</b>	<b>Autor</b>	<b>Algoritmo</b>	<b>Año</b>
C	Cámara	Sistema de múltiples colonias de hormigas	2011
TH	Thangiah	Heurística de agrupamiento global de clientes y un método local de post-optimización	2010
PDR	Gagnon,Dassaulniers y Rousseau	Búsqueda en gran vecindad con ramificación y hojas	2009
HN	Hoshino	Búsqueda local con heurística controlada por el principio dinámico-explosivo-caótico de las redes neuronales.	2007
A	Arrañaga	Sistema de colonia de hormigas con aprendizaje por niveles	2007
PR	Pisinger y Ropke	Búsqueda adaptativa en gran vecindad	2006
M	Mester	Estrategia Evolutiva	2006
HR	Herrera	Sistema de colonia de hormigas con lista restringida	2006

Tabla 3.2. Mejores resultados logrados en el estado del arte que experimentan con Solomon por cada tipo de instancia de [Solomon, 1987].

Tipo	Trabajo	Vehículos	D. Recorrida	Corridas	T.Ejecución
C1	HN	10	828.36	10	146
C2	HN	3	589.86	10	146
R1	PDR	11.92	1210.3	Desconocido	Desconocido
R2	M	2.73	954.27	Desconocido	7200
RC1	M	11.5	1384.2	Desconocido	7200
RC2	PR	3.25	1108.5	Desconocido	Desconocido

Tabla 3.3. Mejores resultados logrados en el ITCM por cada tipo de instancia de [Solomon, 1987]

Tipo	Work	Vehículos	D. Recorrida	Corridas	T.Ejecución
C1	ARA	10	830.72	5	1800
C2	C	3	597.53	5	1800
R1	C	12.58	1266.9	5	1800
R2	C	3	982.85	5	1800
RC1	C	12.125	1416.7	5	1800
RC2	HR	3.37	1171.7	5	1800

Tabla 3.4. Análisis de las estrategias utilizadas en trabajos del estado del arte que experimentan con Solomon.

Autor	Estrategias							
	Búsqueda en gran vecindad	Estrategia evolutiva	Vecinos Cercanos	Búsqueda Local	Lista Restringida Adaptativa	Aprendizaje por Niveles	Lista Restringida Extendida	Regla de aprendizaje de estrategias
[Mester,2006]		✓		✓				
[Pisinger, 2006]	✓			✓				
[Hoshino, 2007]				✓				
[Prescott, 2009]	✓			✓				
[Thangiah, 2010]				✓				

Tabla 3.5. Análisis de las estrategias utilizadas en trabajos del ITCM que experimentan con Solomon.

Autor	Estrategias						
	Vecinos Cercanos	Búsqueda Local	Lista Restringida Adaptativa	Aprendizaje por Niveles	Lista Restringida Extendida	Regla de aprendizaje de estrategias	
[Herrera, 2006]	✓	✓	✓				
[Arrañaga, 2007]	✓	✓		✓			
[Camara,2011]	✓	✓	✓	✓	✓		

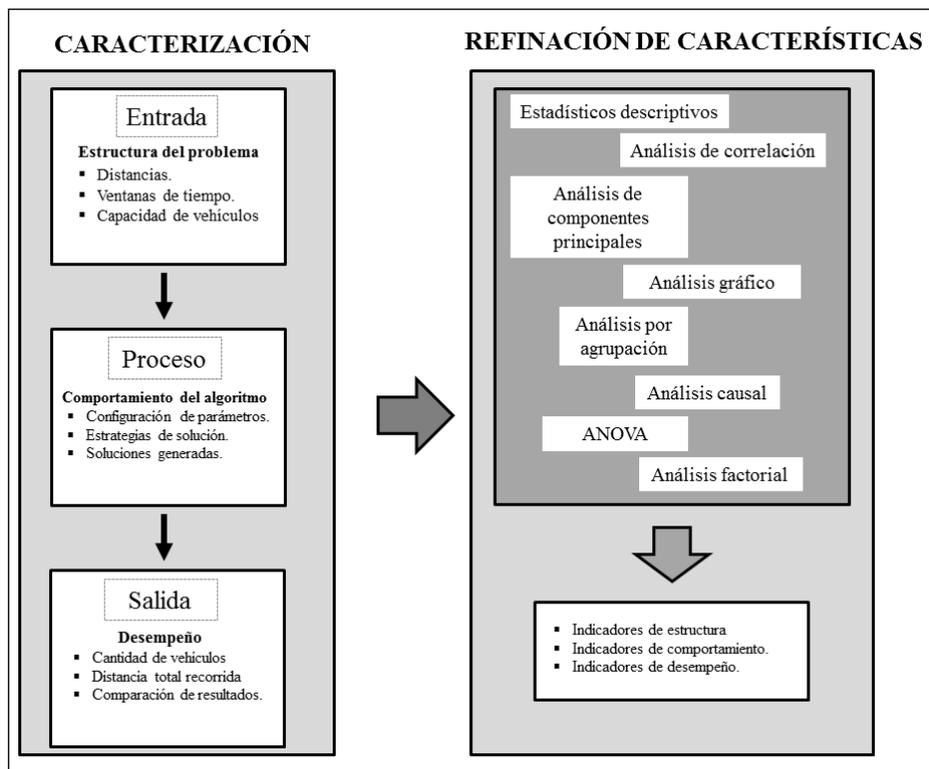
# Capítulo 4

## METODOLOGÍA

### 4.1 ANÁLISIS EXPERIMENTAL

La metodología consta de dos etapas (ver figura 4.1), una de ellas es la *caracterización*, que tiene como objetivo principal identificar, en cada una de las fases del proceso de optimización los factores de desempeño relevantes y factibles de medición. Estos factores son caracterizados a través de funciones de caracterización (métricas) que proporcionan información útil para describir el desempeño algorítmico.

Figura 4. 1: Esquema de metodología propuesta

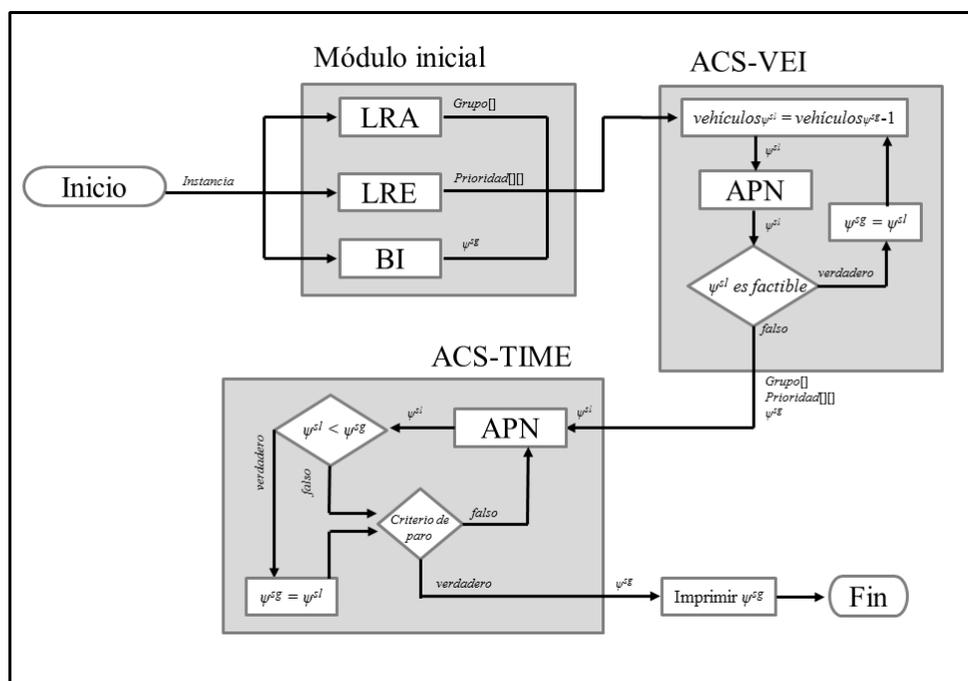


La segunda etapa en figura 4.1 corresponde a la *refinación de características*, las métricas definidas en la etapa de caracterización son analizadas mediante técnicas exploratorias con la finalidad de descartar variables redundantes o irrelevantes. En caso de ser necesario, nuevas variables son definidas mediante el uso de técnicas de análisis multivariado.

## 4.2 CARACTERIZACIÓN DEL COMPORTAMIENTO DEL ALGORITMO

Los algoritmos metaheurísticos incluyen estrategias que tratan de ajustar el proceso de solución al problema de optimización que resuelven. Para estos algoritmos, la medición de funciones de caracterización debe estar ligada directamente con las técnicas heurísticas utilizadas [Quiroz, 2009]. En esta investigación se continuó con lo realizado por [Cámara, 2011], donde se emplea el algoritmo MACS-VRPTW que coordina las actividades de dos colonias [Gambardella, 1999]. El objetivo de la primera colonia, ACS-VEI disminuye el número de vehículos usados, mientras una segunda colonia ACS-TIME optimiza la distancia recorrida encontrada por ACS-VEI. Ambas colonias usan rastros de feromona independiente pero colaboran al compartir una variable de solución global administrada por el MACS-VRPTW. El aspecto general del algoritmo puede observarse en la figura 4.2.

Figura 4.2 Esquema general de solución de [Cámara,2011].



Dentro de ambas colonias se llama al módulo APN (Aprendizaje por niveles), el cual se observa en la figura 4.3 y dentro de dicho modulo se llama a otro modulo llamado *Construcción de  $\psi^{sl}_{numgnumh}$*  mostrado en la figura 4.4.

La figura 4.3 muestra cómo se guarda la información del rastro de la feromona a través del proceso de construcción de la solución parcial (subcapítulo 4.2.4), mientras que la figura 4.4 presenta de forma general como la hormiga construye la solución parcial, además del momento en que se calcula la información heurística, se selecciona el siguiente cliente (subcapítulo 4.2.5) y se ejecuta la búsqueda local durante el algoritmo (subcapítulo 4.2.3).

Figura 4.3 Modulo APN de esquema general de solución de [Cámara, 2011].

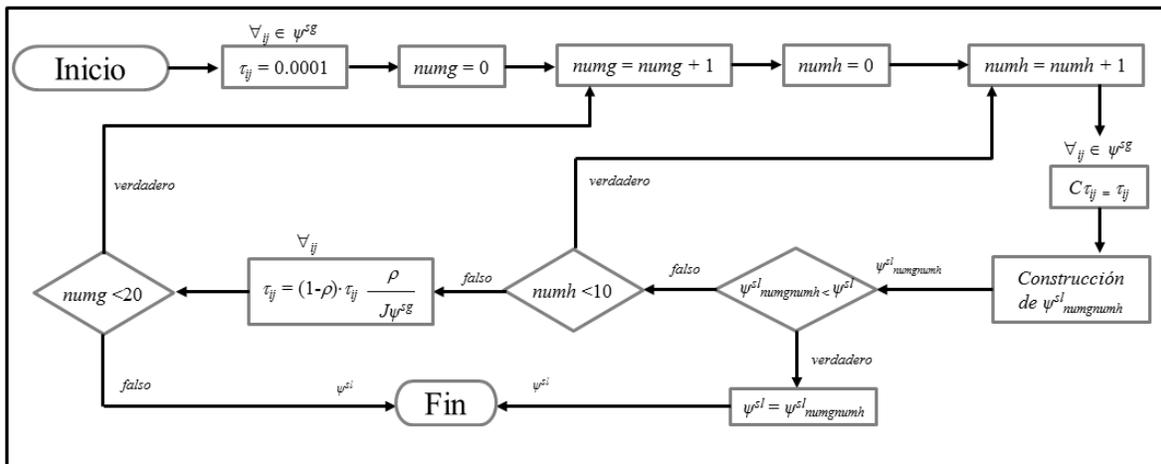
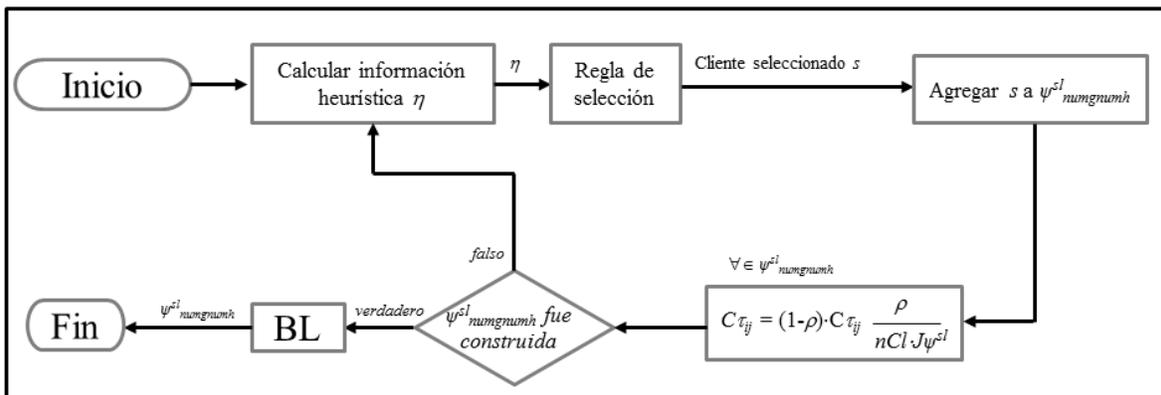
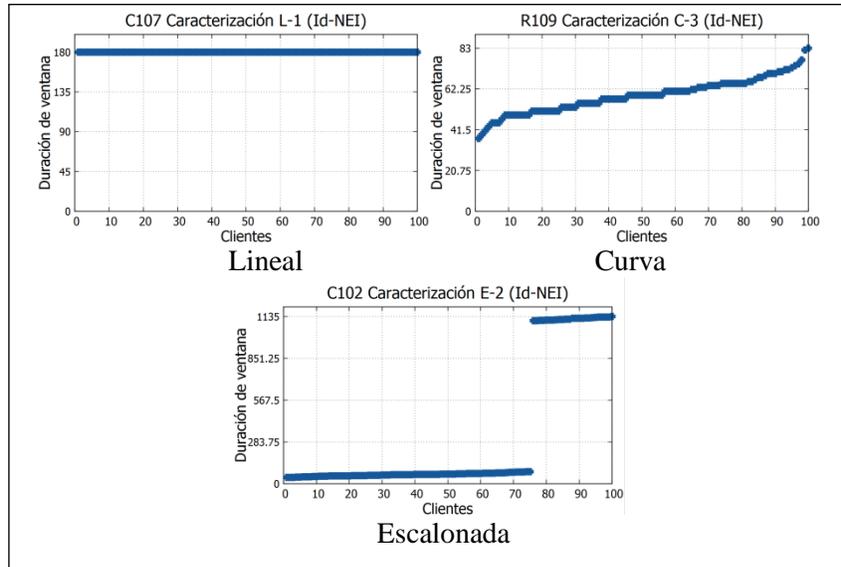


Figura 4.4 Modulo construcción de  $\psi^{sl}_{numgnumh}$  de esquema general de solución de [Cámara, 2011].



Además del algoritmo, las instancias también fueron caracterizadas por [Cámara, 2011] por medio de su distribución de ventanas de tiempo. Dicha caracterización arrojó lo siguiente:

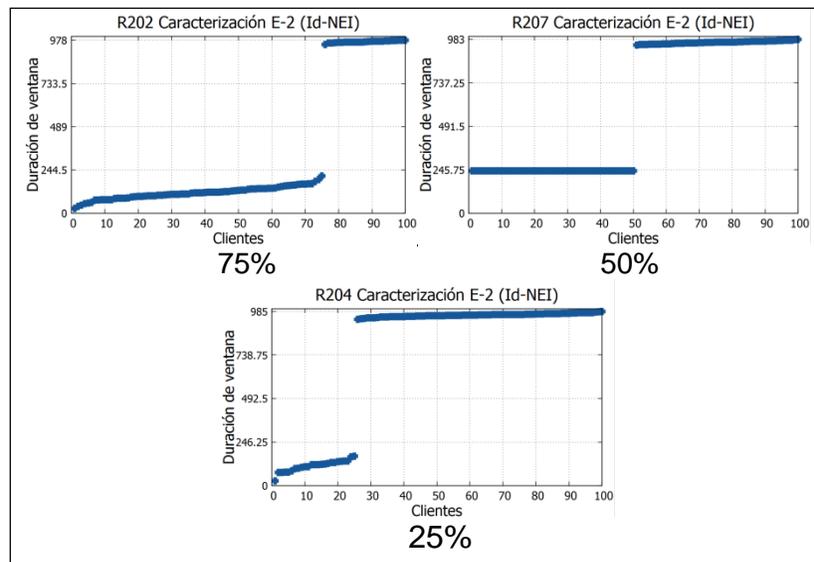
Figura 4.5. Caracterización de las instancias de [Solomon, 1987] en base a su ventana de tiempo.



[Cámara, 2011] también reporta una sub división de las instancias escalonadas mostrada a continuación:

- Escalonada al 25%, Escalonada al 50% y Escalonada al 75%

Figura 4.6. Caracterización de las instancias escalonadas de [Cámara, 2011].



#### 4.2.1 BÚSQUEDA INICIAL (BI)

MACS-VRPTW emplea dos enfoques para realizar la construcción de soluciones, la explotación, que hace uso de la información adquirida por las hormigas, y la exploración, un procedimiento semi-aleatorio basado en una distribución de probabilidad proporcional con base en la preeminencia de la feromona e información heurística. Debido a que el enfoque explotativo es sumamente similar a una búsqueda avara, la posibilidad de usarlo como una búsqueda inicial redundante en soluciones de buena calidad. Dicha búsqueda se realiza mediante:

- Búsqueda de Vecinos Cercanos NNS: dado un conjunto  $S$  de  $n$  puntos en un espacio métrico  $X$ , la tarea es pre procesar dichos punto tal que, desde un punto  $q \in X$ , sea obtenido el punto más cercano a  $q$  [Arya,1994].

#### 4.2.2 HEURISTICAS DE CARACTERIZACIÓN

Como se observa en figura 4.2, son utilizadas diversas heurísticas para auxiliar en el proceso de selección de clientes.

#### LISTA RESTRINGIDA ADAPTATIVA (LRA)

De acuerdo con [Dorigo, 1996], el uso de una Lista Restringida de Candidatos por parte de las hormigas en el proceso constructivo, resulta ventajoso debido a la condición restrictiva en la creación de soluciones factibles y las distintas posibilidades de distribución de los clientes. Una Lista Restringida de Candidatos limita la población global del momento en el proceso de selección del siguiente nodo, a un subconjunto que cumple con cierto criterio previamente especificado. Debido a las variaciones de la densidad geográfica en diferentes puntos del espacio de una misma instancia, la definición de regiones con mínima distancia entre los puntos pertenecientes a dicha región permite restringir la selección del siguiente punto de visita. Para construir dicha lista fueron utilizados algoritmos basados en técnicas de agrupación.

[Herrera, 2006] en su tesis define la agrupación jerárquica a través de los siguientes seis pasos:

- Se genera un árbol de expansión mínimo (MST) que incluya todos los clientes y el almacén de la instancia. Este MST es un sub grafo que conecta todos los clientes a un costo mínimo.
- Desviación estándar  $\sigma$  es calculada de  $tc$  (costo de árbol), donde cada  $tc_{ij}$  es la distancia del cliente  $i$  al cliente  $j$ ,  $(i,j) \in MST$ .
- Porcentaje de variabilidad  $\theta$  de los costos asociados de cada arco perteneciente a MST es calculado por la fórmula 4.1, la cual normaliza  $\sigma$

$$\theta = \frac{\sigma}{2(\max\{tc_{ij}\} - \min\{tc_{ij}\})}, (i, j) \in MST \quad (4.1)$$

En caso que  $\theta < 0.1$  (muy baja variabilidad), la localización de los clientes de la instancia se aproxima a una distribución uniforme, entonces la población constituye un solo grupo. En sentido contrario si  $\theta \geq 0.1$ , esto revela la posible existencia de regiones de diferente densidad.

- La formación del conjunto de grupos  $H$  se lleva a cabo si se satisface la regla de agrupamiento  $\theta \geq 0.1$ , en caso contrario se considera que los clientes pertenecen a un solo grupo  $|H| = 1$ .
- Satisfecha la regla de agrupamiento, cada cliente comienza en su propio grupo; pares de grupos son unidos iterativamente con objetivo de encontrar un conjunto de clientes  $H$  apropiado, basándose para dicho proceso en un umbral de aceptación  $\omega$ . El valor inicial para  $\omega$  se calcula con la ecuación 4.2

$$\omega = 2 \cdot \max\{tc_{ij}\}, (i, j) \in MST \quad (4.2)$$

Para cada par de grupos  $(h_i, h_j)$ , donde  $h_i, h_j \in H$ , se calcula la distancia de Mahalanobis  $dM$ . Esta se usa para la regla: si  $dM(h_i, h_j) < \omega$  y  $h_i \neq h_j$ , entonces  $h_i \cup h_j$ . Dicha regla determina si dos grupos son mezclados en uno solo.

- La formación de grupos continua mientras operaciones de mezcla existan con el mismo valor de  $\omega$ , en caso contrario  $\omega$  es modificado por  $\omega = \omega \cdot (1 + \theta)$ , finalizando si no hubo operaciones de mezcla con este nuevo valor de  $\omega$ .

#### LISTA RESTRINGIDA EXTENDIDA (LRE)

Esta heurística se ejecuta sólo si la distribución de ventana de tiempo de la instancia resulta escalonada. Para este tipo de instancias, se lleva a cabo una clasificación de clientes, tal que; cuando las hormigas hagan la búsqueda de una nueva solución, los clientes con ventana de tiempo corta tengan mayor prioridad de visita. [Cámara, 2011] define la heurística mediante los siguiente cinco pasos:

- Calcular la longitud de ventana de tiempo para cada cliente de la instancia. Ver fórmula 4.3.

$$tw_i = b_i - a_i \quad (4.3)$$

donde  $tw_i$  es la longitud de ventana de tiempo del cliente  $i$ ,  $a_i$  y  $b_i$ , son inicio y fin de ventana respectivamente.

- Calcular media ( $\mu$ ) (fórmula 4.4) y la desviación estándar ( $\sigma$ ) (fórmula 4.5) de la duración de las ventanas de tiempo de la instancia.

$$\mu = \sum_{i=1}^{|V|} tw_i \quad (4.4)$$

$$\sigma = \sqrt{\frac{1}{|V|} \cdot \sum_{i=1}^{|V|} (tw_i - \mu)^2} \quad (4.5)$$

- En caso de que fórmula 4.6 sea verdadera, se identifica a la instancia como escalonada, por lo que procede con el siguiente paso. En caso contrario finaliza el algoritmo.

$$\sigma > \mu \quad (4.6)$$

- La importancia de cada cliente se denota por  $vt_i$ . Dicha importancia se define por la fórmula (4.7).

$$vt_i = \begin{cases} 1, & \text{si } tw_i < \mu \\ 0, & \text{otro caso} \end{cases} \quad (4.7)$$

Se asume con importancia alta a un cliente cuyo tamaño de ventana sea menor a la media ( $\mu$ ), y por lo tanto asignando  $vt_i = 1$ . En caso contrario, el cliente se asume con importancia baja y se asigna con  $vt_i = 0$ .

- Determinar  $priority_{lm}$  para cada par  $(l, m)$ , donde  $l$  es el cliente actual y clientes  $m$  y  $k$  son clientes aún no asignados a la solución (camino). Ver fórmula (4.8)

$$priority_{lm} = \begin{cases} 3, & \text{si } vt_l = 1, vt_m = 1, wait_{lm}^1 < \lambda_1 \\ 2, & \text{si } vt_l = 0 \parallel vt_m = 0, wait_{lm}^1 \geq \lambda_1, \\ & wait_{lm}^2 < \lambda_1, de_{lk} < \lambda_2 \\ 0, & \text{otro caso} \end{cases} \quad (4.8)$$

donde  $wait_{lm}^2 = a_m - (a_l + s_l + de_{lk} + s_k)$ ,  $wait_{lm}^1 = (a_m - a_l + s_l) - de_{lm}$ ,  $sl$  es el tiempo de servicio del cliente  $l$ ,  $de_{lm}$  es la distancia euclidiana entre  $l$  y  $m$  mientras que  $\lambda_1$  y  $\lambda_2$  son umbrales de espera. Los niveles de prioridad dos y tres son únicamente para instancias escalonadas. Nivel tres es para un par de clientes  $(l, m)$  ambos con importancia alta ( $vt_l = 1$  y  $vt_m = 1$ ) y con un tiempo de espera aceptable  $wait_{lm}^1$  delimitado por  $\lambda_1$  entre los clientes  $l$  y  $m$ . Por otro lado, los pares de clientes con nivel dos tienen un tiempo de espera aceptable  $wait_{lm}^2$  también delimitado por  $\lambda_1$ , además de una distancia euclidiana aceptable  $de_{lk}$  entre cliente  $l$  y cliente intermedio  $k$  delimitada por  $\lambda_2$ .

### 4.2.3 BÚSQUEDA LOCAL (BL)

Para la incorporación de la Búsqueda Local al ACS se eligieron los esquemas de intercambio de ejes: 3-opt [Bock, 1958] y Cross-Exchange [Taillard, 1997], operando sobre una y dos rutas respectivamente, ambos incluyen implícitamente otros operadores más sencillos, el primero contiene por naturaleza al 2-opt [Croes, 1958], el segundo permite el uso de segmentos vacíos y comprende movimientos de tipo 2-opt\*, relocalización e intercambio, haciéndolo sumamente versátil [Potvin, 1995] [Prosser, 1996]. Las Figuras 4.7 y 4.8 muestran gráficamente cada uno de los operadores de búsqueda local.

Figura 4.7 Esquema 3-opt, se muestran las ocho posibles combinaciones que ofrece el operador 3-opt, dicho operador sólo puede actuar sobre una misma ruta.

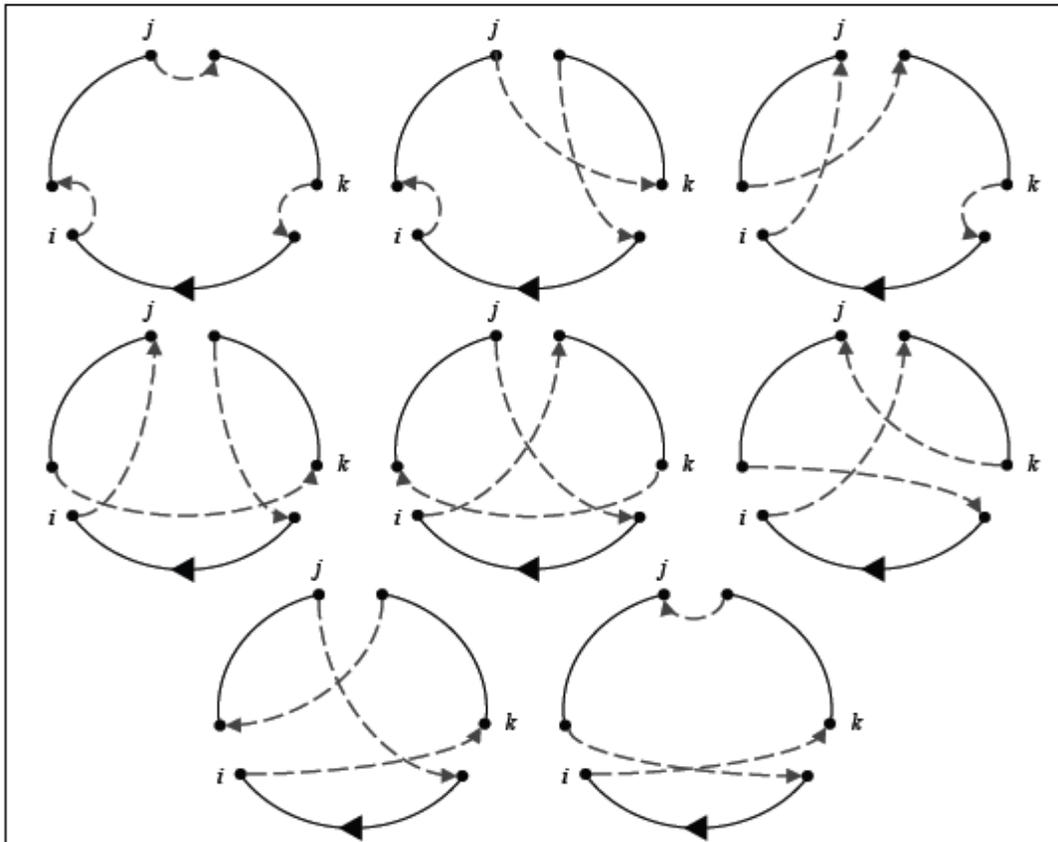
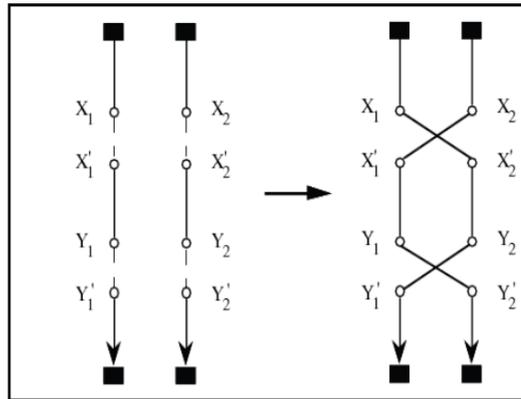


Figura 4.8. Esquema Cross-Exchange, éste operador actúa sobre dos rutas, donde la longitud de los segmentos que interviene de ambas rutas puede variar desde cero hasta tres nodos, este enfoque añade una gran versatilidad al esquema.



#### 4.2.4 APRENDIZAJE POR NIVELES (APN)

[Arrañaga, 2007] propone una heurística de actualización de valores para la tabla de feromonas utilizadas por el ACS, esta es parte de la estrategia de actualización de valores del algoritmo DQL propuesto por [Mariano, 2001] donde se observan dos niveles de conocimiento por parte del algoritmo.

- En un primer nivel se tienen los valores originales de la tabla de feromona que solamente contienen las mejores soluciones obtenidas por las hormigas y sus valores son modificados con la actualización global de feromona natural del ACS, ver fórmulas (4.9) y (4.10).

$$\tau_{ij} = (1 - \rho) * \tau_{ij} \quad (4.9)$$

Posteriormente procede actualizar las  $n - 1$  parejas de la solución en  $\tau$ .

$$\tau_{ab} = \frac{\rho}{fDist} + \tau_{ab} \quad (4.10)$$

Donde  $(a, b)$  es una pareja de cliente  $a$  con cliente  $b$ , la cual pertenece a la solución actual,  $fDist$  es la distancia recorrida por la mejor solución.

- En un segundo nivel se tiene una copia de la tabla que contiene los valores locales de feromona ya que solo es modificada por la actualización local de las hormigas. Ver fórmula (4.11)

$$C\tau_{ab} = (1 - \rho) * C\tau_{ab} + \frac{\rho}{nCl * fDist} \quad (4.11)$$

Donde  $C\tau$  una copia de  $\tau$ ,  $a$  es cliente actual,  $b$  es cliente siguiente,  $fDist$  es la distancia recorrida en la mejor solución, y  $nCl$  es número de clientes visitados hasta el momento por la solución actual.

De igual forma esta heurística conserva los valores de  $\tau$  a través de las generaciones, lo cual permite a las hormigas iniciar desde un estado previo de búsqueda y no desde un estado inicial.

#### 4.2.5 SISTEMAS DE COLONIA DE HORMIGAS

##### SELECCIÓN DE CANDIDATOS

Una vez que la membresía de cada cliente a un grupo fue asignada según su distribución topológica (LRA) y la prioridad de cada cliente con respecto del resto de clientes fue definida según la distribución de ventana de tiempo. Se procede a la selección de candidatos según la regla de selección en fórmula (4.12). La cual actualiza tanto la tabla de feromonas  $\tau$  como la información heurística  $\eta$ .

$$s = \begin{cases} \text{argmax} \{ \tau_{ij} \cdot \eta_{ij}^\beta \}, & \text{si } (q \leq q_0) \\ S, & \text{en otro caso} \end{cases} \quad (4.12)$$

$$S = f(\text{prob}(i, j)) \text{ donde } (\text{prob}(i, j)) = \frac{\tau_{ij} \cdot \eta_{ij}^\beta}{\sum_{k \in N(i)} \tau_{ik} \cdot \eta_{ik}^\beta} \quad (4.13)$$

Donde  $\beta$  es la importancia relativa de la información heurística,  $N(i)$  es el conjunto de clientes vecinos disponibles y  $S$  es la variable aleatoria dada por la fórmula (4.13), la cual determina la probabilidad con que una hormiga en cliente  $i$  se dirige al cliente  $j$ . ACS-VEI y ACS-TIME utilizan el mismo proceso de selección de candidatos, sin embargo cada una mantiene su tabla de feromonas de manera independiente.

#### DEFINICIÓN DEL CÁLCULO DE LA INFORMACIÓN HEURÍSTICA

Los valores de la información heurística  $\eta$  son asignados según la fórmula (4.14). Si el cliente fuente  $v_i$  y el cliente destino  $v_j$  pertenecen a diferente grupo ( $h_l \neq h_m$ ) se aplica un factor de corrección dado por el tamaño del conjunto de clientes  $V$  y el tamaño del conjunto de grupos  $H$ , de lo contrario se aplica la prioridad establecida por la longitud de ventana de tiempo.

$$\eta_{ij} = \begin{cases} \eta_{ij} \cdot |H| / |V| & , \text{ si } h_l \neq h_m \text{ \& } v_i \in h_l \text{ \& } v_j \in h_m \\ \eta_{ij} \cdot \text{priority}_{ij} & , \text{ otro caso} \end{cases} \quad (4.14)$$

Donde  $h_i, h_j \in H$  (conjunto de grupos),  $v_i, v_j \in V$  (conjunto de clientes) y  $\text{priority}$  denota la información de ventana de tiempo.

# Capítulo 5

## PROPUESTA DE SOLUCIÓN

En este capítulo se describen las modificaciones al algoritmo MACS-VRPTW. Para ello, se analizaron las instancias (subcapítulo 5.1.3 inicio de ventana), y el algoritmo (subcapítulo 5.1.4 adaptativo).

*Tabla 5.1. Modificaciones realizadas al algoritmo MACS-VRPTW*

Elemento	Modificaciones
LRE: Lista Restringida Extendida	<ul style="list-style-type: none"><li>- Recalculo después de selección de cliente.</li><li>- Activación por intervalos</li><li>- Considerar Inicio de ventana para generar información de ventanas.</li></ul>
LRA: Lista Restringida Adaptativa	<ul style="list-style-type: none"><li>- Agrupación semi-jerárquica.</li><li>- Agrupación alterna.</li></ul>
BI: Búsqueda Inicial	<ul style="list-style-type: none"><li>- Búsqueda aleatoria.</li><li>- Método Lin-Kernighan.</li></ul>
BL: Búsqueda Local	<ul style="list-style-type: none"><li>- Método Lin-Kernighan.</li></ul>
APN: Aprendizaje por niveles	<ul style="list-style-type: none"><li>- Desactivado.</li><li>- Tabla única.</li></ul>

La tabla 5.1 presenta información general de lo que ha sido modificado en el algoritmo MACS-VRPTW.

### 5.1 LISTA RESTRINGIDA EXTENDIDA (LRE)

En este punto es necesario recordar que esta heurística provee información de las ventanas de tiempo contenida en las instancias. Previo a este trabajo se utilizaron las unidades estadísticas media y desviación estándar para caracterizar las instancias; lo cual generaba confusión entre curvas y escalonadas, además de únicamente calcularse esta información al principio del algoritmo. A continuación se describe lo propuesto para esta heurística.

### 5.1.1 CARACTERIZACIÓN POR INTERVALOS

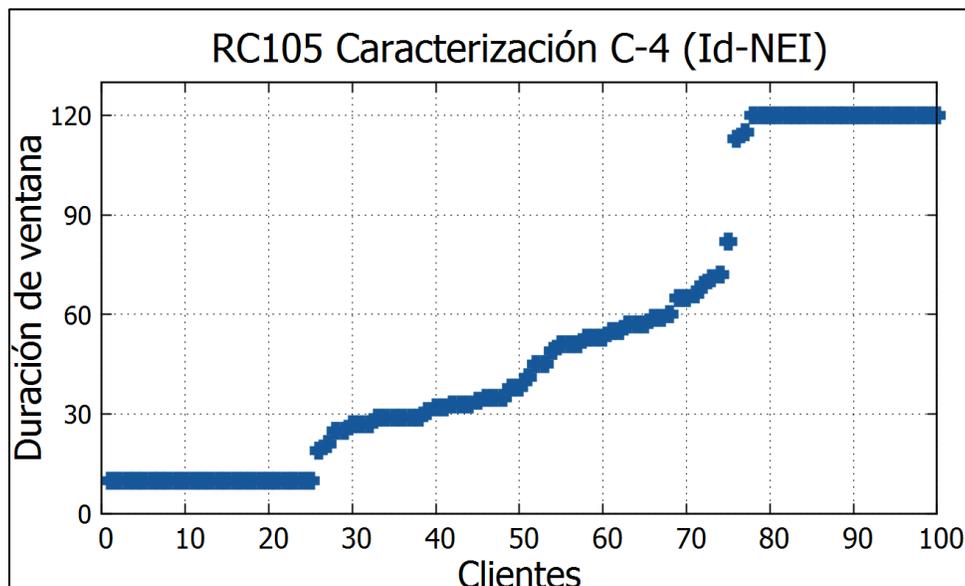
En la versión de [Cámara, 2011], el tipo al que una instancia pertenecía era determinado en base a la observación de su gráfica, siendo la activación de la heurística el único aspecto automático. Las instancias ahora son caracterizadas en base al valor de la variable NEI (número de intervalos no vacíos), ver tabla 5.2.

Tabla 5.2. Caracterización por intervalos según ventana de tiempo.

Tipo	Id	Descripción
Lineal	L	Todas las ventanas se encuentran en un sólo rango, NEI = 1.
Escalonada	E	Las ventanas pueden encontrarse en tan sólo 2 rangos diferentes, NEI = 2.
Curva	C	Las ventanas se encuentran en 3 o más rangos diferentes, NEI > 2.

Si la figura 5.1 mostrara una instancia en donde NEI = 2, dicha instancia se consideraría escalonada, y por lo tanto se procedería con el llenado de la tabla prioridad de la heurística LRE.

Figura 5.1. Gráfica de ventanas de tiempo de instancia RC105. Existe al menos una ventana de tiempo en cada uno de los rangos, por lo que NEI = 4, y por ende no se ejecuta la heurística LRE.



A continuación se describe el algoritmo para identificar las instancias de acuerdo a esta caracterización.

### 5.1.2 ACTIVACIÓN POR INTERVALOS

Los siguientes pasos sustituyen del segundo al quinto paso del algoritmo de LRE en subcapítulo 4.2.2.

- Definir cuatro intervalos de tiempo:  $Int_k = ((k-1) \cdot IntLength, k \cdot IntLength)$ ,  $k = 1, \dots, 4$ . Donde  $IntLength$  denota la longitud del intervalo. Ver fórmula 5.1.

$$IntLength = \frac{\max(tw_i)}{4} \quad (5.1)$$

- Para cada intervalo  $Int_k$ , determinar el número de clientes  $O_k$  cuya ventana se encuentre en dicho intervalo, con el objetivo de clasificar la instancia según su distribución de ventana de tiempo. Ver fórmulas (5.2) y (5.3).

$$O_k = \sum_{i=1}^{|V|} g(i, k) \quad (5.2)$$

donde

$$g(i, k) = \begin{cases} 1, & \text{si } tw_i \in Int_k \\ 0, & \text{otro caso} \end{cases} \quad (5.3)$$

- Si el número de intervalos no vacíos  $NEI$  es igual a 2, ver fórmula (5.4) y (5.5), entonces llenar la tabla de prioridad de MACS-VRPTW. Esta tabla de prioridad de tamaño  $[n, n]$ , donde  $n$  es el número de clientes, se construye de acuerdo a los valores de prioridad de visita entre los clientes.

$$NEI = \sum_{k=1}^4 g(k) \quad \text{y} \quad (5.4)$$

$$g(k) = \begin{cases} 1, & \text{si } O_k > 0 \\ 0, & \text{otro caso} \end{cases} \quad (5.5)$$

- Determinar  $priority_{lm}$  para cada par  $(l, m)$ , donde  $l$  es el cliente actual y  $m$  y  $k$  son clientes aún no asignados a la solución (camino). Ver fórmula (5.6)

$$priority_{lm} = \begin{cases} 3, & \text{si } NEI = 2, tw_l \& tw_m \in Int_{\min\{i|O_i=1, i=1,2,3\}}, wait_{lm}^1 < \lambda_1 \\ 2, & \text{si } NEI = 2, wait_{lm}^1 \geq \lambda_1, wait_{lm}^2 < \lambda_1, de_{lk} < \lambda_2 \\ 0, & \text{otro caso} \end{cases} \quad (5.6)$$

donde  $wait_{lm}^2 = a_m - (a_l + s_l + de_{lk} + s_k)$ ,  $wait_{lm}^1 = (a_m - a_l + s_l) - de_{lm}$ ,  $s_l$  es el tiempo de servicio del cliente  $l$ ,  $de_{lm}$  es la distancia euclidiana entre  $l$  y  $m$  mientras que  $\lambda_1$  y  $\lambda_2$  son umbrales de espera.

Los niveles de prioridad dos y tres son únicamente para instancias escalonadas. Nivel tres es para un par de clientes  $(l, m)$  ambos con ventanas de tiempo corta (pertenecientes al primer intervalo) y con un tiempo de espera aceptable  $wait_{lm}^1$  delimitado por  $\lambda_1$  entre los clientes  $l$  y  $m$ . Por otro lado, los pares de clientes con nivel dos tienen un tiempo de espera aceptable  $wait_{lm}^2$  también delimitado por  $\lambda_1$ , además de una distancia euclidiana aceptable  $de_{lk}$  entre cliente  $l$  y cliente intermedio  $k$  delimitada por  $\lambda_2$ .

### 5.1.3 INICIO DE VENTANA

Hasta este punto, únicamente se consideró el tamaño de la ventana de tiempo para realizar el llenado de la tabla prioridad de la heurística LRE. Entonces, se propuso incluir el inicio de ventana. Esto tiene como objetivo que aquellos clientes con ventana corta y cuya apertura se presente de forma temprana tengan la prioridad más alta.

Tomando en cuenta que aquí se utilizó la activación por intervalos, para poder añadir el inicio de ventana fue necesario realizar las siguientes modificaciones.

- Al comienzo del algoritmo, definir cuatro intervalos de inicio de tiempo:  $InitInt_k = ((k-1) \cdot InitIntLength, k \cdot InitIntLength)$ ,  $k = 1, \dots, 4$ . Donde  $InitIntLength$  denota la longitud del intervalo de inicio de tiempo. Ver fórmula 5.7.

$$InitIntLength = \frac{\max(a_i)}{4} \quad (5.7)$$

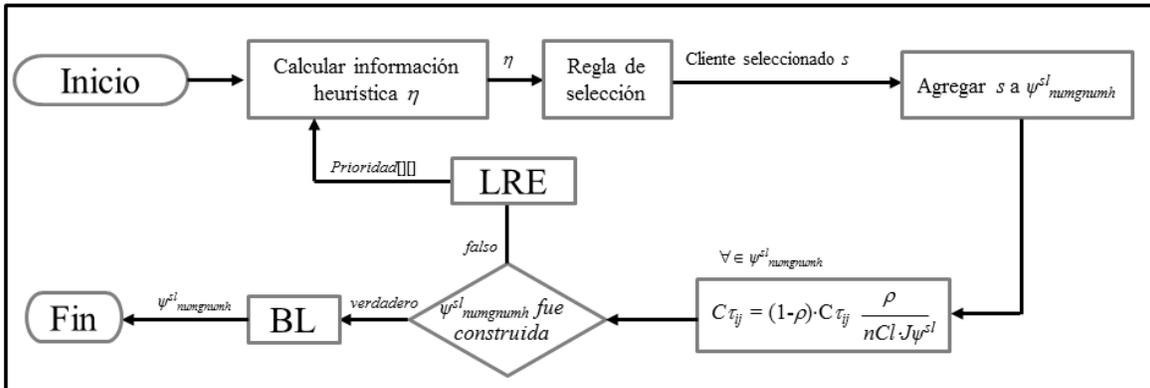
- Determinar  $priority_{lm}$  para cada par  $(l, m)$ , donde  $l$  es el cliente actual y  $m$  y  $k$  son clientes aún no asignados a la solución (camino). Ver fórmula (5.8)

$$priority_{lm} = \begin{cases} 7 & , NEI = 2, tw_l \& tw_m \in Int_{\min\{i|O_i=1, i=1,2,3\}}, wait_{lm}^1 < \lambda_1, \\ & a_l \in InitInt_1, a_k \in InitInt_2 \\ 6 & , NEI = 2, tw_l \& tw_m \in Int_{\min\{i|O_i=1, i=1,2,3\}}, wait_{lm}^1 < \lambda_1, \\ & a_l \& a_k \in InitInt_2 \\ 5 & , NEI = 2, tw_l \& tw_m \in Int_{\min\{i|O_i=1, i=1,2,3\}}, wait_{lm}^1 < \lambda_1, \\ & a_l \parallel a_k \in InitInt_{\{i|i=3,4\}} \\ 4 & , NEI = 2, wait_{lm}^1 \geq \lambda_1, de_{jk} < \lambda_2, a_l \in InitInt_1, a_k \in InitInt_2 \\ 3 & , NEI = 2, wait_{lm}^1 \geq \lambda_1, de_{jk} < \lambda_2, a_l \& a_k \in InitInt_2 \\ 2 & , NEI = 2, wait_{lm}^1 \geq \lambda_1, de_{jk} < \lambda_2, a_l \parallel a_k \in InitInt_{\{i|i=3,4\}} \\ 0 & , \text{otro caso} \end{cases} \quad (5.8)$$

#### 5.1.4 LRE ADAPTATIVO

LRE adaptativo representa el recalculo de la información de ventanas de tiempo, esto quiere decir que a medida que el tiempo progresa, el tamaño de la ventana para cada cliente puede verse modificada. La figura 5.2 muestra el lugar del algoritmo donde se vuelve a calcular la información de ventanas de tiempo es cuando la hormiga construye la solución parcial.

Figura 5.2 Modulo construcción de  $\psi^{sl}_{numgnumh}$  de esquema adaptativo.



## 5.2 LISTA RESTRINGIDA ADAPTATIVA (LRA)

Esta heurística se emplea para extraer información topológica de las instancias, donde a cada cliente se le da una zona geográfica de pertenencia llamada grupo, y así propiciar que sea más probable visitar a otro cliente entre más cercano se encuentre. Previamente se extraía la información mediante el algoritmo de agrupación jerárquica, el cual requería la construcción del árbol de expansión mínimo (MST) de los clientes, sin embargo fue hecho de forma incompleta. Por esta razón fueron propuestos dos algoritmos de agrupación, descritos en este subcapítulo.

### 5.2.1 AGRUPACIÓN SEMI JERÁRQUICA

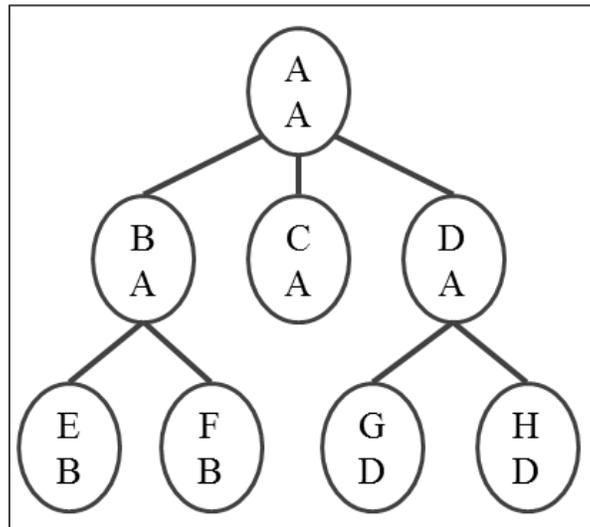
Se utilizó la implementación de la generación del árbol de expansión mínimo (MST) para adaptarla como agrupación semi-jerárquica, el pseudocódigo se puede observar en la figura 5.3

```
1  G = (V,E)
2  Ordenar_Aristas_MenMay();
3  Altura [n]= {0};
4  Grupo [n]= {desde i=1 hasta n};
5  cont_aristas = 0;
6  k=0;
7  While(cont_aristas<= num_clientes -1)
8      Pequeña = ordenado[k];
9      v1 = Pequeña.0;
10     v2 = Pequeña.1;
11     If(v1.padre != v2.padre)
12         If (altura [v1] == altura [v2])
13             Incrementar altura [v1];
14             Grupo [v2] = v1;
15         Fin de If
16     Else
17         If (altura [v1] > altura [v2])
18             Grupo [v2] = v1;
19         Fin de If
20     Else
21         Grupo [v1] = v2;
22     Fin de Else
23     Fin de Else
24     Fin de If
25     Incrementar k;
26     Fin de While.
```

*Figura 5.3, Algoritmo de agrupación semi-jerárquica. La información que indica el grupo al que pertenece cada nodo es almacenada en el vector llamado "Grupo"*

El diagrama de figura 5.4 ejemplifica el árbol que resulta de la agrupación semi - jerárquica. En cada nodo hay dos letras: la etiqueta de cliente y el grupo al que pertenecen. Únicamente el nodo raíz pertenece a sí mismo. El resto de los clientes pertenecerán al mismo grupo siempre y cuando sean hermanos en el árbol.

Figura 5.4, Árbol resultante del algoritmo de agrupación semi-jerárquica.



### 5.2.2 AGRUPACIÓN ALTERNA

Como otra opción a los algoritmos de agrupación anteriormente mencionados, se diseñó la agrupación alterna.

Figura 5.5, Algoritmo de agrupación alterna. La información que indica a qué grupo pertenece cada grupo es almacenada en el vector llamado "Grupo" y el número de grupos se almacena en "grupo\_actual".

1	Definir $Deuc[][]$ // distancias euclidianas entre clientes
2	Definir $cmax$ // capacidad máxima de distancia para un grupo.
3	Definir $dmax$ // distancia máxima para que dos clientes pertenezcan al mismo grupo.
4	Definir $a$ // cliente inicial para iniciar agrupación.
5	Inicializar $Grupo[] = 0$ // guarda a qué grupo pertenece cada cliente.
6	$Agrupado[]$ // guarda los clientes agrupados.
7	$No\_Agrupado[]$ // guarda los clientes no agrupados.
8	Inicializar $grupo\_actual = 1$ // grupo actual al que se agrupará siguiente cliente.
9	Inicializar $cliente\_previo = a$

```

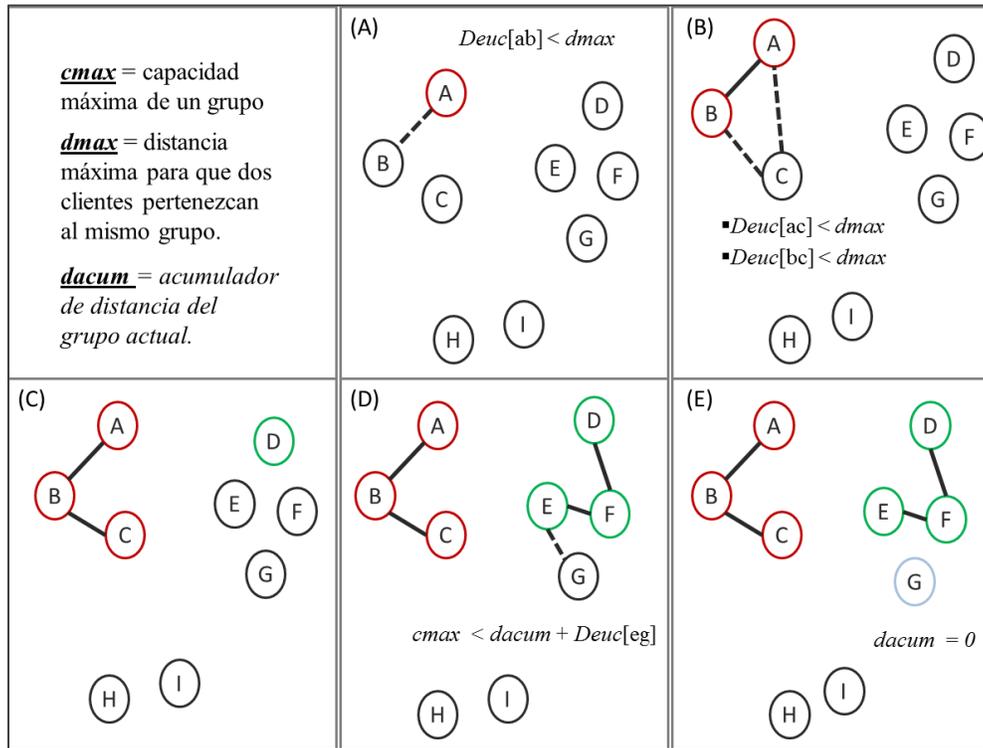
10  Inicializar dacum = 0 // acumulador de distancia para grupo_actual.
11  Grupo[a] = grupo_actual
12  Agrupado.add (a )
13  No_Agrupado.add ( todos excepto a )
14  While (Agrupado.HaySinAgrupar() )
15      Cliente_siguiete = -1
16      Inicializar fMinD = Float.Maxnum // guarda la menor distancia promedio
17      para clientes del grupo actual respecto de los no agrupados.
18      For i en No_Agrupado
19          Inicializar fDaux= 0 // acumulador de distancia para clientes del grupo actual
20          respecto de los no agrupados.
21          Inicializar nSameAcum = 0 //acumulador de clientes agrupados al grupo_actual.
22          For j en Agrupado
23              If (Grupo[j] == grupo_actual)
24                  If (dmax>= Deuc[i,j])
25                      fDaux+ = Deuc[i,j];
26                      nSameAcum++;
27                  Fin de If
28              Fin de For
29          fDaux /= nSameAcum;
30          If (fDaux> 0) // si fDaux = 0 significa que cliente i no puede agruparse a grupo_actual
31              If (fDaux<fMinD)
32                  fMinD = fDaux;
33                  cliente_siguiete = i;
34              Fin de If
35          Fin de If
36      Fin de For
37      If (cliente_siguiete != -1)
38          dacum += Deuc[cliente_previo,cliente_siguiete];
39          If (dacum>cmax)
40              dacum = 0;
41              grupo_actual++;
42          Fin de If
43          Grupo[cliente_siguiete] = grupo_actual;
44      Fin de If
45      Else
46          cliente_siguiete = No_Agrupado[0];
47          Grupo[cliente_siguiete] = ++grupo_actual;

```

48	<i>dacum</i> = 0;
49	Fin de Else
50	<i>No_Agrupado.Remove(cliente_siguiete);</i>
51	<i>Agrupado.Add(cliente_siguiete);</i>
52	<i>Cliente_previo = cliente_siguiete;</i>
53	Fin de While

En la figura 5.6, en el cuadro (A) se comienza por elegir un nodo inicial, el cual pertenecerá al primer grupo. Desde tal nodo seleccionado como inicial, ahora perteneciente al grupo actual, se obtiene el nodo más cercano a él y se le incluye al grupo actual, siempre y cuando *dmax* no sea mayor a la distancia euclidiana entre los dos nodos involucrados. Para el cuadro (B), de nuevo se obtiene el nodo sin agrupar más cercano respecto a los nodos del grupo actual - el inicial y el elegido previamente - siempre y cuando *dmax* no sea mayor a la distancia euclidiana entre los nodos del grupo actual y el nodo más cercano a ellos. En cuadro (C), si respecto de todos nodos del grupo actual, no hay ningún nodo que cumpla con *dmax*, entonces se elige otro nodo --- el primero de los no agrupados--, asignándose al siguiente grupo, reinicializando *dacum*, la cual hasta el momento tenía el valor de la distancia euclidiana de nodo *A* al nodo *B* más la distancia euclidiana del nodo *B* al nodo *C*. Para cuadro (D), otra restricción para asignar un nodo al grupo actual es que la capacidad acumulada del grupo actual más la distancia del nodo más cercano no sea mayor a *cmax*. Para este caso se supone que al agregar el nodo *G*, la capacidad acumulada ha superado la capacidad máxima *cmax*. En el cuadro (E), se asigna el nodo *G* a un grupo distinto, y se reinicializa la capacidad acumulada *dmax* a cero.

Figura 5.6. Proceso de agrupación alterna.



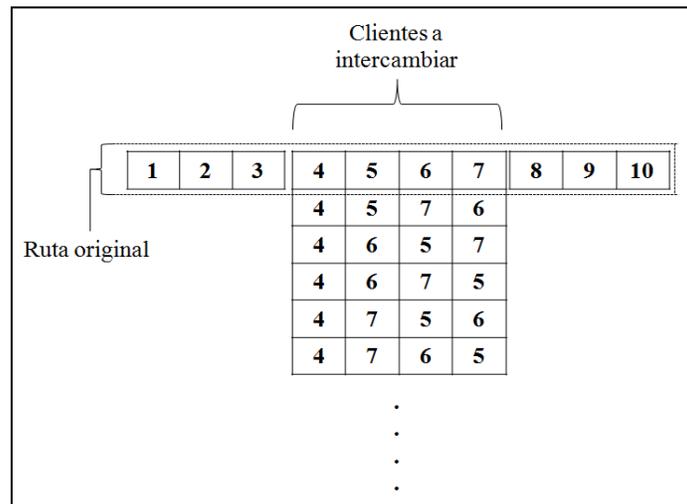
### 5.3 BÚSQUEDA INICIAL (BI) Y LOCAL (BL)

Cámara en [Cámara,2011] realizó una búsqueda inicial mediante el método de vecinos cercanos y una búsqueda local por esquema 3-opt y Cross-Exchange. En este trabajo se realizó lo siguiente:

- **Búsqueda Aleatoria:** como su nombre lo dice, dado un cliente actual  $q$ , elige al azar el cliente siguiente  $q1$ . La búsqueda sigue si  $q1$  fue visitado anteriormente.
- **Algoritmo Lin-Kernighan:** El algoritmo se especifica en términos de intercambios que pueden convertir una ruta en otra. Dada una ruta factible, el algoritmo realiza intercambios repetidamente con objeto de reducir la longitud de la ruta, hasta que se alcance una ruta para la que cualquier intercambio desde ella no repercuta en mejora [Helsgaun, 2000].

En la figura 5.7 sólo es mostrado cuando el vértice etiquetado con 1 es el primer vértice, restándolos otros 3 vértices por ser el primer vértice, lo cual hace un total de 24 permutaciones para  $k = 2$ . El algoritmo de Lin-Kernighan también fue aplicado como búsqueda local dentro de las colonias de hormigas de algoritmo MACS-VRPTW.

Figura 5.7 Ejemplo de permutaciones en movimiento para un entero  $k$  de valor 2.



#### 5.4 APRENDIZAJE POR NIVELES (APN)

El algoritmo utilizado por [Cámara, 2011] utiliza dos niveles de conocimiento, uno global y uno local, almacenando cada uno de ellos en estructuras diferentes,  $\tau$  y  $C\tau$  respectivamente. Mientras que los algoritmos propuestos son los siguientes:

- APN desactivado: Consiste en utilizar el primer nivel de conocimiento de aprendizaje por niveles mostrado en subcapítulo 4.2.4 y descartar el segundo nivel, es decir la actualización local.
- Tabla única: En este se almacenan ambos niveles de conocimiento en una única estructura.

# Capítulo 6

## EXPERIMENTACIÓN Y RESULTADOS

En este capítulo se presenta la secuencia de experimentos que llevó a los resultados obtenidos por el algoritmo MACS-VRPTW. Enseguida se listan dichos experimentos, y posteriormente se describe el proceso de cada uno de ellos.

- Ranking de estrategias.
- Revisión de condición de activación de LRE por valores estadísticos.
- Propuesta de condición de activación de LRE por intervalos no vacíos.
- Revisión de LRA y propuesta de algoritmos de agrupación.
- Algoritmo genético para ajuste de estrategias.

### 6.1 RANKING DE ESTRATEGIAS

Al revisar la estructura de la versión del algoritmo MACS-VRPTW hecha por [Cámara, 2011], se identificaron las siguientes estrategias:

- Lista Restringida Extendida (LRE)
- Búsqueda Inicial (BI)
- Lista Restringida Adaptativa (LRA)
- Búsqueda Local en Colonia de Hormigas enfocada a minimizar el número de Vehículos (BL-AV)
- Búsqueda Local en Colonia de Hormigas enfocada a minimizar el Tiempo de recorrido (BL-AT)
- Aprendizaje por Niveles en Colonia de Hormigas enfocada a minimizar el número de Vehículos (APN-AV)
- Aprendizaje por Niveles en Colonia de Hormigas enfocada a minimizar el Tiempo de recorrido (APN-AT)

El fin de este experimento fue determinar el beneficio independiente de cada una de las estrategias sobre el algoritmo MACS-VRPTW. Para llevarlo a cabo fue identificado el lugar de la implementación donde se encontraba cada estrategia y se modificaron para ser activadas y desactivadas manualmente.

En la tabla 6.1 se muestra un ranking, el cual se obtuvo de ejecutar el algoritmo MACS-VRPTW con una estrategia desactivada a la vez. Después se creó una medida en la que una estrategia es más importante si se encuentra desactivada y su solución disminuye de calidad. Las estrategias se presentan de izquierda a derecha en orden de importancia. Debido a que la estrategia LRE no obtuvo la importancia esperada respecto del resto, se procedió a su revisión.

*Tabla 6.1 Ranking de estrategias.*

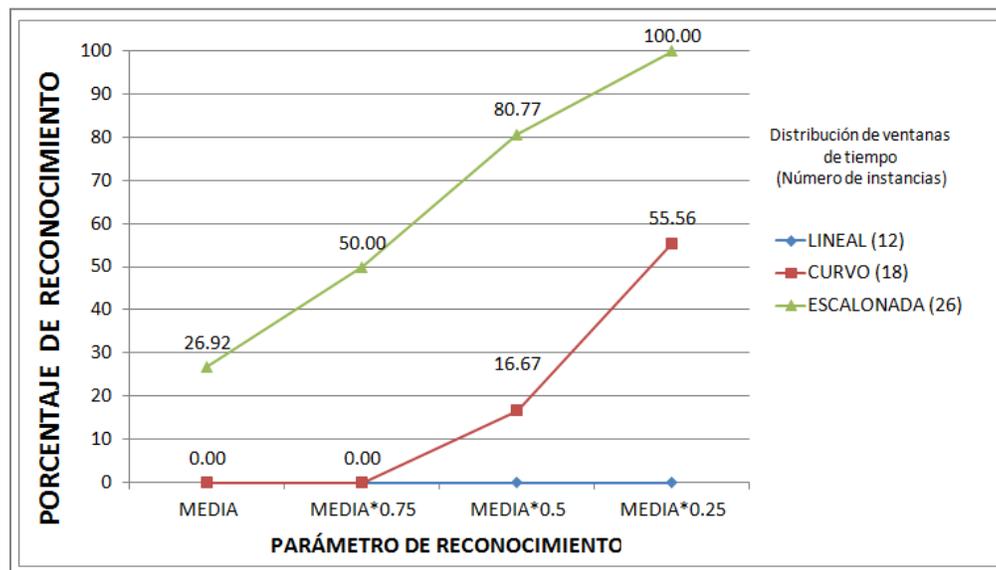
TIPO DE INSTANCIA	RANKING DE ESTRATEGIAS						
	1	2	3	4	5	6	7
C1	APN-AT	BL-AT	LRA	BL-AV	LRE	APN-AV	BI
	0.797138	0.811448	0.822391	0.906566	0.908039	0.916035	0.916877104
C2	LRA	APN-AT	BL-AT	LRE	APN-AV	BI	BL-AV
	0.77178	0.80303	0.822917	0.908144	0.913352	0.913826	0.922111742
R1	APN-AT	BL-AT	LRA	BI	APN-AV	LRE	BL-AV
	0.772727	0.801136	0.842014	0.913037	0.913352	0.915088	0.915246212
R2	APN-AT	BL-AT	LRA	LRE	APN-AV	BL-AV	BI
	0.793561	0.800103	0.837121	0.905131	0.905647	0.910296	0.919938017
RC1	LRA	APN-AT	BL-AT	BI	LRE	BL-AV	APN-AV
	0.790956	0.801136	0.820786	0.908854	0.912169	0.913826	0.937026515
RC2	LRA	APN-AT	BL-AT	BI	LRE	BL-AV	APN-AV
	0.777699	0.807292	0.817472	0.914063	0.915956	0.921875	0.926136364

## 6.2 REVISIÓN DE CONDICIÓN DE ACTIVACIÓN DE LRE POR VALORES ESTADÍSTICOS

En la versión del algoritmo MACS-VRPTW hecha por [Cámara, 2011], se considera a la instancia de entrada como escalonada si la desviación estándar  $\sigma$  de la distribución de ventana de tiempo sea mayor a su media  $\mu$ . Sin embargo, de 26 instancias que gráficamente fueron determinadas como escalonadas, únicamente 7 de ellas fueron reconocidas como tal por la heurística LRE. Con el fin de identificar correctamente a todas las instancias escalonadas fueron definidas las siguientes condiciones:  $\sigma > \mu$ ,  $\sigma > 0.75\mu$ ,  $\sigma > 0.5\mu$ ,  $\sigma > 0.25\mu$ .

En la figura 6.1 puede observarse que ninguna de las condiciones definidas es satisfactoria. Esto se puede concluir debido a que al utilizar  $\sigma > \mu$  y  $\sigma > 0.75\mu$  no pueden ser reconocidas la totalidad de las instancias escalonadas como tal, además de que empleando  $\sigma > 0.5\mu$  y  $\sigma > 0.25\mu$  un porcentaje de las instancias curvas son reconocidas como escalonadas.

Figura 6.1 Efectividad de LRE en el reconocimiento de las instancias de [Solomon, 1987] con valores estadísticos como condición de activación.



### 6.3 PROPUESTA DE CONDICIÓN DE ACTIVACIÓN DE LRE POR INTERVALOS NO VACÍOS.

El experimento anterior evidenció que era necesario proponer una condición de activación para LRE que no dependiera de valores estadísticos. Se propuso NEI lo cual significa número de intervalos no vacíos donde LRE se activa únicamente si se ocupan dos de los cuatro intervalos en los que se dividen las ventanas de tiempo, ver subcapítulo 5.1.1 y 5.1.2. A continuación se presentan las gráficas de cada una de las instancias obtenidas mediante caracterización por intervalos.

Figura 6.2 Instancias identificadas como lineales bajo la caracterización por intervalos (parte 1).

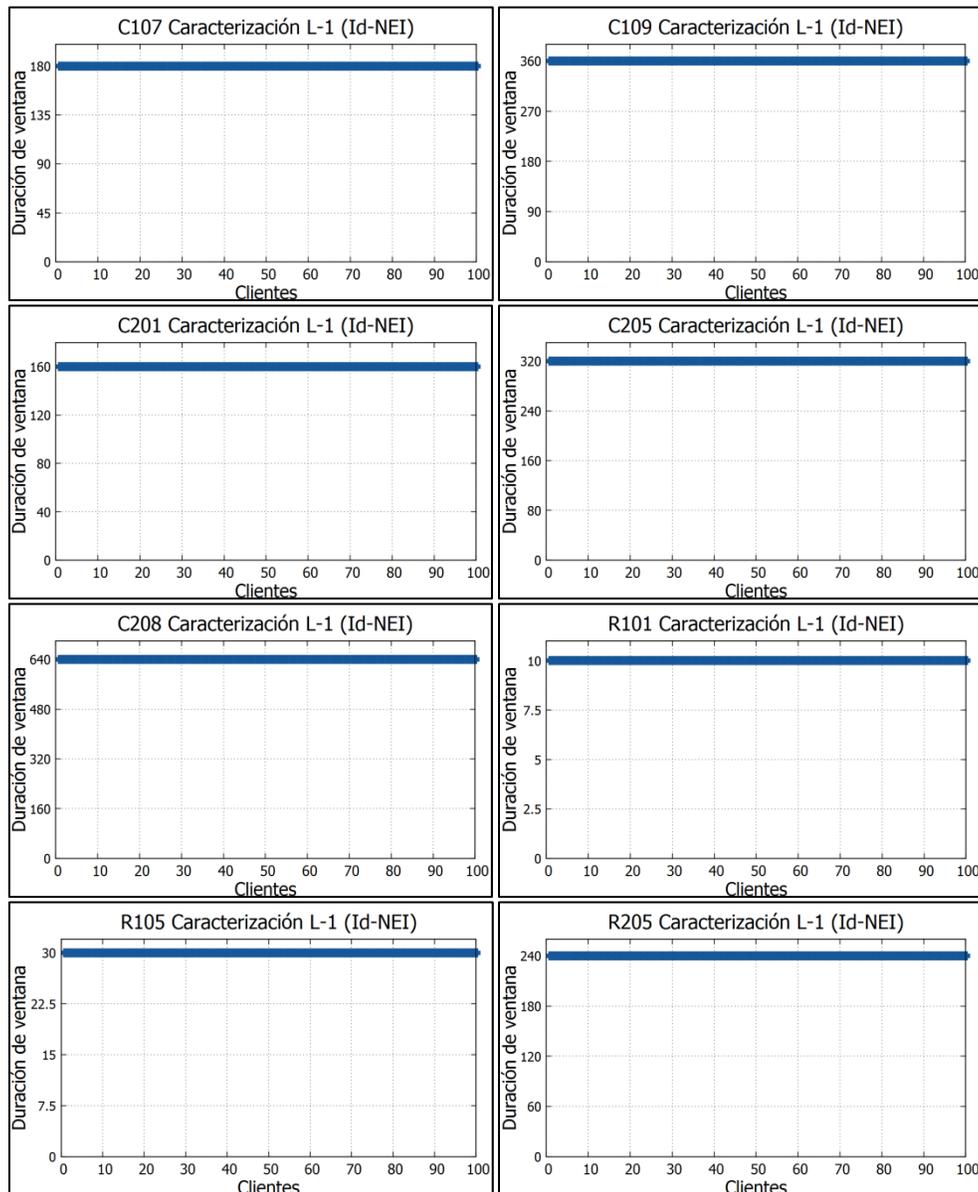


Figura 6.3 Instancias identificadas como lineales bajo la caracterización por intervalos (parte 2).

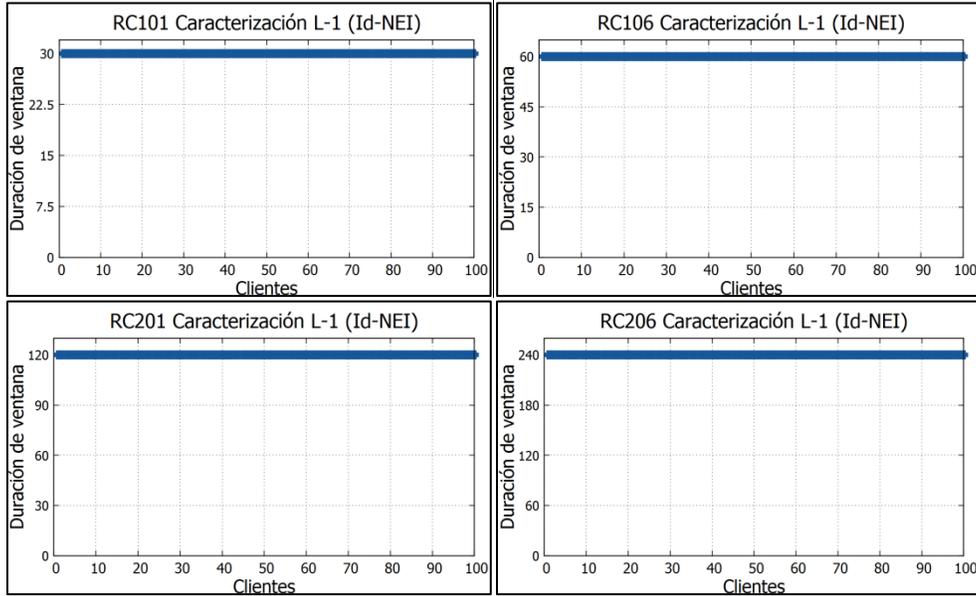


Figura 6.4 Instancias identificadas como curvas bajo la caracterización por intervalos (parte 1).

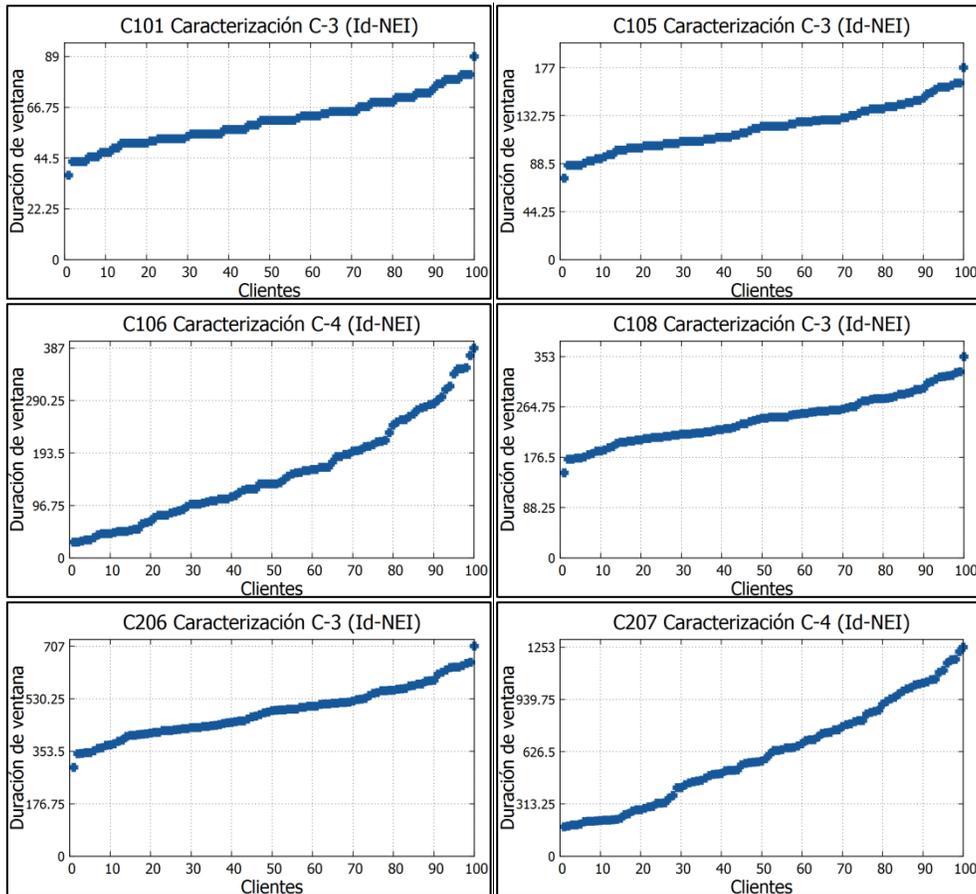


Figura 6.5 Instancias identificadas como curvas bajo la caracterización por intervalos (parte 2).

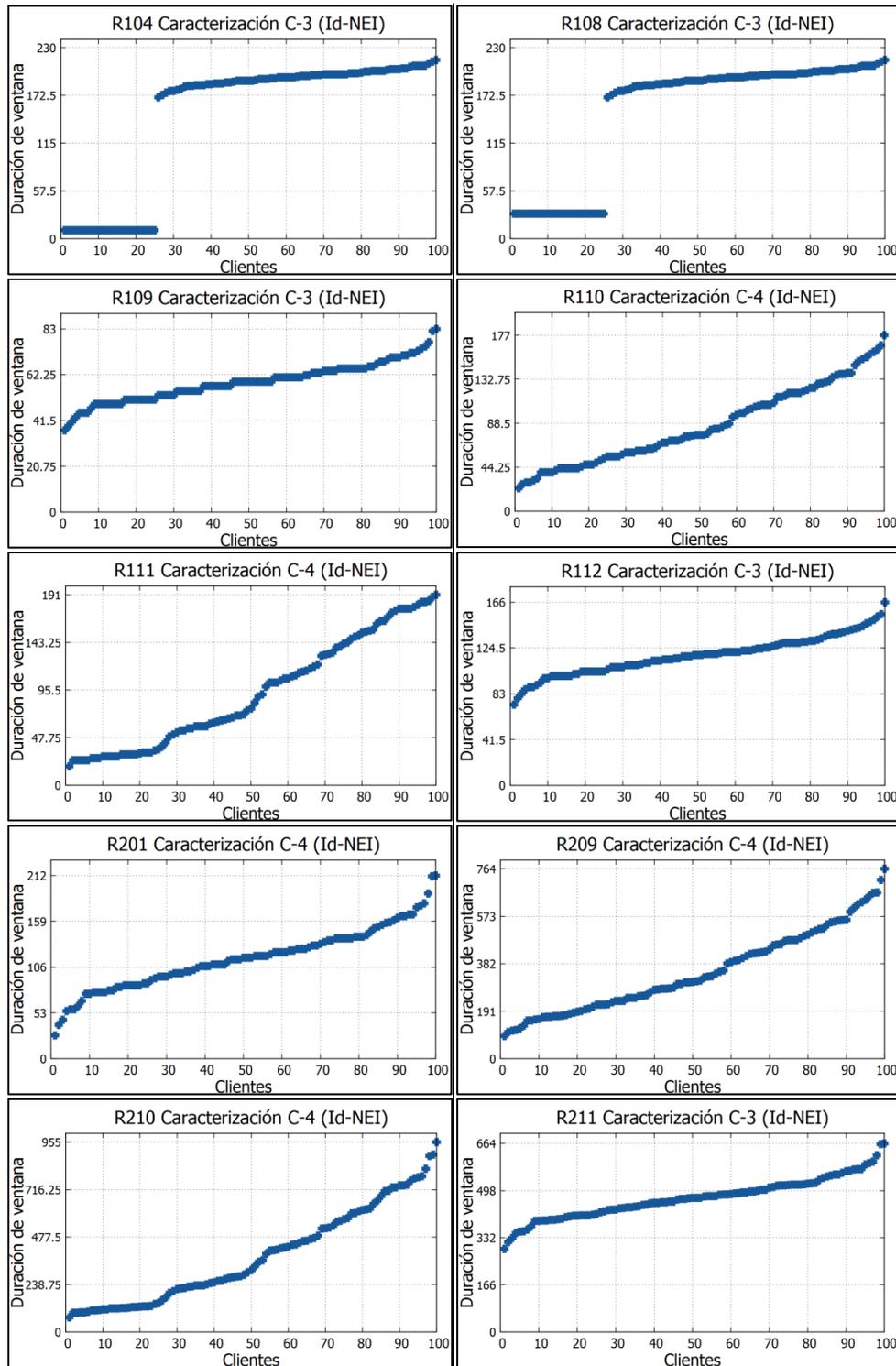


Figura 6.6 Instancias identificadas como curvas bajo la caracterización por intervalos (parte 3).

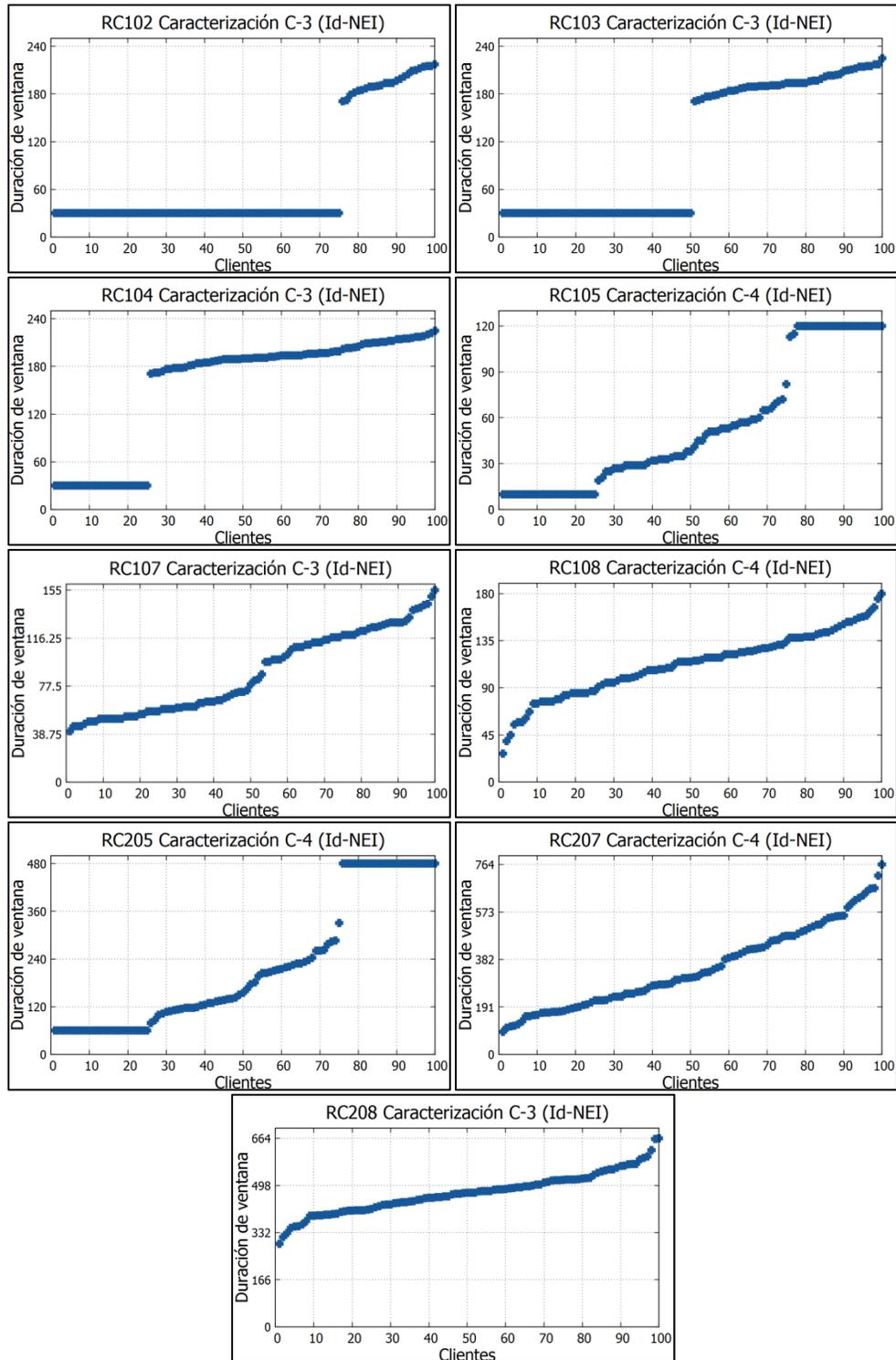


Figura 6.7 Instancias identificadas como escalonadas bajo la caracterización por intervalos (parte 1).

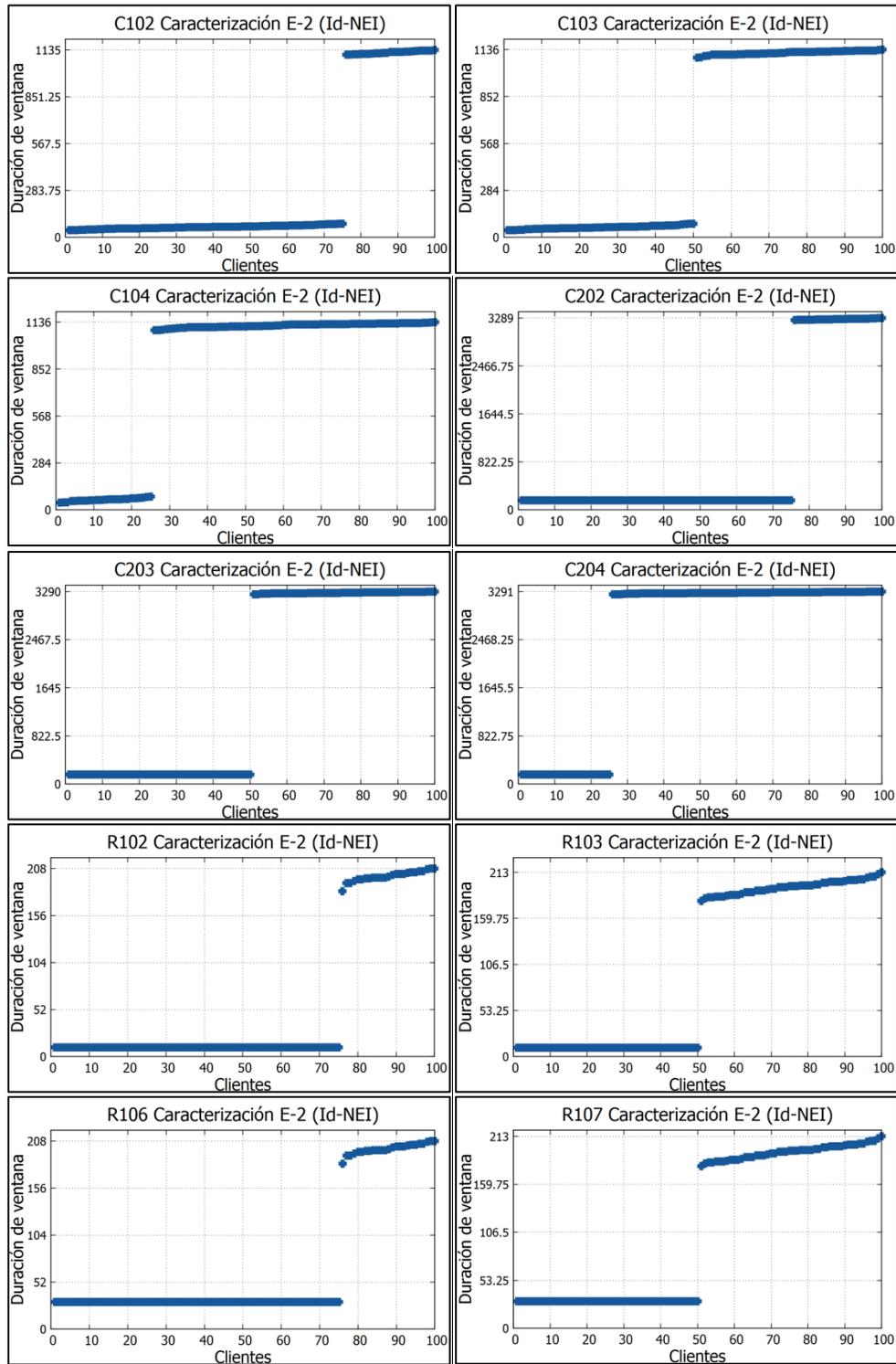
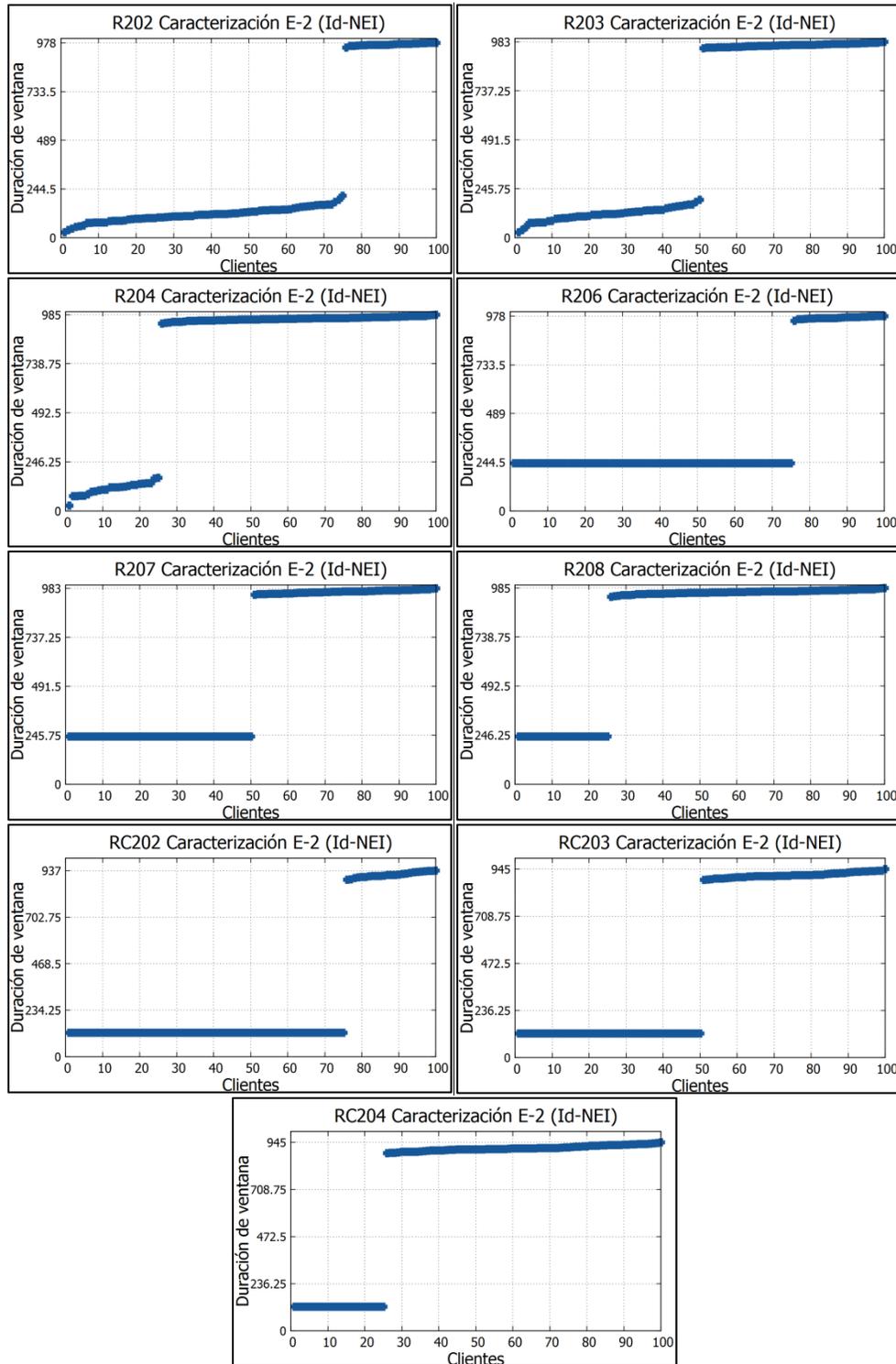


Figura 6.8 Instancias identificadas como escalonadas bajo la caracterización por intervalos (parte 2).

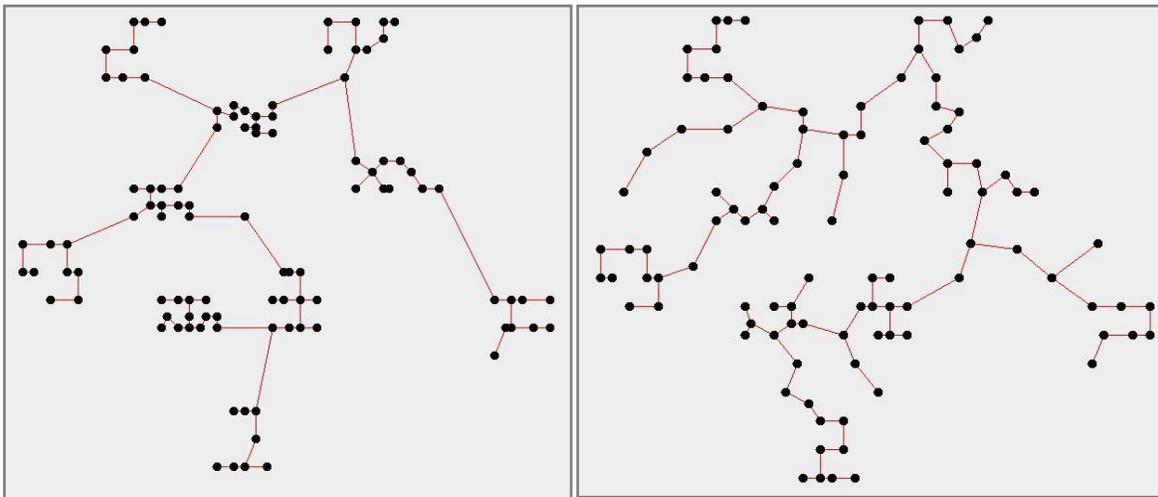


Al considerar NEI como referencia para caracterizar las instancias de [Solomon, 1987] de acuerdo a su distribución de ventana de tiempo 12 de ellas fueron etiquetadas como instancias lineales, 25 como curvas y 19 como escalonadas.

#### 6.4 REVISIÓN DE LRA Y PROPUESTA DE ALGORITMOS DE AGRUPACIÓN

Fue analizado el algoritmo de agrupación jerárquica utilizado por LRA para generar la lista de candidatos, y se encontró que el árbol de expansión mínimo MST era construido carente de una arista en la implementación. Al corregirse el error se produjeron los arboles mostrados en las figura 6.9 a 6.11.

*Figura 6.9* Árbol de expansión mínimo (MST) para instancias C1 y C2.



*Figura 6.10* Árbol de expansión mínimo (MST) para instancias R1 y R2.

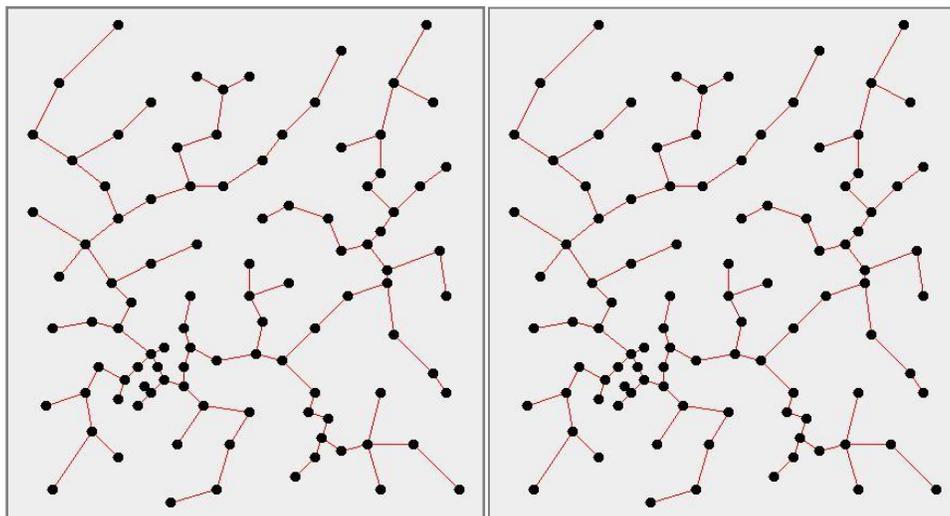
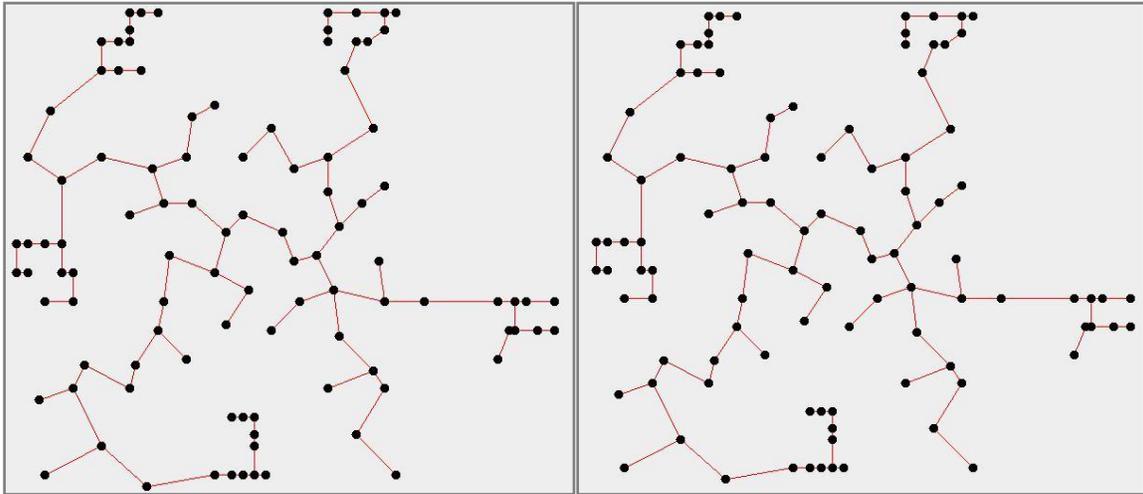


Figura 6.11 Árbol de expansión mínimo (MST) para instancias RC1 y RC2.



Después de corregirse la creación del árbol de expansión mínimo (MST) se volvió a ejecutar el algoritmo de agrupación jerárquica y los clientes fueron agrupados según las imágenes mostradas en las figuras 6.12 a 6.14, donde dos clientes del mismo color pertenecen al mismo grupo.

Figura 6.12 Resultados de agrupación jerárquica para instancias C1 e instancias C2 a izquierda y derecha respectivamente.

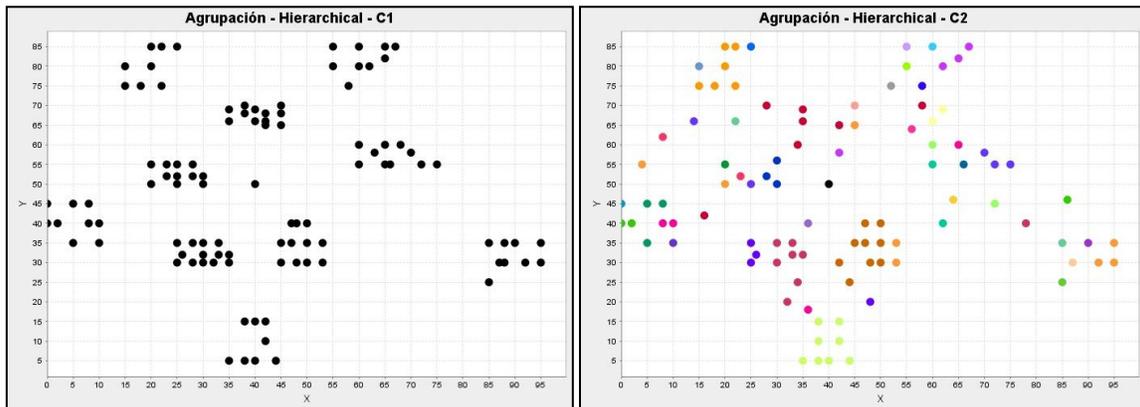


Figura 6.13 Resultados de agrupación jerárquica para instancias R1 e instancias R2 a izquierda y derecha respectivamente.

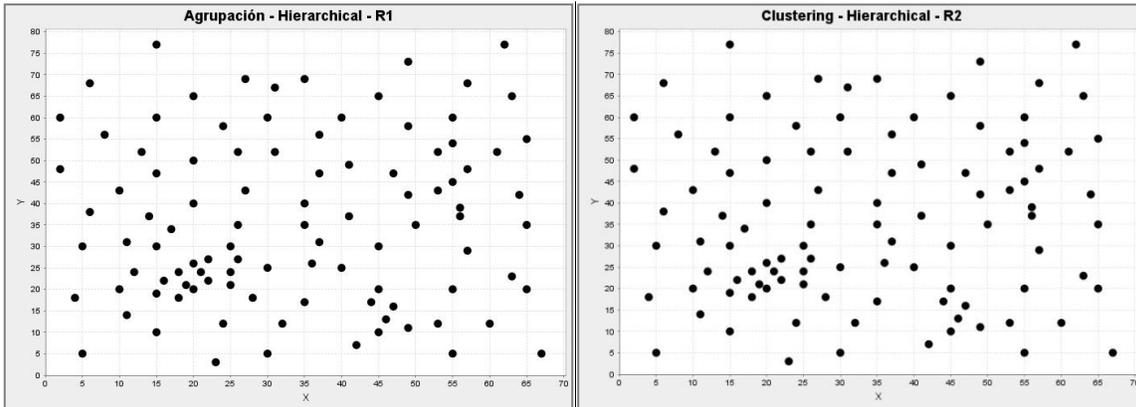
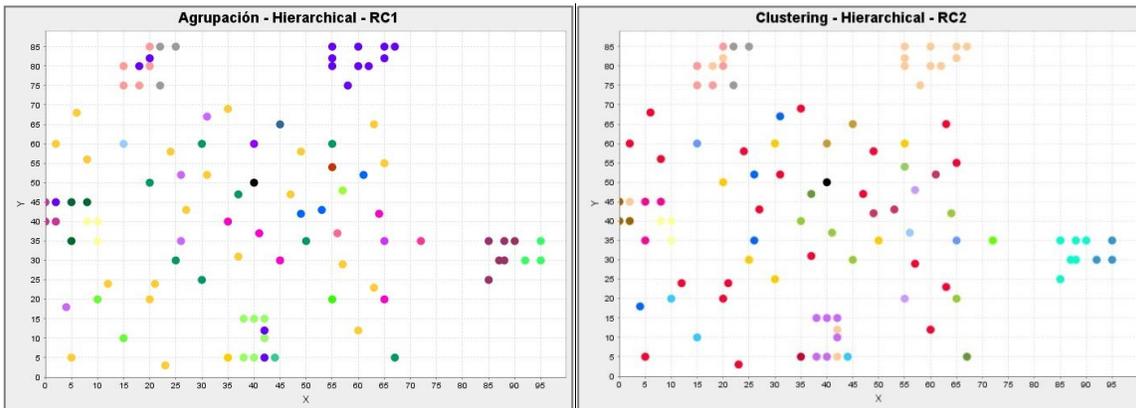


Figura 6.14 Resultados de agrupación jerárquica para instancias RC1 e instancias RC2 a izquierda y derecha respectivamente.



En tabla 6.2 puede observarse que hay tres tipos de instancias para los cuales todos los clientes fueron reunidos en un solo grupo, lo que indica que la agrupación no se realizó correctamente. Debido a esto, se propusieron los algoritmos de agrupación semi-jerárquica y alterna, el primero de ellos utiliza la implementación del árbol de expansión mínimo mientras que el segundo utiliza criterios de proximidad y umbral de clientes en un solo grupo, ver subcapítulo 5.2.1 y 5.2.2.

Tabla 6.2 Cantidad de grupos generados por agrupación jerárquica

INSTANCIA	C1	C2	R1	R2	RC1	RC2
No. Grupos	1	47	1	1	28	28

Enseguida de la figura 6.15 a la figura 6.20 se muestran las imágenes donde se han caracterizado los clientes mediante los dos algoritmos de agrupación propuestos.

Figura 6.15 Agrupación semi-jerárquica para instancias C1y C2, en ambos se obtuvieron 6 grupos.

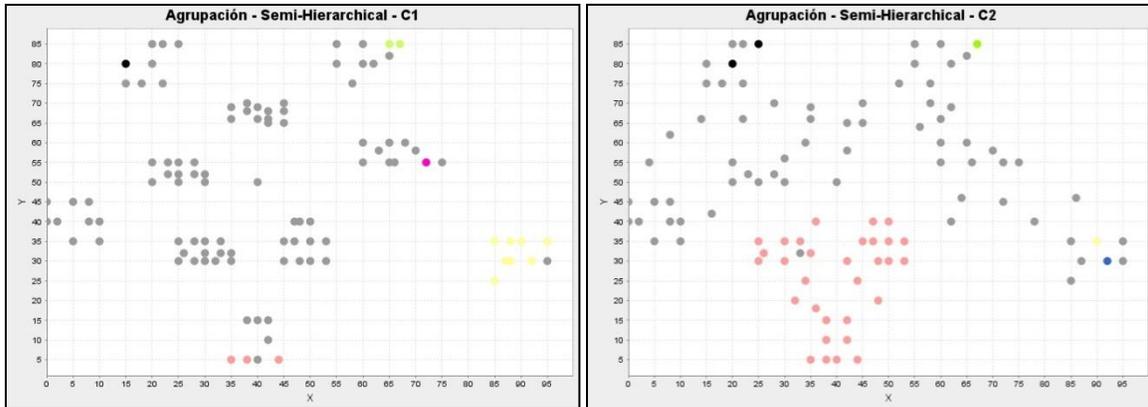


Figura 6.16 Agrupación semi-jerárquica para instancias R1y R2, en ambos se obtuvo 1 grupo.

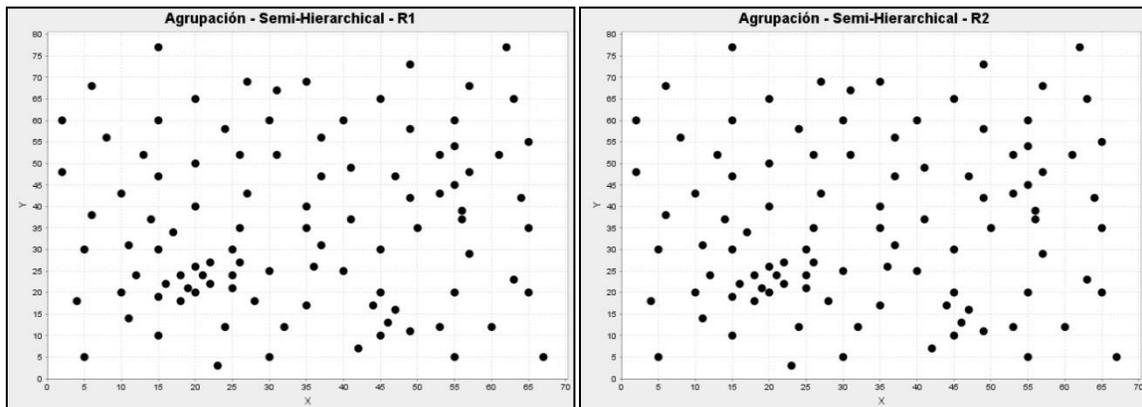


Figura 6.17 Agrupación semi-jerárquica para instancias RC1 y RC2, en ambos se obtuvieron 7 grupos.

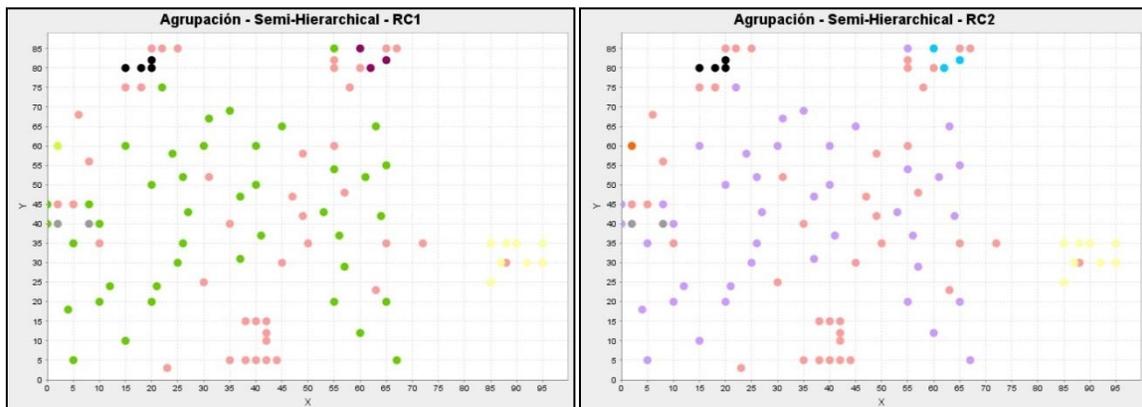


Figura 6.18 Agrupación alterna para instancias C1 y C2, se obtuvieron 10 y 15 grupos respectivamente.

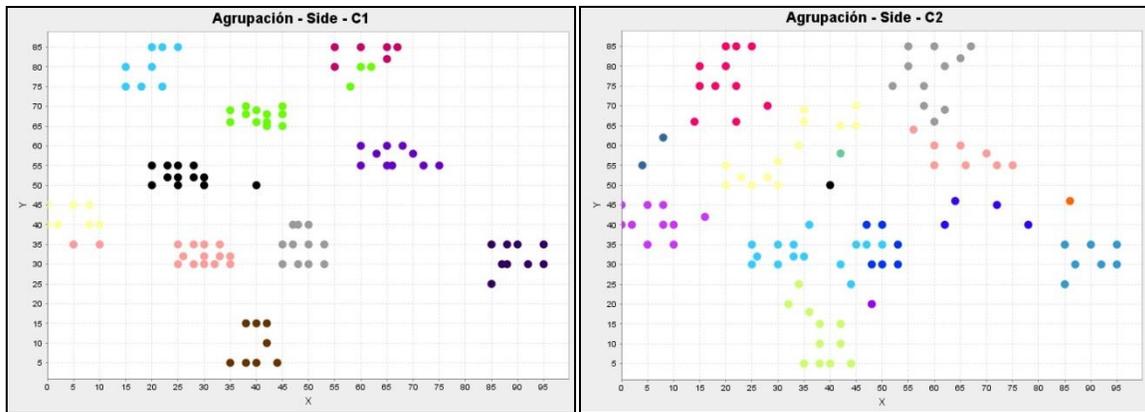


Figura 6.19 Agrupación alterna para instancias R1 y R2, en ambos se obtuvieron 20 grupos.

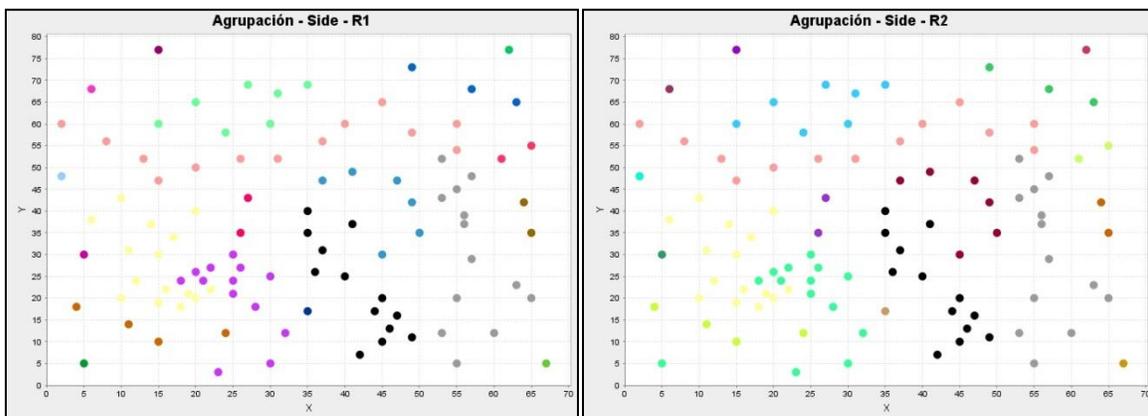
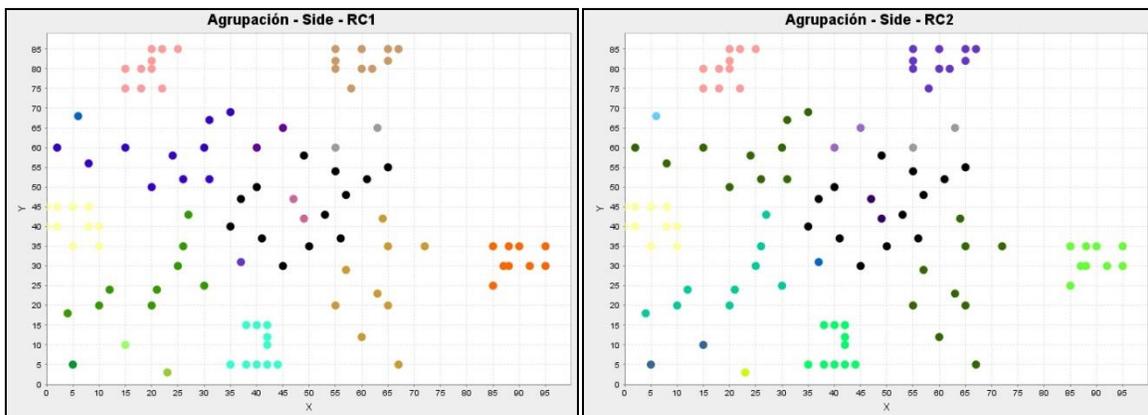


Figura 6.20 Agrupación alterna para instancias RC1 y RC2, en ambos se obtuvieron 17 grupos.



La agrupación semi-jerárquica conforma los clientes en 6 grupos para C1y C2, siendo en ambos casos más conveniente que la agrupación jerárquica donde los clientes en instancias de tipo C1 son considerados como un solo grupo y los clientes en instancias de tipo C2 son distribuidos en 47 grupos diferentes. Para los clientes en instancias de tipo R1 y R2, se mantuvo la cantidad en un grupo, y en instancias de tipo RC1 y RC2 se redujo de 28 a 7 grupos.

La agrupación alterna produce las cantidades de grupos más convenientes ya que el mayor número de grupos generados en algún tipo de instancia por esta agrupación es de 20 en instancias de tipo R1 y R2, comparado con los 47 de la agrupación jerárquica en instancias de tipo C2. De igual forma genera un número no tan reducido como lo son 10 grupos, comparado con los 6 grupos de la agrupación semi-jerárquica, ambos en instancias de tipo C1, ver tabla 6.3.

*Tabla 6.3 Tabla de coeficientes de los algoritmos de agrupación .*

<b>Tipo de instancia</b>	<b>Cantidad de grupos</b>			<b>Porcentaje de clientes por grupo</b>		
	<b>Jerárquica</b>	<b>Semi-Jerárquica</b>	<b>Alterna</b>	<b>Jerárquica</b>	<b>Semi-Jerárquica</b>	<b>Alterna</b>
C1	1	6	10	100	16.67	10
C2	47	6	15	2.13	16.67	6.67
R1	1	1	20	100	100	5
R2	1	1	20	100	100	5
RC1	28	7	17	3.57	14.29	5.88
RC2	28	7	17	3.57	14.29	5.88

## 6.5 ALGORITMO GENÉTICO PARA AJUSTE DE ESTRATEGIAS

Se implementó un algoritmo genético aplicado a las estrategias, en donde el sistema de colonia de hormigas se utiliza para determinar el valor de la función objetivo de cada individuo de la población. Cada individuo representa una diferente configuración para que las heurísticas en el sistema de colonia de hormigas sean activadas o desactivadas en base a dicha configuración.

Para la figura 6.21, los primeros 7 genes del individuo sólo pueden tomar valores de 1 y 0, lo cual denota activación o desactivación respectivamente. El resto de los genes indicarán la versión que se aplicará para cada una de las estrategias, los valores que pueden tomar estos genes se aprecian en detalle en la tabla 6.4.

Figura 6.21 Individuo en población de genético de estrategias.

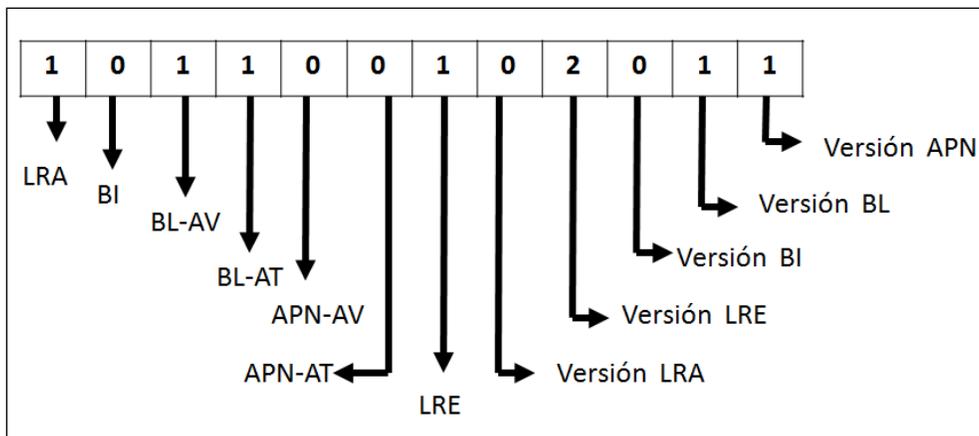
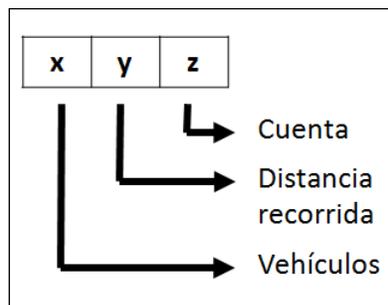


Tabla 6.4 Relación de valores que pueden adquirir del octavo al décimo segundo gen y el algoritmo que se ejecuta de acuerdo a tal valor.

Gen de versión	Valor de versión				
	-1	0	1	2	3
LRA	Desactivada	Semi-jerárquica	Jerárquica	Alternativa	N/A
LRE	Desactivada	Ventanas	Ventanas+Inicio	Ventanas Adaptativo	Ventanas+Inicio Adaptativo
BI	Desactivada	Vecinos cercanos	K-opt	N/A	N/A
BL	Desactivada	Cross-Exchange	K-opt	N/A	N/A
APN	Desactivada	2 tablas	Tabla única	N/A	N/A

Obsérvese la figura 6.22, en ella se muestra la función objetivo. Esta se compone de tres elementos. “Vehículos” y “Distancia recorrida” que se obtienen después de evaluar el sistema de colonia de hormigas con la configuración proporcionada por el individuo. El otro elemento llamado “Cuenta” es la suma de los primeros 7 genes del individuo. Dicho lo anterior, la jerarquía de la función individuo comienza por “Vehículos”, si dos individuos tienen la misma cantidad de estos, se comparan los valores de “Distancia recorrida”, si persiste la igualdad se compara “Cuenta”, Este último elemento de la función objetivo sirve para en caso de que dos individuos obtengan los mismos valores para los dos primeros elementos, se tome como mejor individuo aquel que haya requerido de la activación de menos estrategias.

*Figura 6.22 Elementos de función objetivo para un individuo en población de genético de estrategias.*



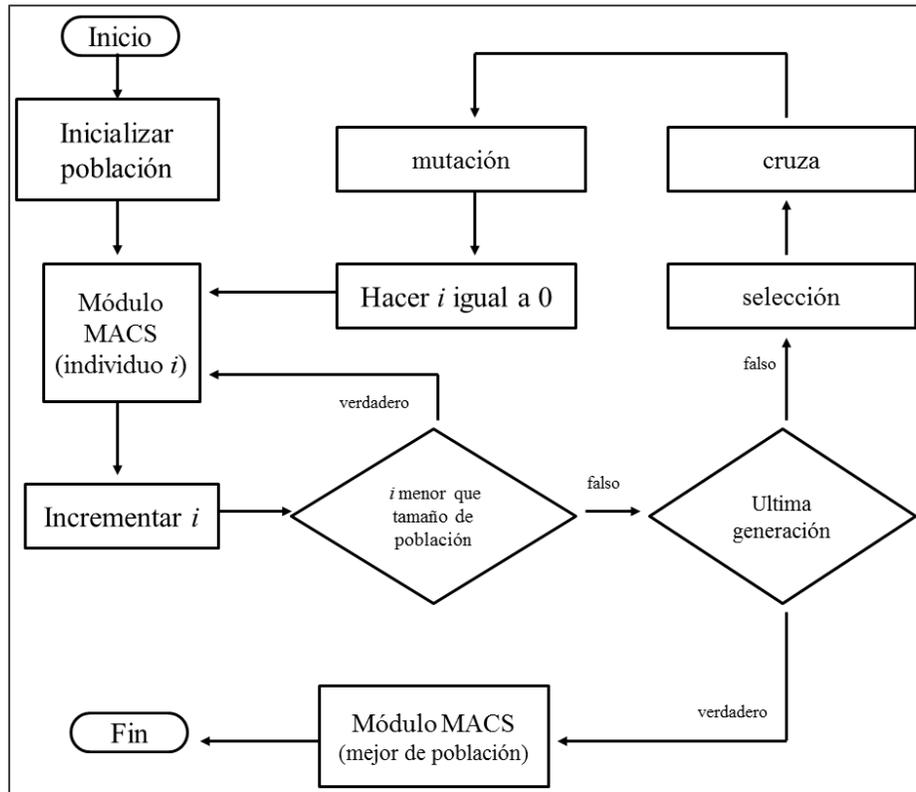
Un aspecto relevante de cualquier algoritmo, y especialmente de cualquier genético son los parámetros, lo cuales determinan la base de la cual parte el mismo proceso de búsqueda del algoritmo. Los parámetros para el genético de estrategias pueden observarse en la tabla 6.5.

Tabla 6.5 Parámetros para un algoritmo genético que selecciona estrategias.

CRITERIO	DESCRIPCIÓN	VALORES UTILIZADOS
Población (individuos)	Número de individuos en la población	50
Generaciones	Número de generaciones que se ejecuta el algoritmo genético.	10
% Activación	Representa la probabilidad que una estrategia sea activada (gen = 1) al momento de generar la población inicial	50
Iteraciones	Número de veces que un individuo llama al sistema de colonia de hormigas para obtener su función objetivo.	10
Selección (cruza)	Criterio para generar los descendientes.	20% mejor con 80% peor
Operadores de cruza utilizados	Operadores de cruza utilizados	➤ De un punto, el punto es elegido al azar.
		➤ Uniforme, con máscara 1010101, (primeros 7 genes)
Selección (mutación)	Número de individuos a ser mutados.	individuos
% Mutación	Numero aleatorio $x$ , debe ser menor a esta tasa para que un individuo seleccionado sea mutado	0.2
Operadores de mutación utilizados	Operadores de cruza utilizados	➤ Mutar gen elegido aleatoriamente.
		➤ Intercambio entre dos genes elegidos aleatoriamente

Hasta aquí han sido descritos los elementos del algoritmo genético implementado, por lo que a continuación en la figura 6.23, muestra el esquema de solución de genético con MACS-VRPTW.

Figura 6.23 Esquema de solución de algoritmo genético.



En la figura 6.23 puede observarse que para evaluar un individuo se llama al módulo MACS, el cual es el sistema de colonia de hormigas que ha sido utilizado durante la realización de este trabajo.

## 6.6 RESULTADOS

Debido a que los algoritmos metaheurísticos se ven afectados por la configuración paramétrica inicial [Díaz, 1996], se hace necesario revisar los valores de los parámetros utilizados para el algoritmo de MACS-VRPTW. La tabla 6.6 muestra los valores utilizados.

Tabla 6.6. Parámetros utilizados en algoritmo MACS-VRPTW por [Cámara, 2011]

Parámetro	Función del parámetro	Valor
$k$	Número de hormigas.	10
$g$	Número de generaciones de colonias de hormigas.	20
$\rho$	Tasa de evaporación de feromona	0.1
$q0$	Parámetro entre [0,1] que determina entre exploración y explotación.	0.6
$\beta$	Parámetro que determina la relevancia de la feromona contra la distancia.	6

Todo fue codificado en lenguaje de programación C# y probado en una computadora con un procesador Intel Xeon 5120 con 2 núcleos de 1.86 Ghz, con 3.0 Gb de memoria RAM, bajo el sistema operativo Windows XP Service Pack 3 de 32 Bits, con el software de Microsoft Visual C# Express 2010, con lo cual se produjeron los resultados mostrados en tabla 6.7.

Tabla 6.7. Promedios de vehículos y distancia recorrida por tipo de instancia obtenidos en este trabajo, en trabajos en el ITCM y estado del arte.

TIPO	ESTE TRABAJO		ITCM			GLOBAL		
	V	DR	V	DR	WORK	V	DR	WORK
C1	9.4444	1012.4209	10	830.72	ARA	10	828.38	PR
C2	3	593.8775	3	597.53	C	3	589.86	PR
R1	12.5	1234.8798	12.58	1266.9	C	11.92	1210.3	PDR
R2	2.8182	1057.4180	3	982.85	C	2.73	954.27	M
RC1	11.875	1441.8945	12.125	1416.7	C	11.5	1384.2	M
RC2	3.125	1390.8806	3.37	1171.7	HR	3.25	1108.52	PR
PROM	7.12710438	1121.895223	7.3458333	1044.4	N/A	7.06666667	1012.588333	N/A

# Capítulo 7

## CONCLUSIONES Y TRABAJOS FUTUROS

En este capítulo se describen las conclusiones de este trabajo, las aportaciones realizadas, así como también las sugerencias para el desarrollo de posibles trabajos futuros.

### 7.1 CONCLUSIONES

En este trabajo de investigación se abordó el Problema de Ruteo de Vehículos con Ventanas de Tiempo conocido como VRPTW. Para la solución a este problema se hizo uso de algoritmos metaheurísticos debido a que el problema está clasificado como NP-Duro [Garey, 1979]. Se logró mejorar el desempeño del algoritmo MACS-VRPTW a través de las estrategias implementadas.

Los algoritmos de Colonia de Hormigas que forman parte de la familia de ACO [Dorigo, 1991], fueron aplicados debido a las características del problema y las estrategias que aportaba el algoritmo para su solución.

Los resultados obtenidos con las diferentes versiones de MACS-VRPTW después de realizar y analizar las pruebas, confirman que son una opción viable para la solución de este problema, debido a que se reportan tiempos de ejecución competitivos respecto a lo presentado en por el estado del arte hasta este momento.

## 7.2 APORTACIONES DE LA INVESTIGACIÓN

Las contribuciones de este trabajo son:

- Esquema de solución adaptativo: se considera el recalcular de ventanas de tiempo después de agregar un cliente a la solución.
- Algoritmo de agrupación alterna para la lista restringida de candidatos.
- Algoritmo de agrupación semi-jerárquica para la lista restringida de candidatos.
- Análisis y rediseño de regla de LRE, en la que se considera NEI (intervalos no vacíos) como condición de activación.
- Genético de estrategias en colaboración con MACS-VRPTW
- Búsqueda local  $k$ -opt con algoritmo de Lin-Kernighan.

## 7.3 PUBLICACIONES

- “Ant Colony System with Characterization-Based Heuristics for a Bottled-Products Distribution Logistics System” 2013. Claudia G. Gómez S., Laura Cruz-Reyes, Juan J. González B., Héctor Fraire H., Rodolfo A. Pazos R., Juan J. Martínez P. Enviado a ICAM Elsevier (en revisión).
- “Análisis de factores que afectan el desempeño de metaheurísticos aplicados a VRPTW” 2013. Juan J. Martínez P, Claudia G. Gómez S., Laura Cruz-Reyes, Juan J. González B. Enviado a 1er Congreso Internacional de Investigación en Enseñanza de las Ciencias por el departamento de ciencias básicas del Instituto Tecnológico de Ciudad Madero (en revisión).

#### **7.4 TRABAJOS FUTUROS**

- Probar otros algoritmos metaheurísticas para hibridar con ACS.
- Ensayar con otros métodos de búsqueda local para producir la solución inicial.
- Incorporar MACS-VRPTW a ROSLOP
- Ejecutar el algoritmo MACS-VRPTW con instancias de mayor tamaño.

#### **7.5 PRODUCTOS CIENTÍFICOS**

- C.G. Gómez S., L. Cruz-Reyes, J.J. Gonzalez B., H. Fraire H., R.A. Pazos R., J.J. Martínez P., Ant colony system with characterization-based Heuristics for a bottled-products distribution logistics system, *Journal of Computational and Applied Mathematics* (2013).
- J.J. Martínez P., C.G. Gómez S., L. Cruz-Reyes, J.J. González B. Análisis de factores que afectan el desempeño de metaheurísticos aplicados a VRPTW. Instituto Tecnológico de Ciudad Madero. *Congreso ciencias básicas*. (2013)

## REFERENCIAS BIBLIOGRÁFICAS

[Ader, 2008] Chapter 12: Modeling. In H.J. Adèr & G.J. Mellenbergh (Eds.) (with contributions by D.J. Hand), *Advising on Research Methods: A consultant's companion* (pp. 271-304). Huizen, The Netherlands: Johannes van Kessel Publishing.

[Arranz, 2010] ”. Arranz Jorge y Parra Antonio. “*Algoritmos Genéticos*”. Universidad Carlos III.

[Arrañaga, 2007] B. Arrañaga, “*Búsqueda Local Mejorada para la solución del Problema VRPTW en un Sistema de Colonia de Hormigas*”, Tesis de Maestría, Posgrado en Ciencias de la Computación, Instituto Tecnológico de Ciudad Madero, 2007.

[Arya, 1994] Arya S, Mount D, Netanyahu N, Silverman R, Wu A, “*An Optimal Algorithm for Approximate Nearest Neighbor Searching in Fixed Dimensions*”

[Barto, 1981] A. G. Barto, R. S. Sutton y P. S. Brower, “*Associative search network: A reinforcement learning associative memory*”, *Biological Cybern.*, vol. 40, pp. 201–211, 1981.

[Bock, 1958] F. Bock, “An algorithm for solving traveling salesman and related network optimization problems”, *Fourteenth National Meeting of the Operational Research Society of America*, St. Louis, MO, Estados Unidos, 1958.

[Cámara, 2011], Cámara Ricardo, “*Impacto de la búsqueda local aplicada al Problema de Rutas de Vehículos con Ventanas de Tiempo*”, División de estudios de posgrado, Instituto Tecnológico de Ciudad Madero, 2009.

[Carnegie, 2009] Mellon University. “*Causal and Statistical Reasoning*”. Open Learning Initiative, 2009. <http://oli.web.cmu.edu/openlearning/>

[Croes, 1958] G.A. Croes, “A method for solving traveling salesman problems,” *Proc. Operations Research*, vol. 5, pp. 791–812, 1958.

[Dantzig, 1959] Dantzig, G. B., Ramser y J. H., “The truck dispatching problem”. *Mgmt Science*, vol. 6, pp. 80–91, 1959.

[Davison, 2003] <http://statwww.epfl.ch/davison/SM/SMSample.pdf>

[Díaz, 1996] Díaz A., González J., Laguna M., Mascato P., Tseng T., Glover F., Ghaziri M. “*Optimización Heurística y Redes Neuronales*”. Editorial Parainfo, España, 1996.

[Dorigo, 1991] A. Colorni, M. Dorigo et V. Maniezzo, Distributed Optimization by Ant Colonies, Actes de la première conférence européenne sur la vie artificielle, Paris, France, Elsevier Publishing, pp. 134-142, 1991.

[Dorigo, 1996] Dorigo, M., Maniezzo, V., Colorini, A., “The Ant System: Optimization by a Colony of Cooperating Agent”, *IEEE Transactions on Systems, Man and Cybernetics*, vol. 26, pp. 29-41, 1996.

[Gambardella, 1999] L. Gambardella, E. Taillar y G. Agazzi, “MACS-VRPTW: A Multiple Ant Colony System for Vehicle Routing Problems with Time Windows”, Reporte Técnico, IDSIA-06-99, IDSIA, 1999.

[Garey, 1979] M. R. Garey and D. S. Johnson, “Computers and Intractability: A Guide to the Theory of NP-completeness”, W. H. Freeman and Co., San Francisco, CA, 1979.

[Glover, 2003] Glover F., Kochenberger G., *Handbook of Metaheuristic*. Springer, 2003.

[Gómez, 2009] Claudia Gómez Santillán “*Afinación Estática Global de Redes Complejas y Control Dinámico Local de la Función Tiempo de Vida en el Problema de*

*Direccionamiento de Consultas Semánticas*”, Instituto Politécnico Nacional , Centro de Investigación en Ciencia Aplicada y Tecnología Avanzada, Unidad Altamira, 2009.

[Gómez, 2012] Claudia Gómez Santillán, Laura Cruz Reyes, María Lucila Morales Rodríguez, Juan Javier González Barbosa, Oscar Castillo López, Elisa Schaeffer. .”*Variants of VRP to Optimize Logistics Management Problems*”, División de Estudio de Posgrado de Instituto Tecnológico de Ciudad Madero, Instituto Tecnológico de Tijuana, Universidad Autónoma de Nuevo León.

[Gutin, 2002] Gutin, G. y Punnen, A. P. “*The Traveling Salesman Problem and its variations*”. Boston: Kluwer, 2002.

[Hashimoto, 2008] H. Hashimoto, M. Yagiura y T. Ibaraki, “*An Iterated Local Search for the Time-Dependent Vehicle Routing Problem with Time Windows*”, Discrete Optimization, vol. 5, pp. 434-456, 2008.

[Helsgaun,2000] Keld Helsgaun. "An Effective Implementation of the Lin-Kernighan Traveling Salesman Heuristic". Department of Computer Science Roskilde University

[Herrera, 2006] Herrera , J: “*Desarrollo de una Metodología Basada en Heurísticas para la Solución Integral de Problemas de Asignación de Rutas, Horarios y Cargas en el Proceso de Distribución y Entrega de Productos*”. División de estudios de posgrado, Instituto Tecnológico de Ciudad Madero, 2006.

[Hoshino, 2007] T. Hoshino, T. Kimura y T. Ikeguchi, “*Two simple local searches controlled by chaotic dynamics for the vehicle routing problems with time windows*”, In Abstract Proceedings of The Seventh Metaheuristics International Conference MIC 2007, Montreal, Canada, pp. 25-29, Jun. 2007.

[Johnson, 1995] D.S. Johnson, L.A. McGeoch, “*The Traveling Salesman Problem: A Case Study in Local Optimization*”, Local Search and Combinatorial Optimization, Editorial Wiley and Sons: New York, Estados Unidos, 1995.

[Kallehauge, 2006] K. B., “*The vehicle routing problem, monographs on discrete mathematics and applications, formulations and exact algorithms for the vehicle routing problem with time windows*”. (2006).

[Landeró, 2008] Landeró V. “*Desarrollo de un Método Formal que Muestre la Interrelación entre las Características de un Conjunto de Casos y las de un Algoritmo que los Resuelve Eficientemente, para el Problema de Distribución de Objetos en Contenedores*”. Tesis de doctorado, Dpto. de Ciencias Computacionales, Centro Nacional de Investigación y Desarrollo Tecnológico, Cuernavaca, Mor., 2008.

[Lemeire, 2007] Lemeire J. “*Learning Causal Models of Multivariate Systems and the Value of it for the Performance Modeling of Computer Programs*”. PhD thesis, Vrije Universiteit Brussel, 2007.

[Mariano, 2001] Mariano C. y Morales E. F., “DQL: A New Updating Strategy for Reinforcement Learning Based on Q-Learning”, *Proceedings of the 12th European Conference on Machine Learning*, Verlag, London, vol. 2167, pp. 324-335, 2001.

[Martí,2006] Martí Rafael "Procedimientos Metaheurísticos en Optimización Combinatoria" Departament d'Estadística i Investigació Operativa". Facultat de Matemàtiques.Universitat de València

[Menzies, 2009] T., Mizuno O., Takagi Y., Kikuno T. “*Explanation vs Performance in Data Mining: A Case Study with Predicting Runaway Projects*”. *J. Software Engineering & Applications*, 2: 221-236, 2009.

[Merli, 2008]

[http://webdelprofesor.ula.ve/economia/jramoni/MODELOS%20LINEALES%202/Clases/Estructuras\\_Datos.pdf](http://webdelprofesor.ula.ve/economia/jramoni/MODELOS%20LINEALES%202/Clases/Estructuras_Datos.pdf)

[Mester, 2006] D. Mester, O. Bräysy y W. Dullaert, “*A Multi-parametric Evolution Strategies Algorithm for Vehicle Routing Problems*”, *Expert Systems with Applications*, vol. 32, pp. 508-517, 2006.

[Montgomery, 2008] Montgomery D. “*Design and Analysis of experiments 7<sup>th</sup> edition*” . John Wiley and Sons ,2008

[Moore, 2005] Moore David. “*Estadística Aplicada Básica*”. Antoni Bosch Editor 2005.

[Nettleton, 2012].Nettleton David “*Técnicas para el análisis de datos clínicos*”. Ediciones Díaz de Santos, 2012.

[Parra,2002] Javier Parra Olivares.: “*Análisis exploratorio y análisis confirmatorio de datos*”. Espacio Abierto, enero-marzo, vol. 11, número 1. Cuaderno Venezolano de Sociología. Asociación Venezolana de Sociología. Maracaibo, Venezuela. Pp. 115-124. (2002).

[Pérez, 2007] Pérez Verónica: “*Modelado Causal del Desempeño de Algoritmos Meta heurísticos en Problemas de Distribución de Objetos*”. División de estudios de posgrado e investigación, Instituto Tecnológico de Ciudad Madero.

[Pisinger, 2006] D. Pisinger y S. Ropke, “*A General Heuristic for Vehicle Routing Problems*”, *Computers & Operations Research*, vol. 34, pp. 2403-2435, 2006.

[Potvin, 1995] J.Y. Potvin, y J.M. Rousseau, “An Exchange Heuristic for Routeing Problems with Time Windows,” *Proc. Journal of the Operational Research Society*, vol. 46, pp. 1433-1446, 1995.

[Prescott, 2009] E. Prescott-Gagnon, G. Desaulniers, y L.-M. Rousseau, “*A Branch-and-Brice Large Neighborhood Search Algorithm for the Vehicle Routing Problem with Time Windows*”, *Networks*, vol. 54, pp. 190-204, 2009.

[Prosser, 1996] P. Prosser y P. Shaw, “*Study of Greedy Search with Multiple Improvement Heuristics for Vehicle Routing Problems*”, Reporte Técnico, University of Strathclyde, Glasgow, Scotland, 1996.

[Quiroz ,2009]. Quiroz Castellanos Marcela. “*Caracterización de factores de desempeño de algoritmos de solución de BPP*”. División de estudios de posgrado, Instituto Tecnológico de Ciudad Madero,2009.

[Rutherford,2012], Rutherford A. “*ANOVA and ANCOVA: A GLM Approach*”. Wiley and Sons 2012.

[Sánchez, 1990] Julio Sánchez Meca, Estadística exploratoria y confirmatoria con el paquete Systat, EDITUM 1990

[Shaw, 1998] P. Shaw, “*Using Constraint Programming and Local Search Methods to Solve Vehicle Routing Problems*”, *Proceedings of the Fourth International*, pp. 417-431, 1998.

[Solomon, 1987] M. Solomon, M. “*Algorithms for the Vehicle Routing and Scheduling Problem with Time Window Constraints*” *Operations Research*, 1987.

[Stassen, 2007] Stassen Berger Kathleen: *Psicología del desarrollo (infancia y adolescencia)*, Ed. Médica Panamericana 2007

[Taillard, 1997] E. Taillard, P. Badeau, M. Gendreau, F. Guertin y J.Y. Potvin, “A Tabu Search Heuristic for the Vehicle Routing Problem with Soft Time Windows,” *Transportation Science*, vol. 31, pp.170-186, 1997.

[Thangiah, 2003] Thangiah Sam, “A site dependent vehicle routing problem with complex road constraints”, Proceedings of International Conference on Metaheuristics for Optimization (2003) ,1-17.

[Thangiah, 2010] Thangiah Sam R: “*Vehicle Routing with Time Windows using Genetic Algorithms*”, Artificial Intelligence and Robotics Laboratory, Computer Science Department, Slippery Rock University

[Vilar, 2005] José Francisco Vilar Barrio y Teresa Delgado Tejada, Control Estadístico de los Procesos (SPC), FC Editorial 2005